Monte Carlo EM for Data-Association and its Applications in Computer Vision

Frank Dellaert

21st September 2001

CMU-CS-01-153

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy.

Thesis Committee:

Charles E. Thorpe
Sebastian Thrun
Takeo Kanade
Richard Szeliski

Keywords: Computer vision, Correspondence Problem, Data Association.

for Katrien, Thomas and Zoe.

Abstract

Estimating geometry from images is at the core of many computer vision applications, whether it concerns the imaging geometry, the geometry of the scene, or both. Examples include image mosaicking, pose estimation, multi-baseline stereo, and structure from motion. All these problems can be modeled probabilistically and translate into well-understood statistical estimation problems, provided the correspondence between measurements in the different images is known.

I will show that, if the correspondence is *not* known, the statistically optimal estimate for the geometry can be obtained using the expectation-maximization (EM) algorithm. In contrast to existing techniques, the EM algorithm avoids the estimation bias associated with computing a single "best" set of correspondences, but rather considers the distribution over all possible correspondences consistent with the data. While the latter computation is intractable in general, I show that it can be approximated well in practice using Markov chain Monte Carlo sampling. As part of this, I have designed an efficient sampler specifically tuned to the correspondence problem.

The resulting Monte Carlo EM approach represents the first truly multi-view algorithm for geometric estimation with unknown correspondence. This is especially relevant in the structure from motion domain, where the state of the art relies on robust estimation of two or three-view geometric constraints. In addition, I will show that the probabilistic approach I propose allows for a seamless and principled way of integrating prior knowledge, appearance models, and statistical models for occlusion and clutter.

Acknowledgments

I am deeply grateful to Hans Moravec, Chuck Thorpe and Sebastian Thrun, my advisors (in order of appearance). Hans is an inspiration, Chuck a great listener, and Sebastian a delight to work with. Especially the interaction with Sebastian has shaped many of the ideas presented in this dissertation and in other research projects. I would also like to thank Takeo Kanade, who has been very creative in making sure I was somehow financially supported, no questions asked. Finally, I thank Rick Szeliski for agreeing to be on my committee and providing wonderful feedback.

I am also indebted to the many researchers at CMU whom I interacted with over the years, among them Steve Seitz, Dieter Fox, Wolfram Burgard, Yanxi Liu, Bob Collins, Martial Hebert, and Tom Minka. Many other people in the various research groups of the Robotics Institute have helped me in one way or another, and I thank them for that.

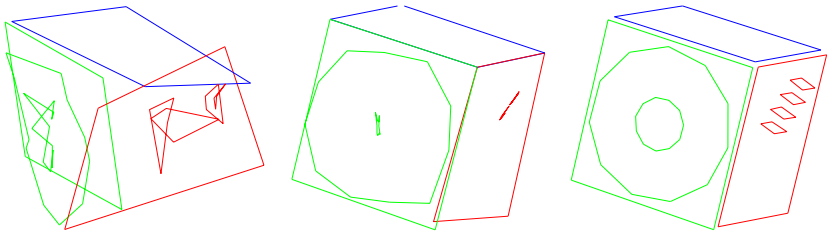
Many thanks also to my successive officemates, Somesh Jha, Daniel Morris, Farhana Kagalwaga, and to my friends at Carnegie Mellon and in Pittsburgh. Especially our friendly Coral Street neighbors and the OCT have made our social life more interesting.

Finally, I want to thank my family: Thomas and Zoe, and most of all, Katrien. Her love, support and understanding have made this all possible.

Contents

Chapter 1

Introduction



In this dissertation I advance the following **thesis**:

> The Monte Carlo EM algorithm provides a practical way to accurately approximate the optimal solution of multi-view geometric estimation problems with unknown correspondence.

Below I explain what this means, why it is novel and important, and how the dissertation is structured in order to support the thesis.

1.1 Geometric Estimation Problems

Many applications in computer vision can be summarized as *geometric estimation problems*. As an example, consider an early application of computer vision, illustrated in Figure 1.1 on the following page, where the goal is to align two aerial photographs taken from an airplane in order to create a larger "photo-mosaic". In this case, we are trying to estimate

Figure 1.1: An instance of a 2D image registration problem: the goal is to estimate the translation between the two images (images courtesy of the US Geological Survey).

the translation between the images, i.e. the geometry of the imaging situation. Hence, this is an instance of a geometric estimation problem. In the example, we used knowledge about the problem (e.g. the airplane was flying in a straight line) in order to *model* the situation in terms of two parameters: vertical and horizontal translation. Under different circumstances, we might have to use more complex models and estimate additional parameters, e.g. the rotation between the images or perspective distortion effects, etc.

Geometric estimation problems are at the core of many computer vision applications that have practical uses in society. 2D image registration problems, like the example of Figure 1.1, underlie photo-mosaicking software that nowadays comes with many digital cameras. Many industrial machine vision setups use a different type of 2D registration:here the goal is to locate a known template in a cluttered image, a process called pose estimation. Geometric estimation is not limited to 2D images: a common step in acquisition of 3D models using laser ranging is the registration of 3D scans to each other, or, in 3D pose estimation, locate a known object in a cluttered 3D scene. Finally, we can estimate the pose of a 3D object given (possibly multiple) 2D images.

A different type of geometric estimation involves recovering the structure and/or identity of an object. The applications mentioned above involve parameters of the imaging situation, i.e. calibration parameters, camera pose, or (equivalently) the pose of a known object with respect to the camera. Below I will frequently refer to this type of parameter as *motion* parameters. The converse estimation problem arises when we know the imaging setup, but would like to recover the *structure* of the scene or object that is seen. Stereo vision, with its countless applications, is the prototypical example. A related problem is that of object recognition, which can be viewed as structure recovery where the solution has to be

Figure 1.2: Structure from motion example. The goal is to estimate the 3D structure and the camera location associated with the images (4 out of 5 images shown).

selected from a restricted set of object prototypes and their allowable variations.

1.2 Structure from Motion

In many respects the most difficult geometric estimation problem attempts the recovery of structure and motion *simultaneously*. In other words, given a set of 2D images, recover both the imaging geometry and the 3D structure of the scene. This problem is known as *structure from motion* (SFM), and it has many applications ranging from building virtual worlds to robot mapping and localization. In addition, the class of structure from motion problems can be regarded as a superset of many of the other geometric estimation problems mentioned above, in which either the motion or structure are known. Figure 1.2 shows an instance of a structure from motion problem. In this example, features of interest were extracted from each of the images, and the goal is to recover the 3D position (structure) for

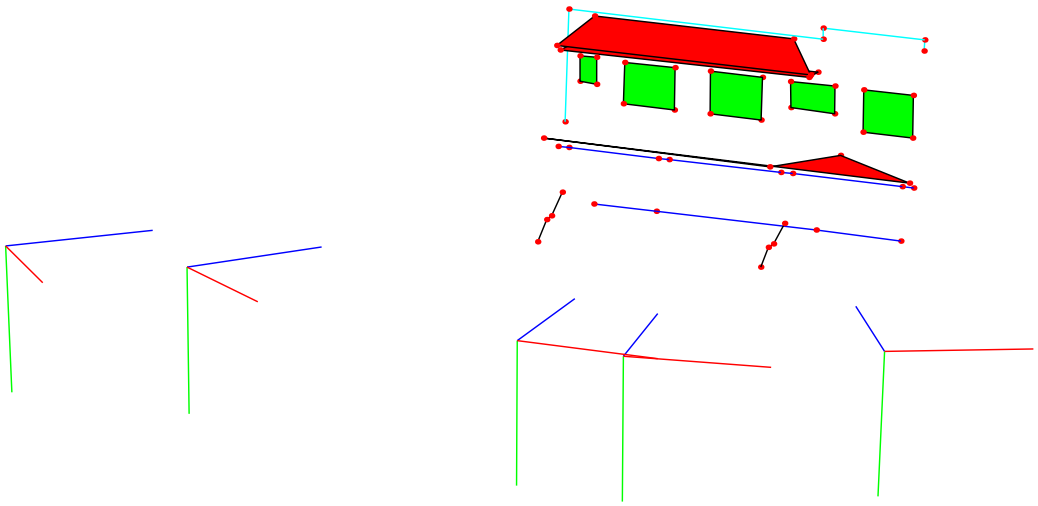


Figure 1.3: Estimated structure and motion for the image set in Figure 1.2.

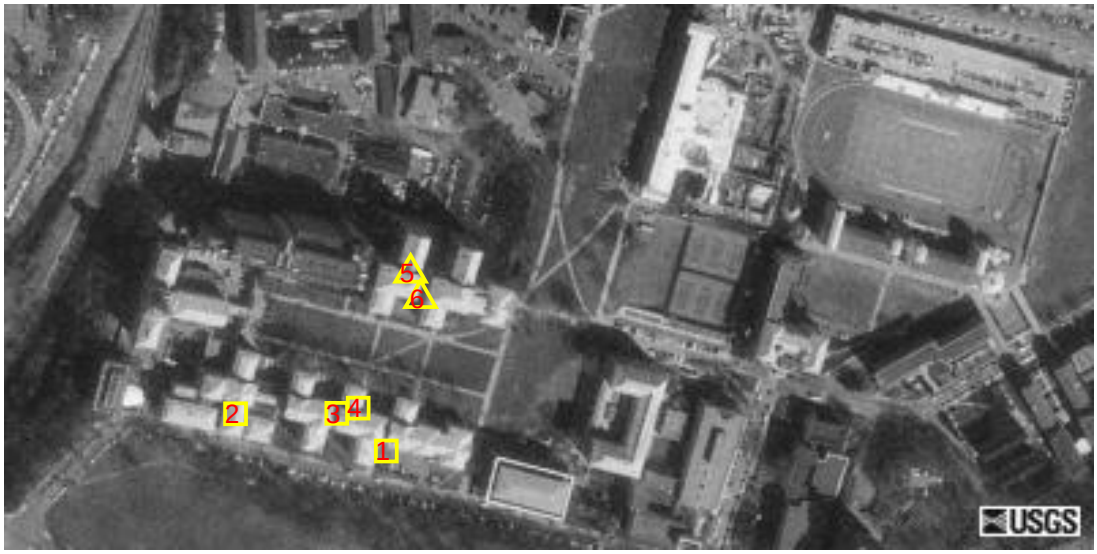
each of the features as well as the camera pose (motion) for each image. A rendering of such a structure and motion estimate is shown in Figure 1.3.

In this dissertation I will focus on feature-based structure from motion, as it is representative of many other geometric estimation problems.

1.3 Feature-Based Methods

In feature-based methods, rather than using the pixel-values of the images themselves (the "direct" method), a feature detector is used to first extract features of interest in the images. This dramatically reduces the amount of data we need to process, and hence much larger problems can be considered. In addition, it can be argued that low-texture regions in the image do not provide a lot of information with respect to the geometry, and hence concentrating on salient features captures most of the available information. Figure 1.4 illustrates the feature-based approach using a simple pose-estimation example. In this example, the goal is to recover the location (translation only) of the idealized model of the CMU quad (top panel, Figure 1.4a) in an aerial image of the CMU campus (bottom panel, Figure 1.4b). The quad model is specified as a set of 5 features, corresponding to "corner-like" buildings, of which there are two types. To find this model in the bottom image, a "corner-building detector" was used to extract 6 feature *measurements* that recapitulate the image. However, the detector is not ideal: the location of the features is not very precise, and one of the

(a) Model



(b) Image measurements

Figure 1.4: Model and image measurements (images courtesy of USGS).

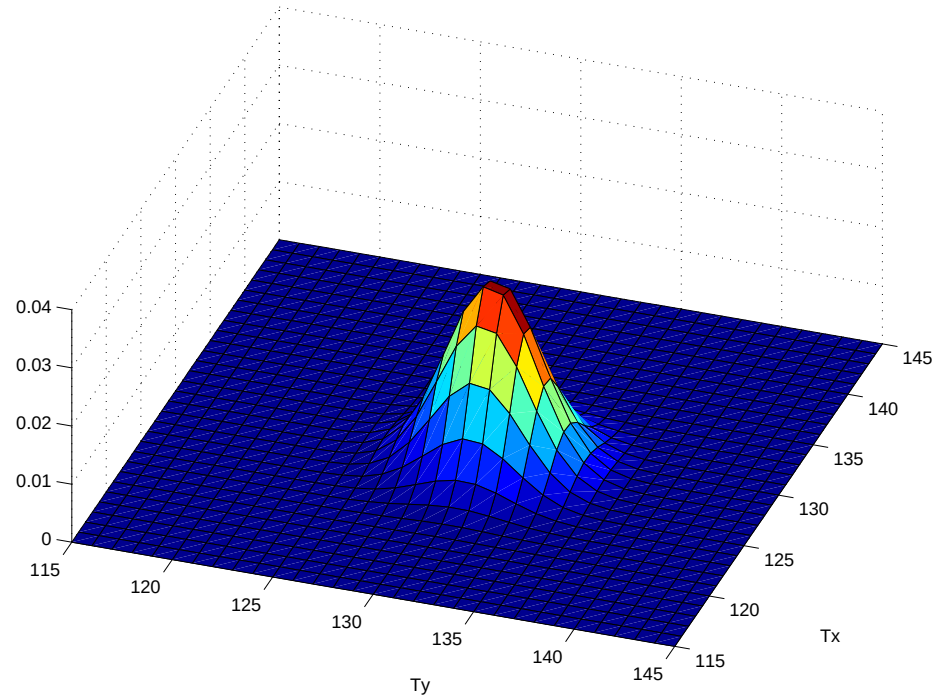


Figure 1.5: Likelihood function.

buildings yielded two measurements (labeled as measurements 1 and 4 in the image).

If the correspondence between the measurements in the image (i.e., the detected features in Figure 1.4b) and the features in the model (Figure 1.4a) is known, then estimating the translation is simple. In that case, the problem can be transformed (given some assumptions about the nature of the measurement noise) into an optimization problem. For example, in the pose-estimation problem of Figure 1.4, the optimal estimate for the two translation parameters can be found by maximizing the objective function shown in Figure 1.5. The function shown is called the *likelihood function*, and measures how probable the measurement data is given a value for the translation. It is defined over the *parameter space*, which in this case is two-dimensional, as there are two translation parameters to be estimated. The location of the maximum of the likelihood function is known as the *maximum likelihood estimate* (MLE) of the translation. Maximum likelihood estimation will be explained in more detail in Chapters 2 and 4.

1.4 The Correspondence Problem

Using extracted features to solve geometric estimation problems induces a *data-association problem*, also known as the *correspondence problem*. Indeed, if we consider the example from Figure 1.4, the flip-side of using features is that we now have to establish the correspondence between features in the model, and the measurements (detected features) in the image. If this correspondence is not given to us, it is unclear from looking at Figure 1.4b which measurements are associated with which model features. In particular, we cannot tell whether measurement 1 or 4 should be matched with the lower-right model feature. Also, it might just be possible that measurements 5 and 6 have switched location due to the large uncertainty in the position measurement of the feature detector.

Isolating a particular solution for the correspondence problem can lead to biased estimates of the unknown parameters. This is illustrated quite nicely in the simple pose-estimation example we have been considering. If we were to choose, arbitrarily, to use measurement 4 and discard measurement 1 as spurious, the location of the CMU quad would be biased towards the upper left. Conversely, if were to use measurement 1 and discard 4, the location would be estimated more towards the lower right. Below, in Section 1.6, I will show that this problem can be resolved in a principled manner.

While a hard problem even for two views, the correspondence problem becomes exponentially harder when multiple views are considered. In fact, there exist polynomial-time algorithms that can find an optimal match between two sets of features, given some measure of affinity between them. In contrast, finding an optimal multi-way correspondence between more than two sets is NP-complete. Nevertheless, this is the problem that will be considered in this dissertation:

This dissertation deals with multi-view, feature-based geometric estimation problems where the correspondence or data-association is unknown.

1.5 Existing Approaches to the Correspondence Problem

Solving the correspondence problem is crucial to feature-based geometric estimation, and is often described as the most difficult part of the problem (Torr et al., 1998). Consequently, it has received much attention in the literature. A detailed review of the literature on the correspondence problem in vision and the data-association problem in target-tracking is

given in Chapter 3. However, a comprehensive solution to the problem of structure and motion recovery with unknown correspondence has remained elusive:

- Until now, no true multi-view method for solving the correspondence problem for structure from motion (or any other application) has been proposed. The state of the art for SFM is based on multiview constraints that are limited to working with two or three views at a time. Hence, for larger sequences or image sets, the problem has to be split up in pairs or triplets whose solutions then have to be pieced together somehow. Citing (Hartley and Zisserman, 2000), this is still to some extent "a black art". The approach proposed in this thesis is inherently multi-view and considers the data in all images simultaneously to arrive at an optimal solution.

- Most approaches to correspondence in computer vision can be characterized as "pre-processing" algorithms. In isolating a single "best" correspondence, the resulting structure and motion estimate is biased by that arbitrary choice, as illustrated by the example from Figure 1.4. From a decision-theoretic point of view, a statistically optimal estimate should be obtained by considering a *distribution* over all possible correspondences consistent with the data, rather than a single one. The Bayesian approach I propose is statistically optimal, and has the additional benefit that prior knowledge about the solution can be added in a seamless manner.

- There have been efforts, both in the computer vision and target tracking literature, to use "soft correspondences" to capture the same idea of considering a distribution over correspondences, albeit in more restricted settings. The calculation of these soft correspondences is intractable for all but trivial problem instances. As a consequence, various ways of approximating them have been proposed, but all of them are unable to capture the important information provided by the *mutual exclusion constraint* (see below). In this dissertation I propose the use of *sampling* as a practical way to obtain more accurate estimates for these quantities.

All of these deficiencies are addressed by the Monte Carlo EM approach proposed in this dissertation. The following section provides a brief introductory overview of the proposed approach.

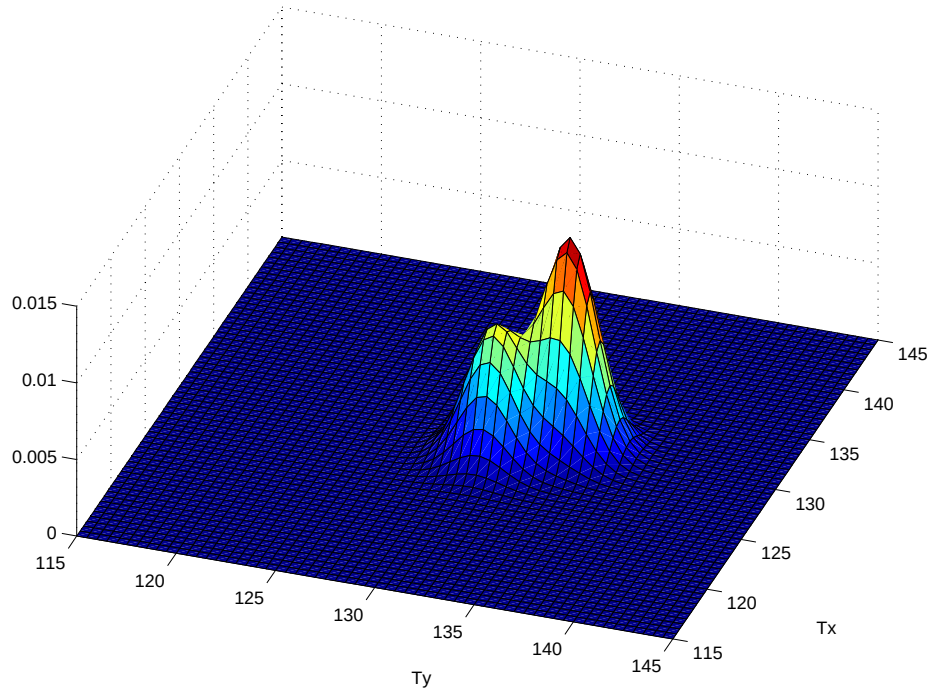


Figure 1.6: Likelihood function with unknown correspondence.

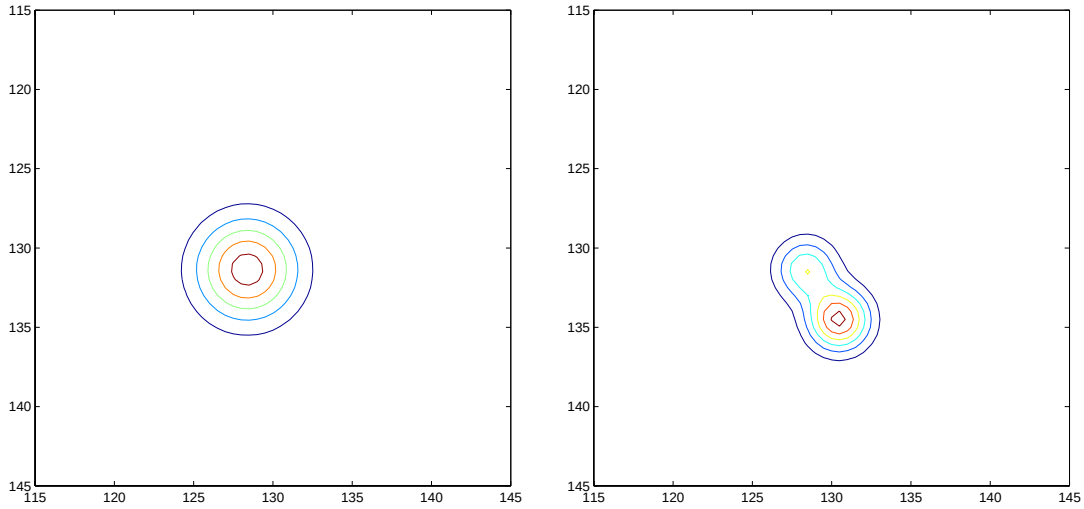
1.6 MCEM for Data-Association: Overview

1.6.1 Likelihood without Correspondence

There is a statistically correct way to model geometric estimation problems with unknown correspondence. In the same manner that a likelihood function can be constructed for the case of known correspondence (e.g. the function shown in Figure 1.5), we can construct a likelihood function for the parameters given just the measurements, with *no* correspondence information. For the pose estimation example above, this function is shown in Figure 1.6. For comparison, both functions are shown side by side as contour plots in Figure 1.7. Note that the new likelihood function has multiple peaks, i.e. it is multi-modal. In fact, for this example the local maximum closest to the ML estimate for known correspondence is actually the less likely one if we do *not* know the correspondence.

1.6.2 Deconstructing the Likelihood Function

As will be more formally derived in Chapter 4, the likelihood function given unknown correspondence can be obtained by summing together *all* the individual likelihood functions



(a) Known Correspondence (b) Unknown Correspondence

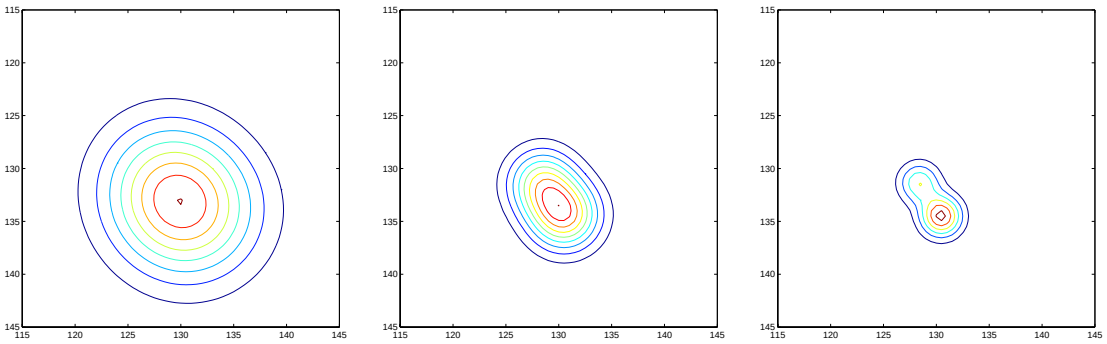
Figure 1.7: Likelihood with known and unknown correspondence.

for all possible correspondences. For the example, if for simplicity's sake we assume that all model features are actually seen in the image, the number of possible ways to match measurements to model features is equal to:

$$\binom{6}{5} \times 5! = 6 \times 120 = 720$$

This is because there are 6 ways to choose a subset of 5 measurements as corresponding to the model features, and for each of these sets there are 120 ways to permute them. However, if we also take into account that there are 2 distinct types of features, the number of possibilities is narrowed down to $\left[\binom{2}{2} \times 2!\right] \times \left[\binom{4}{3} \times 3!\right] = 2 \times 24 = 48$. For each of these possible ways to do the correspondence we have an objective function in 2D (the parameter space). They are all shown as contour plots in Figure 1.8. Note that the maximum likelihood estimate for translation shifts around, depending on how the correspondence is made. What is not obvious from the figure is that some of the functions have (much) higher values than others, because the probability of the measurement under a given correspondence varies. Adding all these constituent likelihoods together yields the function in Figure 1.6.

Figure 1.8: All components



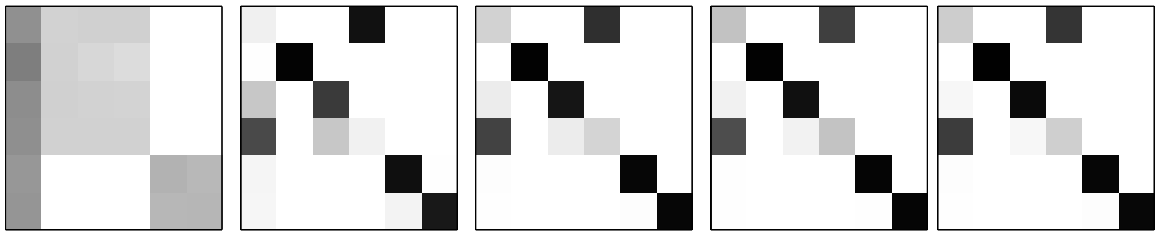
(a) high annealing (b) medium annealing (c) no annealing

Figure 1.9: Deterministic annealing can be used to smooth out local maxima.

1.6.3 An EM Approach to Correspondence

While in principle we could use this way of constructing the likelihood function in order to solve the problem with unknown correspondence, there are two substantial problems to be overcome. First, the number of possible correspondences grows combinatorially with the number of features, and, in the case of multiple views, with the number of views. In other words, this approach does not scale well. Second, the resulting likelihood function can be very complex and multi-modal. This is especially so in the case of structure from motion problems, where the parameter space has many more dimensions, and even the individual likelihood surfaces represent a coupled, non-linear optimization problem to be solved.

In this dissertation, I propose to circumvent the first problem, intractability, using the *expectation-maximization* (EM) algorithm. Whereas optimizing the true likelihood directly is intractable in general, because of the combinatorial nature of the problem, the EM algorithm provides an indirect way to find its maxima. Unfortunately, the EM algorithm is not guaranteed to find the global maximum of the likelihood, as it is performs a series of local approximations. Given that the objective function is complex and multi-modal, this is a significant hurdle. In order to cope with this local maxima problem, we can apply a standard trick from the optimization literature: *deterministic annealing*. In annealing, illustrated in Figure 1.9, the objective function is smoothed in such a way that many small local maxima disappear, and (hopefully) only strong, global maxima survive. By gradually decreasing the annealing factor and tracking the maximum throughout, one can hope to converge to the global maximum at termination.



(a) it 1 (b) it 2 (c) it 3 (d) it 4 (e) it 5

Figure 1.10: Evolving soft correspondences between the 6 measurements (each row) and the 5 model features (each column) from Figure 1.4, for 5 iterations of the EM algorithm. The first column in each panel is reserved to indicate "spurious" measurements.

1.6.4 Soft Correspondences as Marginal Probabilities

As will be discussed in detail in Chapter 4, the EM algorithm calls for the calculation of, for each measurement, the probability that it is associated with a certain feature given a current estimate for the unknowns. These probabilities are then used to re-estimate the unknowns, until they converge to a consistent estimate. The marginal probabilities can be interpreted as "soft correspondences", as illustrated in Figure 1.10 for the pose-estimation example from Figure 1.4. The EM algorithm was applied to this example and ran for 5 iterations, with a linearly decreasing annealing factor. The panels (a)-(e) in Figure 1.10 corresponds to the evolving soft correspondences for each of the 5 iterations, displayed as images. The darker a pixel is for a specific row-column intersection, the more probable the association between the measurement (associated with rows) and the feature (associated with columns). In the first iteration, the translation estimate is not very good, but the high annealing factor avoids making hard commitments to any given correspondence. Then, as EM converges and annealing decreases, we converge on a specific soft assignment of measurements to features.

This example also illustrates how ambiguous correspondences influence the final structure and/or motion estimate without making a commitment to a single, "best" correspondence. Indeed, the soft correspondences in Figure 1.10 are displayed in such a way that the ground truth correspondence corresponds to a diagonal matrix. However, the soft correspondence at convergence, shown in panel (e), accords more probability to measurement 1 as corresponding to a model feature rather than measurement 4, even though the latter is actually the true assignment. This can be related back to the multi-modal likelihood function from

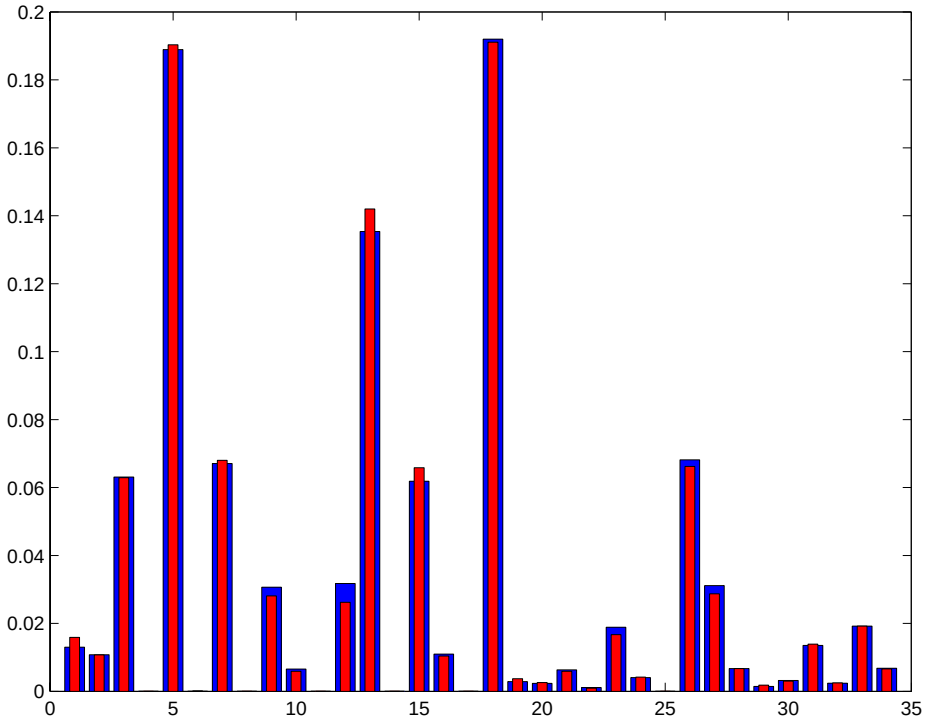


Figure 1.11: Result of sampling to approximate the true distribution (blue) by a sample histogram (red).

Figure 1.6: here the highest peak corresponds to the former assignment, and the lower peak to the latter (ground truth) assignment. Thus, even at convergence the possible ambiguity between those two possibilities is left unresolved.

1.6.5 Sampling

The final piece of the puzzle concerns the calculation of the marginal probabilities (i.e. the soft correspondences) themselves. It will become apparent (in Chapters 4 and 5) that this calculation is itself intractable. In fact, the problem of computing the marginals is intimately related to calculating the permanent of a matrix, a well studied problem in complexity theory which is known to be P#-complete, i.e. it is as hard as counting the number of solutions to certain NP-complete problems. In this dissertation, I propose the use of a Monte Carlo approximation to estimate the marginals, i.e. to estimate them by *sampling* over the space of possible correspondences. This is illustrated in Figure 1.11 (for a different example). In the figure, the true probability distribution over a set of possible correspondences is shown in blue, with correspondences lined up in an arbitrary order along the

ordinate axis. Thus, each bar is associated with a given correspondence, of which there are 34 different possible ones. Some correspondences are more probable than others, because the measurement error they imply can be more or less likely given the measurement model. Because the number of possible correspondences is very large in general, it is impossible in practice to calculate these probabilities exactly. However, in Chapter 5 it will be shown that a technique called Markov chain Monte Carlo sampling provides a practical way of approximating them. As part of this, a new, efficient sampler tuned to the correspondence problem is proposed. As an illustration, Figure 1.11 shows, in red, an approximation to the blue distribution obtained using a few hundred samples.

1.7 Thesis Revisited

Using the EM algorithm in conjunction with a Monte Carlo method to approximate the associated probability distributions is referred to as *Monte Carlo EM*. Given all of the above, the thesis can now be restated with no term undefined:

> The Monte Carlo EM algorithm provides a practical way to accurately approximate the optimal solution of multi-view geometric estimation problems with unknown correspondence.

The last section in this introduction discusses how the dissertation is structured in order to derive the MCEM approach to correspondence and to support the thesis that it is indeed a practical and useful tool for computer vision.

1.8 Dissertation Outline

In Chapter 2, I present an overview of the *structure from motion* (SFM) problem, since it is the application that motivated the work described in this dissertation. In addition, SFM can be seen as a superset of many other geometric estimation problems in computer vision, and hence is the ideal model-application to illustrate some key concepts.

In Chapter 3, I examine the state of the art in solving the correspondence problem, both in the context of structure from motion as well as in other related geometric estimation problems in vision.

In Chapter 4, I propose the EM algorithm as a practical way to estimate structure and motion parameters, given that the correspondence information is unknown.

Chapter 5 explains how Markov chain Monte Carlo sampling can be used to approximate a distribution over correspondence assignments. This can then be used to approximate the E-step in the MCEM-based approach to correspondence discussed in Chapter 4.

In Chapter 6, I demonstrate that the Monte Carlo EM approach does indeed provide a practical way to to approximate the optimal solution of multi-view geometric estimation problems with unknown correspondence. The results shown in this chapter assume that there is no occlusion or clutter in the images, which is a strong assumption.

This assumption is relaxed in Chapter 7, where I extend the MCEM approach to handle occlusion and clutter. Results under these assumptions are shown in Chapter 8.

The MCEM approach was derived under the assumption that the only information available is the position of the measurements in the images. Chapter 9 discusses how appearance information can be incorporated into the geometric estimation process. Finally, results with appearance are shown in Chapter 10.

Chapter 2

Structure from Motion

In this chapter I present an overview of the *structure from motion* (SFM) problem, since it is the application that motivates the work described in this dissertation. In addition, SFM can be seen as a superset of many other geometric estimation problems in computer vision, and hence is the ideal model application to illustrate some key concepts.

The SFM problem is introduced below under the assumption that the correspondence between measurements and 3D features is known, i.e. there is no data-association problem. At the end of this chapter, in Section 2.6, the correspondence problem or data-association is defined. Existing approaches to solve the correspondence problem are reviewed in the next chapter.

2.1 Problem Statement

The structure from motion problem is this: given a set of images of a scene, taken from different viewpoints, recover a 3D model of the scene along with the camera poses.

We will only be concerned here with a *feature-based* approach (Torr and Zisserman, 1999), where one assumes that there is a set of 3D features that can easily be detected in the images, using a feature detector. The problem then reduces to finding the most probable location of the 3D features given the location of their detected image projections. This is in contrast to *direct* or *image-based* approaches (Irani and Anandan, 1999), where typically the 3D structure is defined in image space, e.g. as a collection of depth or disparity values, and there is no feature detector.

The feature-based approach is characterized by the following set of properties:

- The 3D scene structure of interest consists of a collection of n 3D features, described by the *structure parameters* $\mathbf{X} = \{\mathbf{x}_j | j \in 1..n\}$.
- A set of m images is taken under under distinct circumstances described by the *motion parameters* $\mathbf{M} = \{\mathbf{m}_i | i \in 1..m\}$.
- Each image consists of K_i distinct *image measurements* $\mathbf{U}_i = \{\mathbf{u}_{ik} | k \in 1..K_i\}$, with $i \in 1..m$, each of which either corresponds to one of the n features $\mathbf{x}_j$, or represents a spurious measurement.

In addition to this standard formulation of the structure from motion problem, here we also explicitly model the correspondence between 2D measurements and 3D features:

- To indicate which measurement corresponds to which 3D feature, we introduce m *image correspondence vectors* $\mathbf{j}_i = \{\mathbf{j}_{ik} | k \in 1..K_i\}$ where the meaning of $\mathbf{j}_{ik}$ is the following (illustrated in Figure 2.1):
  - If $\mathbf{j}_{ik} = 0$. $\mathbf{u}_{ik}$ is considered spurious
  - otherwise, $\mathbf{u}_{ik}$ corresponds to the $\mathbf{j}_{ik}^{\text{th}}$ feature $\mathbf{x}_{\mathbf{j}_{ik}}$

The goal is to estimate the motion parameters $\mathbf{M}$ and the structure parameters $\mathbf{X}$.

2.2 Structure from Motion Applications

The typical SFM problem addressed in the literature is the one where the structural features are 3D points, and the measurements are their projections in images taken under orthographic or perspective projection. In this case, the structure parameters $\mathbf{X}$ consist of 3D coordinates $\mathbf{x}_j \in \mathbb{R}^3$, and the image measurements are their 2D projections $\mathbf{u}_{ik} \in \mathbb{R}^2$. As an illustration, consider Figure 2.1 where the various variables are illustrated for the perspective case.

However, more general structure from motion problems can also be accommodated within this framework. Here are some examples:

- The structure can be parameterized as a heterogeneous collection of features, e.g points and lines.

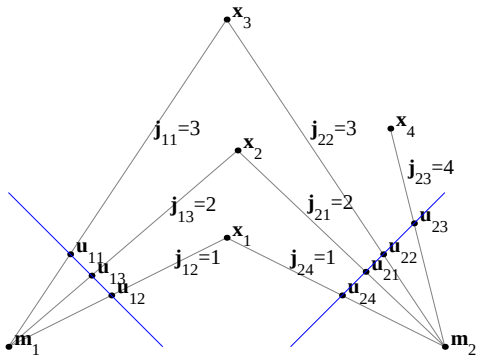


Figure 2.1: An example with 4 point features seen in 2 images. The 7 measurements $\mathbf{u}_{ik}$ are assigned to the individual features $\mathbf{x}_j$ by means of the correspondence variables $\mathbf{j}_{ik}$.

- The structure can be parameterized by a smaller number of variables, e.g. points that are constrained to lie on a plane or on the corners of a parameterized shape model.
- The camera model does not have to be a classical perspective projection. For example, omni-directional cameras are readily accommodated. In addition, the motion parameters $\mathbf{m}_i$ can include varying camera parameters such as focal length.
- The parameters $\mathbf{m}_i$ can also be parameterized by a smaller set of motion model parameters, if it is known that the camera was undergoing a smooth motion.
- Finally, time-varying structure can be accommodated using straightforward modifications.

2.3 SFM as Maximum Likelihood

Most existing approaches to SFM can be viewed as *maximum likelihood* (ML) methods, if we assume that the correspondence between the measurements and the 3D features is known. ML methods attempt to find those model parameters $\mathbf{\Theta}$ that are most likely to have generated the data. In our case we have

1. The model parameters $\mathbf{\Theta}$ consist of the 3D feature locations $\mathbf{X}$ and the camera poses $\mathbf{M}$, i.e., $\mathbf{\Theta} = (\mathbf{X}, \mathbf{M})$, the *structure* and the *motion*.
2. The data consists of the 2D image measurements $\mathbf{U}$, and the correspondence vector $\mathbf{J}$ that assigns measurements $\mathbf{u}_{ik}$ to 3D features $\mathbf{x}_{\mathbf{j}_{ik}}$.

The *maximum likelihood estimate* $\mathbf{\Theta}^*$ given the data $\mathbf{U}$ and $\mathbf{J}$ is given by

$$\mathbf{\Theta}^* = \underset{\mathbf{\Theta}}{\operatorname{argmax}} \ \log L(\mathbf{\Theta}; \mathbf{U}, \mathbf{J}) \tag{2.1}$$

where the likelihood $L(\mathbf{\Theta}; \mathbf{U}, \mathbf{J})$ is proportional to $P(\mathbf{U}, \mathbf{J}|\mathbf{\Theta})$, the conditional density of the data given the model.

To evaluate the likelihood, we need to assume a generative model. In particular, we assume that the generation of each measurement $\mathbf{u}_{ik}$ can be modeled by an ideal *measurement function* $\mathbf{h}$ followed by corruption with additive noise $\mathbf{n}$:

$$\mathbf{u}_{ik} = \mathbf{h}(\mathbf{m}_i, \mathbf{x}_{\mathbf{j}_{ik}}) + \mathbf{n}$$

This formulation implicitly assumes that a given measurement $\mathbf{u}_{ik}$ depends only on the camera parameters $\mathbf{m}_i$ for the image in which it was observed, and on the 3D feature $\mathbf{x}_{\mathbf{j}_{ik}}$ to which it is assigned. If global camera, motion, and/or structure parameters are modeled, they need to be included appropriately.

As a typical example, consider the case in which the features $\mathbf{x}_j$ are 3D points and the measurements $\mathbf{u}_{ik}$ are points in the 2D image (refer to Figure 2.1). In this case the measurement function can be written as a 3D rigid displacement followed by a projection:

$$\mathbf{h}(\mathbf{m}_i, \mathbf{x}_j) = \Pi_i[\mathbf{R}_i(\mathbf{x}_j - \mathbf{t}_i)] \tag{2.2}$$

where $\mathbf{R}_i$ and $\mathbf{t}_i$ are the rotation matrix and translation of the i-th camera, respectively, and $\Pi_i : \mathbb{R}^3 \rightarrow \mathbb{R}^2$ is a projection operator which projects a 3D point to the 2D image plane. Various camera models can be defined by specifying the action of this projection operator on a point $\mathbf{x} = (x, y, z)^T$ (Morris et al., 1999). For example, the projection operators for orthography and calibrated perspective are defined as:

$$\Pi_i^o[\mathbf{x}] = \begin{pmatrix} x \\ y \end{pmatrix}, \quad \Pi_i^p[\mathbf{x}] = \begin{pmatrix} x/z \\ y/z \end{pmatrix}$$

Finally, in order to perform ML estimation, we need to assume a distribution for the noise $\mathbf{n}$. In the case that the noise $\mathbf{n}$ on the measurements is i.i.d. zero-mean Gaussian noise with standard deviation σ, the negative log-likelihood is proportional to the sum of squared re-projection errors:

$$\log L(\mathbf{\Theta}; \mathbf{U}, \mathbf{J}) \propto -\sum_{i=1}^{m} \sum_{k=1}^{K_i} \|\mathbf{u}_{ik} - \mathbf{h}(\mathbf{m}_i, \mathbf{x}_{\mathbf{j}_{ik}})\|^2 \tag{2.3}$$

ML estimation with Gaussian noise models is equivalent to non-linear least-squares optimization. Substituting (2.3) into (2.1), one can see that in this case the maximum likelihood estimates $\mathbf{\Theta}^*$ for the structure and motion parameters $\mathbf{\Theta}$ are those that minimize the sum of squared errors between the measured and predicted 2D measurements:

$$\mathbf{\Theta}^* \stackrel{\Delta}{=} \underset{\mathbf{\Theta}}{\operatorname{argmin}} \sum_{i=1}^{m} \sum_{k=1}^{K_i} \|\mathbf{u}_{ik} - \mathbf{h}(\mathbf{m}_i, \mathbf{x}_{\mathbf{j}_{ik}})\|^2$$

A more realistic model for automatic feature detectors, where each measurement can have its own individual covariance matrix $\mathbf{R}_{ik}$, can also be accommodated. In that case we have

$$\log L(\mathbf{\Theta}; \mathbf{U}, \mathbf{J}) \propto -\sum_{i=1}^{m} \sum_{k=1}^{K_i} (\mathbf{u}_{ik} - \mathbf{h}(\mathbf{m}_i, \mathbf{x}_{\mathbf{j}_{ik}}))^T \mathbf{R}_{ik}^{-1} (\mathbf{u}_{ik} - \mathbf{h}(\mathbf{m}_i, \mathbf{x}_{\mathbf{j}_{ik}}))$$

In the following section, I review the most commonly used methods for solving this maximum likelihood problem.

2.4 Existing Methods for Structure from Motion

The structure from motion problem has been studied extensively in the computer vision literature over the past three decades. A good survey of techniques can be found in Hartley and Zisserman's recent book on multiple view geometry (Hartley and Zisserman, 2000; Faugeras and Luong, 2001).

The earliest work focused on reconstruction from two images only (Ullman, 1979; Longuet-Higgins, 1981; Tsai and Huang, 1984). Later methods were developed to handle multiple images, and they can all be viewed as minimizing an objective function such as (2.3), under a variety of different assumptions.

In certain cases, matrix *factorization* techniques can be used to solve the least-squares problem associated with the structure from motion problem. In the case of orthographic projection, i.e., the projection is orthogonal to the image plane and has its focus at infinity, the estimate $\mathbf{\Theta}^*$ for the model parameters that minimize (2.3) can be found efficiently by factorizing a measurement matrix (Tomasi and Kanade, 1992). Using this technique, singular value decomposition (SVD) is first applied to a matrix derived from the data $\mathbf{U}$ in order to obtain *affine* structure and motion, denoted by $\mathbf{X}^a$ and $\mathbf{M}^a$. They are called affine because they are only defined up to a 3D affine transformation. The correspondence

information $\mathbf{J}$ is needed to re-arrange the data $\mathbf{U}$ in the correct order needed for SVD. To get Euclidean structure and motion, an additional step is needed that imposes metric constraints on $\mathbf{M}^a$. The factorization method has the advantages that it is fast and does not need a good initial estimate of structure and motion to converge. It has been applied to more complex camera models, i.e., weak- and para-perspective models (Poelman and Kanade, 1997), and even to fully perspective cameras (Triggs, 1996). These are well developed techniques, and the reader is referred to (Tomasi and Kanade, 1992; Poelman and Kanade, 1997; Morris and Kanade, 1998) for details and additional references.

In more general cases, one needs to resort to non-linear optimization to minimize the re-projection error (2.3). For example, in the case of full perspective cameras the measurement function $\mathbf{h}(\mathbf{m}_i, \mathbf{x}_j)$ is non-linear, as the projection involves a division by the depth of a feature point relative to the camera. Solving the associated nonlinear least-squares problem is known in photogrammetry and computer vision as *bundle adjustment* (Spetsakis and Aloimonos, 1991; Szeliski and Kang, 1993; Weng et al., 1993; Hartley, 1994; Cooper and Robson, 1996; Kang and Szeliski, 1997; Triggs et al., 1999). The advantage with respect to factorization is that it gives the exact ML estimate, when it converges. It is also more robust to noise. The disadvantage, however, is that it can get stuck in local minima, and thus a good initial estimate for structure and motion needs to be available. To alleviate this, recursive estimation techniques can be used to process the images as they arrive (Broida and Chellappa, 1991; Azarbayejani and Pentland, 1995). As an aside, techniques based on non-linear minimization can handle very general problems, e.g. more complex camera models (for example omnidirectional cameras) or measurements that mix points, lines, curves, etc. For example, a recent paper that discusses how to work with line segment measurements is (Taylor and Kriegman, 1995).

Most of the structure from motion results shown in this document are obtained using my own implementation of bundle adjustment. It uses the sparse solver techniques described in (Hartley, 1994) in order to accommodate large problems, and implements inner constraints (Cooper and Robson, 1996) to obtain well-behaved problems in the face of the position and scale ambiguity inherent to the SFM problem. Finally, it can easily deal with a variety of camera models and prior-knowledge constraints through the use of automatic differentiation. These techniques are described in more detail in Appendix A.

2.5 Incorporating Prior Knowledge

If prior knowledge is available, it can be readily incorporated by performing *maximum a posteriori* (MAP) estimation rather than maximum likelihood estimation. While most existing SFM methods assume no prior knowledge on either structure or motion, at their core they are all optimization methods and thus can be easily extended to incorporate such prior information. The MAP estimate $\mathbf{\Theta}^*$ for structure and motion $\mathbf{\Theta}$ is the one that satisfies

$$\mathbf{\Theta}^* = \underset{\mathbf{\Theta}}{\text{argmax}} \ \log P(\mathbf{\Theta}|\mathbf{U},\mathbf{J}) = \underset{\mathbf{\Theta}}{\text{argmax}} \ \{\log L(\mathbf{\Theta};\mathbf{U},\mathbf{J}) + \log P(\mathbf{\Theta})\} \tag{2.4}$$

where $L(\mathbf{\Theta};\mathbf{U},\mathbf{J})$ is the likelihood term, as discussed above, and $P(\mathbf{\Theta})$ is a prior probability density on the structure and motion parameters, e.g. a motion smoothness prior.

2.6 The Correspondence Problem

All of the above assumes that the correspondence between measurements $\mathbf{u}_{ik}$ in the different images and the 3D features $\mathbf{x}_j$ is known, i.e. the image correspondence vectors $\mathbf{j}_i$ are *known*. In a more general case this association between measurements and model parameters is not known, i.e. we are faced with the *correspondence problem*. In the tracking literature, this problem is more commonly known as the *data-association* problem.

Typically, the correspondence problem is seen as a separate step, to be done before nonlinear minimization is even attempted. In this dissertation a different approach is taken, where the correspondence problem is "solved" in parallel with the estimation of structure and motion. In the next chapter I review the literature on the correspondence/data-association problem, where I attempt to both give a chronological history of approaches to the problem, as well as trace the origins of the ideas that underlie my own work.

Chapter 3

Related Work

In this chapter I examine the state of the art in solving the correspondence problem, both in the context of structure from motion as well as in other related geometric estimation problems in vision. The approaches to recover 3D structure from motion, reviewed in the previous chapter, assume that the correspondence between 2D measurements and 3D features is known. If the correspondence is not known, typically a pre-processing step is applied which outputs a single "best" correspondence between the features projections in the different images. Alternatively, a robust estimator is used which estimates the structure and the correspondence at the same time (for image pairs or triplets). Both approaches are discussed in detail below.

Methods to solve the correspondence problem have a rich history, and play a central role in a variety of computer vision applications. Feature correspondence is crucial not just to recovering structure from motion, but also to image registration, 2D and 3D object recognition, and (multiple-baseline) stereo. Historically, these applications have spawned much of the literature on the correspondence problem. The literature review below is organized according to these different applications, in order of complexity, which roughly reflects their chronological development.

It is my thesis that many of these approaches, in which typically a "best" correspondence solution is singled out, are deficient. They do not solve the correct geometric estimation problem, but are biased by insisting on a single solution to the correspondence problem. Rather, to be correct, a distribution over all possible correspondence solutions should be considered.

In this dissertation, these deficiencies will be remedied by proposing a technique based on

expectation-maximization (EM). Insofar EM-type algorithms have been used in the computer vision literature, it has not been realized that EM is indeed the correct framework (with the exception of (Wells, 1997) in the context of recognition). In addition, the E-step (even if it is not called that) is always approximated in such a way that mutual exclusion constraints can not be fully and correctly exploited. This will be remedied here by the introduction of MCMC sampling to implement the E-step.

The EM-based approach also provides a solution to the correspondence problem in structure from motion. None of the algorithms for finding correspondences directly apply to the SFM problem in the wide-baseline case, because the appearance of a 3D structure can radically change when projected in separated 2D views. The current state of the art relies on the robust recovery of multi-view constraints. However, these approaches are (a) limited to working with either image pairs or triplets, and (b) again suffer from isolating a single "best" correspondence, even though there might be ambiguity in the data. Both deficiencies are remedied by the use of the EM-based approach proposed here.

The correspondence problem has also been studied in depth in the target tracking literature, where it is more commonly known as the data-association problem. The methods employed in this community were (and are) often more sophisticated than those used in the computer vision community at the same time, and are always based on firm probabilistic principles. Data-association techniques are reviewed below for the single and multiple target tracking cases. One particular development, though less well known, is of particular interest in the current context: the use of the EM algorithm for multiple-target smoothing in the PMHT filter (see Section 3.2.3). In its use of the EM algorithm, the PMHT is completely equivalent to the EM-based approach introduced in this dissertation. However, the PMHT (a) does not address the recovery of motion (or sensor) parameters as it was developed for tracking or smoothing, and (b) the E-step is either implemented using an intractable brute force method or approximated in such a way that mutual exclusion constraints are totally disregarded.

3.1 The Correspondence Problem in Vision

In the computer vision literature, the data-association problem is often phrased as a token matching problem, and is typically referred to as the *correspondence problem*. In contrast to the tracking literature, which is focused on time-recursive formulations, the correspondence problem is mostly about matching between different images or point sets. Sometimes these images are actually part of a time sequence, but typically no use is made of this fact. Of

course, token tracking is used extensively in vision as well, but we can refer to the tracking literature for those applications. Cox gives a good overview in (Cox, 1993).

Below, I review and comment on the literature following a rough chronological outline. Historically, correspondence first showed up in simple (translational) image registration problems, which were later extended to deal with more complicated transformations (affine, projective), registering 3D to 3D points sets, and 2D to 3D alignment for pose recovery/recognition applications. All of these are focused on recovering a global transformation or *motion* between two sets or graphs, and the individual location of the points in the image or reference model is assumed known or unimportant. Recovering *structure* is the focus of sparse (possibly multi-view) stereo applications. Finally, the desire to recover both motion *and* structure underlies the techniques for estimating the fundamental matrix and more general structure from motion approaches.

3.1.1 2D to 2D Matching

Non-geometric Matching

The first uses of correspondence analysis made no reference to geometry at all. By "non-geometric matching" I mean the problem of finding an optimal match between two points based on some measure of optimality that depends on the application. Whereas there are several polynomial algorithms (Papadimitriou and Steiglitz, 1982; Bertsekas, 1991; Cook et al., 1998) to solve the bipartite assignment problem if a general edge cost function is given, some other approaches explored in the vision literature are worth discussing for the ideas they introduce, even though they are sub-optimal.

The seminal reference for framing the correspondence problem and the general principles involved is (Ullman, 1979). He proposed an algorithm that was designed with a biological implementation in mind, i.e. it based on parallel, local computations. More theoretically motivated techniques were inspired by statistical physics and used graduated-convexity optimization as a means to recover the match between two sets. In (Kosowsky and Yuille, 1994), it is shown that the optimal assignment between two point sets (with an equal number of points) can be found through minimizing an effective energy function derived from a mean-field approximation. Minimizing this criterion is done by slowly decreasing the temperature in the mean-field approximation and tracking the solution, in order to select the proper global minimum at the lowest temperature. The entities minimized are continuous variables that converge on $\{0, 1\}$ at the solution for $T = 0$.

Effectively the same technique is used in (Rangarajan and Mjolsness, 1994; Gold and Rangarajan, 1996) in order to match two graphs, and this paper was the first in a long string of papers based on the "soft-assign" algorithm. Inexact graph matching is introduced in (Shapiro and Haralick, 1981) for purposes of recognition, and it involves essentially the same problems as the matching of point sets. In essence, binary assignment variables are replaced by continuous values, and two-way constraints that enforce mutual exclusion are gradually enforced as to arrive at the permutation matrix that constitutes a solution.

Recovering Translation

The earliest geometric application involving the correspondence problem was matching two sets of 2D points for purposes of (translational) image registration. One of the earliest approaches to the problem was chamfer matching (Barrow et al., 1977), and related approaches are still popular today, see e.g. (Huttenlocher et al., 1993; Gavrila and Davis, 1996; Olson, 2000). Chamfer matching is based on the distance transform (Rosenfeld and Pfalz, 1966), which can be used to efficiently compute the distance of a point to the nearest line or point in a reference image.

Relaxation labeling (Ranade and Rosenfeld, 1980; Wang et al., 1983; Price, 1986) is an iterative technique to recover the translation between images. It identifies point pairs whose translation has large support among other point pairs, and as such it is a precursor of RANSAC (see below). (Ton and Jain, 1989) introduces the enforcement of two way mutual exclusion constraints for relaxation labeling, and (Li, 1992) used graduated convexity to improve the global convergence.

The downside of these approaches, however, is that none are readily applied to transformations other than translation. In addition, they were developed without the benefit of a firm theoretical framework.

More General Transformations

In the literature on object recognition more general transformations need to be considered. Typically a match is sought between a 2D model and a 2D image that contains a transformed copy of the model. Because of the presence of many distracting features and the resulting huge space of possible matchings, early research focuses on efficiently eliminating large parts of the search space, through indexing and pruning (Baird, 1985; Ayache and Faugeras, 1986; Grimson and Lozano-Pérez, 1987).

A different, SVD-based approach to recovering a rigid 2D or affine transformation between points sets is taken by (Scott and Longuet-Higgins, 1991; Shapiro and Brady, 1992). In the former paper, a clever technique based on the SVD of a proximity matrix yields a correspondence technique without iteration. However, it is not able to deal with large rotations, prompting the latter paper in which point sets are analyzed in terms of a shape description before the matching process. An eigenstructure analysis is also applied to registering and recovering the transformation parameters between two graphs (Umeyama, 1988; Umeyama, 1991; Umeyama, 1993).

Approaches based on mean-field approximations (Lu and Mjolsness, 1994) and the softassign algorithm (Rangarajan et al., 1997; Gold et al., 1998) were also applied to the problem of recovering global transformations. In addition, these papers introduce the idea of iterating between a (soft) assignment between the points and pose transformation parameters. This thus parallels the development in tracking/smoothing applications of using EM to iterate between the sought parameters and the "nuisance" assignment variables (Section 3.2.3), albeit less formalized and theoretically motivated. Recently, (Chui and Rangarajan, 2000) showed it is possible to recover non-rigid 2D transformations as well using these same ideas, and (Cross and Hancock, 1998) use an EM-type algorithm to match and recover transformation parameters between graphs.

Finally, (Boykov and Huttenlocher, 1999) uses graph-algorithms to optimize for both correspondence and pose, modeling the correspondence using a Markov random field.

All of these algorithms single out a "best" correspondence and recover the 2D global transformation associated with it. Hence, in essence they solve an incorrectly posed problem. In fact the correspondences are nuisance variables that should in principle be integrated out. Even the EM-type approaches by Rangarajan and colleagues anneal the temperature down to zero, forcing the data-association matrix into a binary stochastic matrix. By introducing the EM framework that pitfall is avoided here. In addition, the sampling-based E-step introduced here models the mutual exclusion constraint in a more powerful way than possible with the mean-field approximation.

3.1.2 3D to 3D

Many of the techniques developed for 2D to 2D matching can conceivably be used to register 3D point sets as well, and indeed some authors explicitly mention this (Gold et al., 1998). Bipartite matching has been used in the 3D domain, to match features extracted

from both the 3D scene and a model (Kim and Kak, 1991). However, when working with the raw 3D data itself, there is no clear similarity between 3D points, except Euclidean distance after transforming the model to the scene, given some estimate of the transformation. One of the most popular algorithms that uses this idea is the *iterative closest point (ICP)* algorithm (Besl and McKay, 1992), which iterates between solving for the best possible transformation and the best possible assignment. A non-rigid extension is given in (Feldmar and Ayache, 1996).

ICP can be seen as an EM-type approach in which the E-step approximates the distribution over possible correspondences using a single (optimal) correspondence set. Thus, again the possible ambiguity in the data is discarded. As a result, the resulting algorithm is not guaranteed to converge to a local maximum of the likelihood function. This is analogous to the difference between using K-means or EM in a clustering application.

Thus, of considerable interest are the "soft" versions of such algorithms, which behave like EM in allowing some ambiguity with respect to the matching. In particular, Szeliski proposes the use of "slippery springs" in (Szeliski, 1989). These can be visualized as springs that can slide across a 3D surface, and hence remain ambiguous as to exactly which location on the 3D surface they correspond to. This concept is partly based on earlier work on "elastic nets" used to solve travelling salesman problems(Durbin and Willshaw, 1987; Durbin et al., 1989), which used a similar objective function in combination with annealing (in a similar manner to what is proposed in this dissertation). Finally, these ideas have also been used in the registration of 3D medical images in (Grimson et al., 1996).

3.1.3 3D to 2D

The interjection of a projection in the transformation process makes 2D to 3D matching somewhat different. Its applications are both the recovery of 3D pose from a 2D image (or images) and the recognition of 3D objects. A survey of early work in this field is given in (Binford, 1982). An overview of bounded search-based methods is given in (Grimson, 1990).

It is in this domain that the RAndom SAmple Consensus (RANSAC) algorithm (Bolles and Fischler, 1981; Fischler and Bolles, 1981) was first developed. RANSAC is a robust fitting technique, which, in general, searches for a transformation between a model and a dataset with maximal matching support. Applied to 3D pose recovery from a 2D perspective image, the algorithm works by sampling minimal sets of correspondences, and evaluating the

associated recovered pose in terms of how many additional matches are close under this transformation. RANSAC is a completely general robust fitting scheme and has been used in many more contexts (see below).

Some recent probabilistic methods are especially worth mentioning. In (Hornegger, 1997; Paulus et al., 1997) a description of an object is given in probabilistic terms, and their approach focuses on deriving a joint probability density function for a set of points, parameterized by pose. In this derivation, the marginalization over individual point assignments is already done, so pose recovery is simply done by maximizing the likelihood of the joint set of points. Once pose is recovered, object recognition is done in the same way.

(Wells, 1997) uses an EM-based approach for correspondence/pose recovery in object recognition. In his posterior marginal pose estimation (PMPE) algorithm, he frames the recovery of pose as a MAP estimation problem under the assumption that correspondence is hidden, and uses the EM algorithm to optimize for pose. This is again the same use of EM as in the tracking/smoothing literature and in this dissertation. To quote Wells, using EM for pose recovery is an "attractive alternative to combinatorial search, particularly when combined with indexing methods, which typically yield somewhat inaccurate pose estimates, since they are based on minimal sets of corresponding features". However, in the E-step, Wells does not attempt to enforce a mutual-exclusion constraint on the matching process.

3.1.4 (Multiple-Baseline) Stereo

Correspondence also plays a central role in stereo applications, by which I mean any multi-view setup in which the geometry of the imaging setup is known and the cameras are internally calibrated. Sparse stereo, where one works with features extracted from the images, is then similar to smoothing approaches in the tracking literature, in that the characteristics of the sensors are assumed known. What remains is a large data-association problem, i.e. determining the correspondence of features in the different images. In contrast to the multi-target smoothing problem, however, the number of 'targets' is vastly larger.

Sparse stereo is to be contrasted with dense stereo, typically using two or three views, where a dense depth map is recovered. The use of matching algorithms for dense stereo has been investigated (Fielding and Kam, 1997; Fielding and Kam, 2000), but the authors concluded that dynamic programming remained the algorithm of choice. The nature of the dense stereo problem is somewhat different, as the depth map automatically establishes a correspondence between the images.

The literature on stereo is vast, and here I only mention those papers that have an obvious relationship to the work presented in this document. (Cheng et al., 1994) used the eigenvalue-based approach of (Scott and Longuet-Higgins, 1991) to solve the correspondence problem in a sparse two-view situation. The more fundamental idea in the paper concerned the affinity between points in the two images, which is based on computing a 3D "pseudo-intersection" point. In a later paper (Cheng et al., 1996), more efficient bipartite matching algorithms are used. (Pilu, 1997) uses the eigenvalue approach but adds an appearance term to the proximity measure.

In (Yuille et al., 1991), the stereo problem is formulated explicitly with a matching field, entirely analogous to the data-association vectors in multi-target tracking (Section 3.2.2). Whereas in the tracking problem the posterior probability of the state sequence is maximized, in stereo the posterior probability of the disparity field is maximized. Yuille et. al. explore both the elimination of the disparity field (equivalent to Rao-Blackwellization, as in (Bergman and Doucet, 2000), see below), as the elimination of the matching field. The latter is done through a mean-field approximation, and minimization of the resulting effective energy function of the disparity field is done using deterministic annealing. Effectively, this implements a winner-take-all strategy, where at convergence only one match determines the disparity at a given feature location.

Multiple-baseline stereo (Okutomi and Kanade, 1993) presents an added challenge, as three or in general N-view matching is an NP-complete problem: there is no known algorithm that can find an optimal matching in polynomial time. (Cox et al., 1996) gives a maximum-likelihood formulation of the N-view, pixel-based stereo problem (where pixel intensities themselves are used to match between images, rather than image neighborhoods). Their method to account for occlusion and clutter is inspired by the tracking literature, in particular (PattiPati et al., 1990). However, again dynamic programming was found to be more suitable to deal with the unique properties of stereo than an optimal matching algorithm, where, for one, it is not impossible to include smoothness constraints. An alternative, maximum flow based approach is presented in (Roy and Cox, 1998). For solving the sparse N-view problem, (Bedekar and Haralick, 1996) take a brute force approach where all possible N-view correspondences are tested using a χ^2 test.

The shortcomings of winner-take-all approaches have already been discussed above. As remarked before, the mean-field approximation is problematic as it does not accurately model mutual exclusion. A novel and more correct approach to multi-view stereo based on the ideas presented here would use EM combined with sampling to approximate the (otherwise intractable) E-step.

3.1.5 Structure from Motion

The majority of literature on SFM considers special situations where the data association problem can be solved easily. Some approaches simply assume that data correspondence is known *a priori* (Ullman, 1979; Longuet-Higgins, 1981; Tsai and Huang, 1984; Hartley, 1994; Morris and Kanade, 1998). Other approaches consider situations where images are recorded in a sequence, so that features can be tracked from frame to frame (Aggarwal et al., 1981; Broida and Chellappa, 1991; Tomasi and Kanade, 1992; Szeliski and Kang, 1993; Lee and Joshi, 1993; Poelman and Kanade, 1997; Kang and Szeliski, 1997).

Several authors considered the special case of correct but incomplete correspondence, by interpolating occluded features (Tomasi and Kanade, 1992; Jacobs, 1997; Basri et al., 1998), or expanding a minimal correspondence into a complete correspondence (Seitz and Dyer, 1995). In (Forsyth et al., 1999), it is shown that Markov chain Monte Carlo sampling can be used to identify small errors in a given set of correspondences. However, all these approaches require that a non-degenerate set of correct correspondences be provided a priori.

An early approach to solve the correspondence problem in the SFM domain was (Lee and Huang, 1988), which took a brute force approach by enumerating all possible correspondences of 4 points, and assessing the quality of the resulting solution. In this sense, it can be considered a fore-runner of the RANSAC approach (see below). Another approach used the moments of the point cloud in the image in order to estimate the relative orientation of two images (Goldgof et al., 1989; Goldgof et al., 1992).

Since the landmark papers on the fundamental matrix (Luong and Faugeras, 1996) and the trifocal tensor (Shashua and Werman, 1995; Hartley, 1997), projective approaches to SFM became very popular. They proceed by first computing multi-view constraint matrices between two or three views, after which obtaining projective structure is easy (Hartley and Zisserman, 2000). Upgrading to a metric reconstruction is then done through specialized algebraic methods and/or non-linear optimization. The correspondence problem only appears in the first step. Torr and colleagues (Torr and Murray, 1993; Torr and Murray, 1997; Beardsley et al., 1996; Torr and Zisserman, 1998) propose the use of RANSAC (Bolles and Fischler, 1981) to perform robust estimation of the fundamental matrix between two views, or the trifocal tensor between image triplets. This is done by first hypothesizing a seed set of possible matches, and then using RANSAC to search for the minimal sets with the most support. The recovered constraint is then used to guide matching, which is fed back to RANSAC, and so on until the estimate has stabilized. The use of least me-

dian square robust estimators for the same purpose is discussed in (Zhang and Katsaggelos, 1996). Robust estimation of the multi-view constraints can be done solely to guide the correspondence matching, even when used with a non-projective reconstruction method such as bundle-adjustment.

Robust methods to recover the epipolar geometry have their problems. They can cope with moderate to large inter-frame displacements and can be very effective in practice. However, they depend crucially on the ability to identify a reasonably reliable set of initial correspondences, and this becomes more and more difficult with increasing inter-frame motion. In the most general case, images are taken from widely separated viewpoints. This problem has largely been ignored in the SFM literature, due to the difficulty of the data association problem, which has been referred to as the most difficult part of structure recovery (Torr et al., 1998). Note that this is particularly challenging in 3D: traditional approaches for establishing correspondence between sets of 2D points as discussed above are of limited use in this domain, as the projected 3D structure can look very different in each image. One approach is to use image-based methods to bring the images in rough correspondence, e.g. by estimating a homography as done in (Pritchett and Zisserman, 1998b; Pritchett and Zisserman, 1998a), and then applying a RANSAC-based method. This can account for large orientation changes, e.g. switching from landscape to portrait images, but it is still not able to cope with large translations.

The most fundamental problem with methods based on multiview constraints is that they can only be formulated for two, three, or four views. The motion/correspondence recovery can then only proceed by working with batches of pairs or triples (the quadrifocal tensor is seldomly used), and stitching these sets together is, quoting (Hartley and Zisserman, 2000), "still something of a black art". The EM-based approach proposed here, in contrast, uses all the images at the same time and hence uses all of the available data instead of parsing it in chunks which then have to be stitched together somehow.

3.2 Data Association for Target Tracking

It comes as no surprise that data-association has been studied extensively in the target tracking community. Tracking is the process of recursively estimating the state of one or multiple targets, based on measurements perceived by one or multiple sensors. The process of deciding which measurements are associated with which targets is a crucial step in estimating the state of the targets. Indeed, this data-association problem is perceived by many

as the most difficult aspect of the multi-target tracking problem (Molnar and Modestino, 1998).

Textbook references for data-association in the context of target-tracking are (Bar-Shalom and Li, 1993; Popoli and Blackman, 1999), whereas (Cox, 1993) discusses the use of these techniques in the context of computer vision. For a thorough background on tracking and smoothing, see the textbooks (Jazwinsky, 1970; Maybeck, 1979; Maybeck, 1982; Bar-Shalom and Li, 1993).

3.2.1 Single-Target Tracking

Methods to solve the data-association problem in the case of tracking a single target are not of particular relevance in a computer vision context. It seldom occurs in vision that we have only one feature to work with, and many geometric estimation problems in fact depend on the existence of a minimal number of features. However, sometimes these algorithms are still used to track features in a sequence, and, more importantly, the multiple target literature was developed in many cases by extending the single target case.

Even in single-target tracking, the *optimal* data-association algorithm is intractable. The data-association problem arises when at each time step we have multiple measurements, but at most one measurement actually originates from the target, with the others being *clutter*. Typically a pre-processing step involves *gating* the measurements to exclude improbable associations from the outset. However, even if gating eliminates all but a few possible associations in each time step, the number of possible hypotheses over the entire sequence of data-associations for each time-step grows exponentially with time. An optimal but impractical algorithm would keep track of all these different hypotheses.

Tractable approximations to the optimal algorithm can be obtained by pruning and combining hypotheses, as in the optimal Bayesian filter of (Singer et al., 1974), and Reid's multiple hypothesis filter (Reid, 1979). A different and popular approach is to reason about the data-association only in the *current* time-step, and regard choices in the past as fixed. This tack is taken in the simplest of all algorithms, nearest neighbor (NN) tracking (Blackman, 1986; Li and Bar-Shalom, 1996), which simply picks the measurement closest to the predicted measurement at each time step. "Closeness" is defined in terms of Mahalanobis distance, such that the uncertainty with respect to the prediction is taken into account. A more accurate approximation is obtained by the probabilistic data association filter (PDAF) (Bar-Shalom and Tse, 1975; Bar-Shalom and Fortmann, 1988), which enumerates all possible

association hypotheses in the current time step, and then calculates for each measurement the marginal probability β_k of being associated with the target. These marginal probabilities are then used to compute a weighted measurement which is used to update the target state.

3.2.2 Multi-Target Tracking

Techniques for tracking multiple-targets with unknown data-association are very relevant in the current context. When there are multiple targets to be tracked, the combinatorics make the data-association problem still harder. This was realized very early on, in a paper by Sittler (Sittler, 1964), which foreshadowed most of the later development in data-association even before they could be implemented on computers.

A number of approaches actually represent a large number of complete data-association hypotheses over multiple time-steps. The track splitting filter (Smith and Buechler, 1975) keeps a tree of hypotheses for each target individually, and uses a maximum likelihood criterion to prune the tree. (Morefield, 1977) models the interaction of the multiple targets more accurately, and phrases the resulting data-association problem as an integer programming problem. Basically, the maximum likelihood partition of the measurements into disjoint tracks is found. Finally, Reid's multiple hypothesis tracker (MHT) (Reid, 1979) constructs a tree of all possible hypotheses, including all possible new track initiations at every time step. Reid discusses a number of strategies to prune the tree in order to achieve reasonable computation times. Several algorithms restrict the number of generated hypotheses by only considering the m-best possible assignments of measurements to targets (Danchick and Newnam, 1993; Cox and Hingorani, 1996; Cox and Miller, 1995), leading to more efficient implementations of the MHT (Cox and Hingorani, 1994).

In contrast to the MHT and related approaches, the joint probabilistic data association filter (JPDAF) (Bar-Shalom et al., 1980; Fortmann et al., 1980; Fortmann et al., 1983) only reasons about the association in the current time step, in a straightforward extension of the single-target PDAF. However, since now multiple targets have to be associated with multiple measurements, many of which may be clutter, even the combinatorics of enumerating the set of hypotheses in a single time-step can be intractable. Thus, several authors have proposed ways to approximate the calculation of the β values (the marginal association probabilities), e.g. using a Hopfield neural network (Sengupta and Iltis, 1988; Sengupta and Iltis, 1989; Hopfield and Tank, 1985), or by branch and bound type algorithms (Zhou and Bose, 1993; Zhou and Bose, 1995). Recently, a particle filtering version

of the JPDAF has been proposed in (Schulz et al., 2001).

The above multi-target tracking algorithms have been used extensively in the context of computer vision. Some examples are the use of nearest neighbor tracking in (Deriche and Faugeras, 1990), the multiple hypothesis tracker in (Cox and Leonard, 1994; Cox and Hingorani, 1994), and the JPDAF in (Rasmussen and Hager, 1998; Rasmussen and Hager, 2001).

Multi-Target Multi-Sensor Tracking

Optimal multi-sensor association problems are NP-complete, and a relaxation approach has been proposed by Pattipati and colleagues (PattiPati et al., 1990; Pattipati et al., 1992; Deb et al., 1997; Kirubarajan et al., 2001).

3.2.3 Approaches based on the EM Algorithm

In the past decade, an alternative way of approximating the multiple-hypothesis filter has been proposed, based on the expectation-maximization (EM) algorithm. Since these approaches are very related to the techniques described in this dissertation, I discuss them separately.

In a 1992 paper (Avitzour, 1992), Avitzour proposed to use the EM-algorithm to obtain the maximum likelihood state sequence, given a batch of measurements. In the E-step, a probability distribution over all possible data-associations in each time step is computed. Since this is conditioned on the current estimate of the state sequence, this can be done independently for each image. The marginal association probabilities are then used, as in the JPDAF, to create a weighted measurement that is used in the M-step to update the ML estimate of the state sequence. These two steps are iterated to convergence.

Essentially the same mechanism is used in the probabilistic multi-hypothesis tracker (PMHT) of Streit et. al. (Streit and Luginbuhl, 1994; Gauvrit et al., 1997). However, in the PMHT case the marginal probabilities are approximated by ignoring the problem of mutual exclusion, i.e. treating the assignment of measurements of targets as independent of each other. A comparison between the PMHT and the JPDAF is done in (Rago et al., 1995), and a number of improvements to the PMHT are surveyed in (Willett et al., 1999).

Molnar (Molnar and Modestino, 1998) presents a recursive version of the PMHT, where the EM algorithm is used to obtain MAP state estimate for the current time-step only, in

contrast to the original PMHT, which performs smoothing over the entire track. Of course, the PMHT can be used as a fixed lag smoother and, with zero-lag, as a pure tracking algorithm. The main contribution of (Molnar and Modestino, 1998) is a MRF formulation of the E-step, where the marginal probabilities are obtained using a mean-field approximation.

The use of EM first by Avitzour and then Streit in the PMHT is completely analogous to what is proposed in this dissertation. However, in geometric estimation problems, the "structure" is assumed fixed (unless non-rigid motion is allowed), whereas in tracking it is the sensors that are fixed and the targets that move. In addition, because in computer vision applications there are typically many features, a brute-force approach to the E-step as in Avitzour is intractable. The approximation to the E-step proposed by Streit, on the other hand, completely disregards the mutual exclusion constraint. The mean-field approximation used by Molnar will not correctly model mutual exclusion either. The introduction of sampling to implement the E-step in a MCEM scheme addresses both these problems.

A different use of the EM algorithm can be found in the EM data association (EMDA) algorithm (Pulford and Scala, 1996; Pulford and Logothetis, 1997). In that approach, the state sequence is regarded as the missing data, rather than the associations. The M-step then finds a maximum likelihood association using the Viterbi algorithm. The use of an EM-algorithm with a discrete parameter space to search over in the M-step is dubious, however: the convergence proof of EM is valid only for continuous spaces.

Finally, the Monte Carlo data association (MCDA) algorithm (Bergman and Doucet, 2000) is very similar in spirit to PMHT, in that it obtains a distribution over the unknown data-associations. However, it uses Markov chain Monte Carlo sampling to obtain this distribution, and, in a technique called Rao-Blackwellization, the state sequence for each sample is integrated out analytically. To sample, they use Gibbs sampling by drawing from the data-association probability at each time step in turn. Since the number of data-associations can be quite large, however, it is clear that this approach will not scale up to many more targets/measurements.

3.3 Data-association and Simultaneous Localization and Mapping[1]

Simultaneous Localization and Mapping (SLAM) (Dissanayake et al., 2000; Durrant-Whyte et al., 2001) in robotics, also called Concurrent Mapping and Localization (CML) (Thrun et al., 1998; Leonard and Feder, 1999), is the problem of reconstructing a robot's environment (in 2D or 3D) from a time-series of odometry and sensor measurements collected by one or more robots. Sensors that are commonly brought to bear on this task include cameras, sonar and laser range finders, radar, and GPS.

Online versions of mapping algorithms are based on extensions of well-known tracking methods, i.e. variable dimension Kalman filters (Cox, 1991; Leonard and Durrant-Whyte, 1991a; Leonard and Durrant-Whyte, 1991b; Castellanos et al., 1999; Castellanos and Tardos, 2000), and face some of the same data-association issues. In particular, in the common case where the goal is to estimate the location of landmarks, sensor readings have to be correctly associated with landmarks. For a relatively small number of landmarks, the multiple hypothesis filter (MHT, discussed above in Section 3.2.2) has been successfully applied (Cox and Leonard, 1994; Jensfelt and Kristensen, 1999; Roumerliotis and Bekey, 2000; Reuter, 2000).

In the case the data-association is known, off-line versions approaches to the SLAM problem are essentially equivalent to the structure from motion problem in vision, and similar to track smoothing. Maximum likelihood versions of this (i.e. without a smoothing prior) were implemented by Lu and Milios (Lu and Milios, 1997) and Gutmann (Gutmann and Nebel, 1997).

Off-line building of maps is considerably harder with unknown data-association, and is similar to the geometric estimation problems with unknown correspondence considered here. In this context, the EM algorithm has been suggested by Thrun and colleagues. (Thrun et al., 1998). In their version of EM, the robot robot location is considered the latent or nuisance variable, whereas the environment map is the quantity of interest. The data-association problem is implicitly solved in this approach by using a grid-based probabilistic representation of the map, which is constructed in the M-step by integrating sensor measurements over the distribution of all robot poses. A different approach that uses essentially the same EM paradigm is presented in (Liu et al., 2001), but here the environment model is parameterized in terms of a set of objects (e.g. planes corresponding to walls) which

[1]This section is based in part on an unpublished note by Sebastian Thrun.

is a much more constrained problem, and hence computationally more efficient and more accurate (where the model assumptions hold).

Chapter 4

An EM Approach to Correspondence

In this chapter, I propose the EM algorithm as a practical way to estimate structure and motion parameters, given that the correspondence information is unknown (Dellaert et al., 2001). Whereas previous approaches first single out a "best" correspondence and then estimate the structure and motion given the image measurements and correspondence, I will show that the correct MAP estimate is obtained by integrating over *all* possible correspondence vectors. When applied to estimating the alignment between 2D structures, this EM approach formalizes a number of previous algorithms such as the softassign algorithm (Rangarajan and Mjolsness, 1994; Gold and Rangarajan, 1996). In addition, because the correspondence is modeled from measurements to structure as opposed to between measurements, the algorithm automatically generalizes to multiple views. This is especially relevant when applied to the structure from motion domain, as up to now there is no multiple-view algorithm to estimate structure and motion. As discussed in Chapter 3, the state of the art proceeds via multi-view constraints that are limited to two, three, or four views.

4.1 Generalizing Structure from Motion

Whereas the EM approach below is developed within the context of structure from motion, it applies directly to all the different geometric estimation problems discussed in the previous chapter. Indeed, structure from motion can be seen as a superset of many other geometric estimation problems in computer vision and target tracking. Below I explicitly provide the connection to some important applications:

2D to 2D alignment. Applications such as image mosaicking or 2D object recognition can be seen as special cases of the SFM problem where the structure is a collection of 2D points (or lines, or both). In 2D object recognition the structure (model object) is known and the motion parameters to be recovered are 2D rigid transformations. In image mosaicking we can regard the 2D fiducial points as unknowns, and the motion to be recovered can be as simple as 2D translation or as complex as a full projective transformation. The measurement model is simply a transformation followed by additive noise, i.e. there is no projection involved.

3D to 3D alignment. This can easily be extended to the 3D case, for 3D alignment and/or 3D object recognition. In this case both the structure *and* the measurements are 3D points. The motion parameters are rigid 3D transformations.

3D to 2D alignment. In the case of 3D object recognition from 2D images, we have a known 3D structure, and a measurement model which involves a 3D rigid transformation followed by a projection (and additive noise).

(Multi-Baseline) Stereo. The connection between multi-baseline stereo and full structure from motion is simply that the motion in the former is known.

2D Structure from Motion. The 2D robot mapping problem from bearing measurements only can be seen as a special case of the SFM problem where the structure consists of 2D points (the landmarks), and the measurements are projections of the landmarks in a 1D image.

Even tracking applications can be straightforwardly modeled using the same notation and terminology we have used before for structure from motion, and the data-association problem can be handled in the same way. However, in the remainder we will focus primarily on structure from motion and its related applications in computer vision, as enumerated above.

4.2 Maximum a Posteriori Estimation

A direct approach to maximum a posteriori estimation of structure and motion *without* being given the correspondence is intractable. Recall from Chapter 2 that the MAP estimate of structure and motion Θ^* given the measurements $\mathbf{U}$ *and* the correspondence $\mathbf{J}$ is given by (equation 2.4 on page 32):

$$\Theta^* = \underset{\Theta}{\text{argmax}} \ \log P(\Theta|\mathbf{U}, \mathbf{J}) = \underset{\Theta}{\text{argmax}} \ \{\log L(\Theta; \mathbf{U}, \mathbf{J}) + \log P(\Theta)\}$$

where $P(\mathbf{\Theta})$ is a prior on structure and motion, and which can be solved for using one of the various optimization methods discussed in Section 2.4. If the correspondence $\mathbf{J}$ is unknown we cannot directly apply these methods. However, at least formally, we can still write down the MAP criterion:

$$\mathbf{\Theta}^* = \underset{\mathbf{\Theta}}{\text{argmax}} \;\; P(\mathbf{\Theta}|\mathbf{U}) = \underset{\mathbf{\Theta}}{\text{argmax}} \;\; P(\mathbf{U}|\mathbf{\Theta})P(\mathbf{\Theta}) \tag{4.1}$$

Although this might seem counterintuitive at first, equation (4.1) above states that *we can find the MAP estimate for structure and motion without explicitly reasoning about which assignment might be correct*. We "only" need to maximize the posterior $P(\mathbf{\Theta}|\mathbf{U})$, which does directly not depend on $\mathbf{J}$. Note that if we assume no prior information $P(\mathbf{\Theta})$, the above MAP criterion becomes a maximum likelihood (ML) criterion.

Although we can still frame this case as a problem of ML or MAP estimation, solving it directly is intractable due to the combinatorial nature of the data association problem. Indeed, the expression for the posterior density of structure and motion $\mathbf{\Theta}$ can be obtained by marginalizing the joint density over the space $\mathcal{J}$ of all possible correspondences $\mathbf{J}$:

$$P(\mathbf{\Theta}|\mathbf{U}) = P(\mathbf{\Theta}) \sum_{\mathbf{J} \in \mathcal{J}} P(\mathbf{U}, \mathbf{J}|\mathbf{\Theta}) \tag{4.2}$$

Unfortunately, the number of possible assignments grows combinatorially in m and n. Even if we assume there is no clutter or occlusion, there are $n!$ possible assignment vectors $\mathbf{j}_i$ in each image, yielding a total of $n!^m$ assignments $\mathbf{J}$. In summary, $P(\mathbf{\Theta}|\mathbf{U})$ is hard to obtain explicitly, as it involves summing over a combinatorial number of possible assignments.

4.3 The Expectation-Maximization Algorithm

A key insight is that we can use the well-known expectation-maximization (EM) algorithm (Hartley, 1958; Dempster et al., 1977; McLachlan and Krishnan, 1997) to find the MAP estimate $\mathbf{\Theta}^*$ for structure and motion, while regarding the correspondence information $\mathbf{J}$ as a hidden variable. EM naturally comes to mind in missing data problems such as this, and EM has been used in a similar setting in the tracking literature in (Avitzour, 1992; Streit and Luginbuhl, 1994), where it forms the basis of the Probabilistic Multiple-Hypothesis Tracker (PMHT). It has also been employed by Wells in the context of object recognition (Wells,

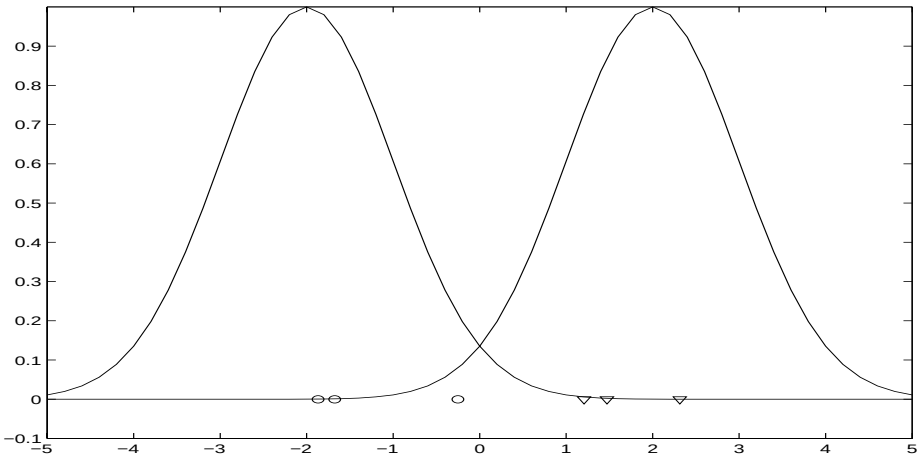


Figure 4.1: EM example: Mixture components and data. The data consists of three samples drawn from each mixture component, shown above as circles and triangles. The means of the mixture components are -2 and 2, respectively.

1997). While a direct approach to computing the posterior (4.2) is generally intractable, EM provides a practical method for finding its maxima.

The intuition behind EM is an old one (even though it is not the complete story): alternate between estimating the unknowns $\mathbf{\Theta}$ and the correspondence $\mathbf{J}$. This idea has been around for a long time (see Section 3.1 for a thorough discussion). However, instead of finding the best correspondence $\mathbf{J}$ given an estimate $\mathbf{\Theta}$ at each iteration, EM computes a *distribution* over the space of correspondences $\mathcal{J}$. In practice, only the sufficient statistics of this distribution are needed, and these can be regarded as a "soft correspondence" that is used instead of a single "best correspondence" vector. In this light, the recent 2D alignment algorithms based on softassign by Rangarajan and colleagues (Rangarajan and Mjolsness, 1994; Gold and Rangarajan, 1996; Gold et al., 1998) can be regarded as EM all but in name, albeit with a sub-optimal approximation to the E-step.

One of the most insightful explanations of EM, that provides a deeper understanding of its operation than the intuition of alternating between variables, is in terms of lower-bound maximization (Neal and Hinton, 1998; Minka, 1998). In this derivation, the E-step can be interpreted as constructing a local lower-bound to the posterior distribution, whereas the M-step optimizes the bound, thereby improving the estimate for the unknowns. This is demonstrated below for a simple example.

Consider the mixture estimation problem shown in Figure 4.1, where the goal is to estimate the two component means θ_1 and θ_2 given 6 samples drawn from the mixture, but

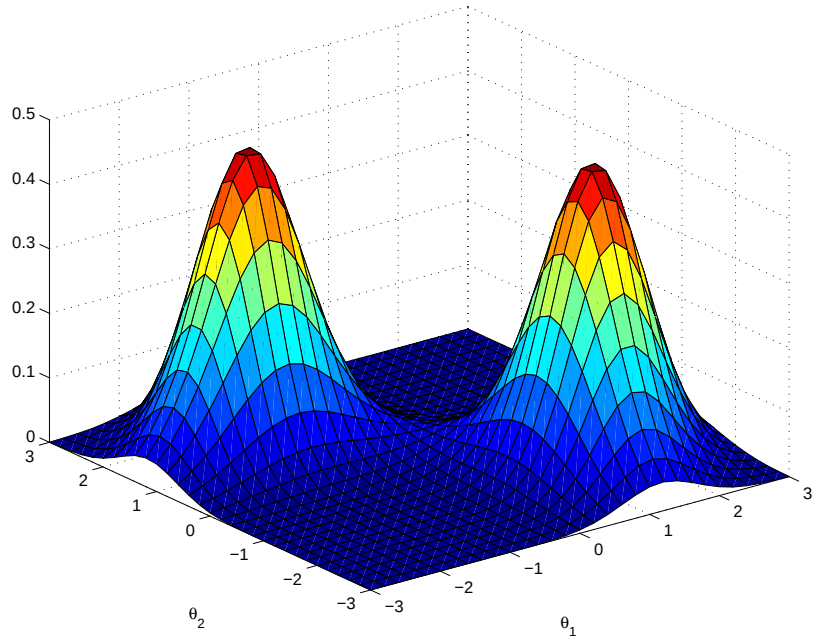


Figure 4.2: The true likelihood function of the two component means θ_1 and θ_2, given the data in Figure 4.1.

without knowing from which mixture each sample was drawn. This is analogous to the correspondence problem. The state space is two-dimensional, and the true likelihood function (corresponding to equation 4.2 on page 51) is shown in Figure 4.2. Note that there are two modes, located respectively at about $(-2, 2)$ and $(2, -2)$. This makes perfect sense, as we can switch the mixture components without affecting the quality of the solution. Note also that the true likelihood is computed by integrating over all possible data associations, and hence we can find a maximum likelihood solution without solving a correspondence problem. However, even for only 6 samples, this requires summing over the space of 64 possible data-associations in equation 4.2.

EM proceeds as follows in this example. In the E-step, a "soft" assignment is computed that assigns a posterior probability to each possible association of each individual sample. In the current example, there are 2 mixtures and 6 samples, so the computed probabilities can be represented in a 2×6 table. Given these probabilities, EM computes a tight lower bound to the true likelihood function of Figure 4.2. The bound is constructed such that it touches the likelihood function at the current estimate, and it is only close to the true likelihood in the neighborhood of this estimate. The bound and its corresponding probability table are computed in each iteration, as shown in Figure 4.3. In this case, EM was run for 5

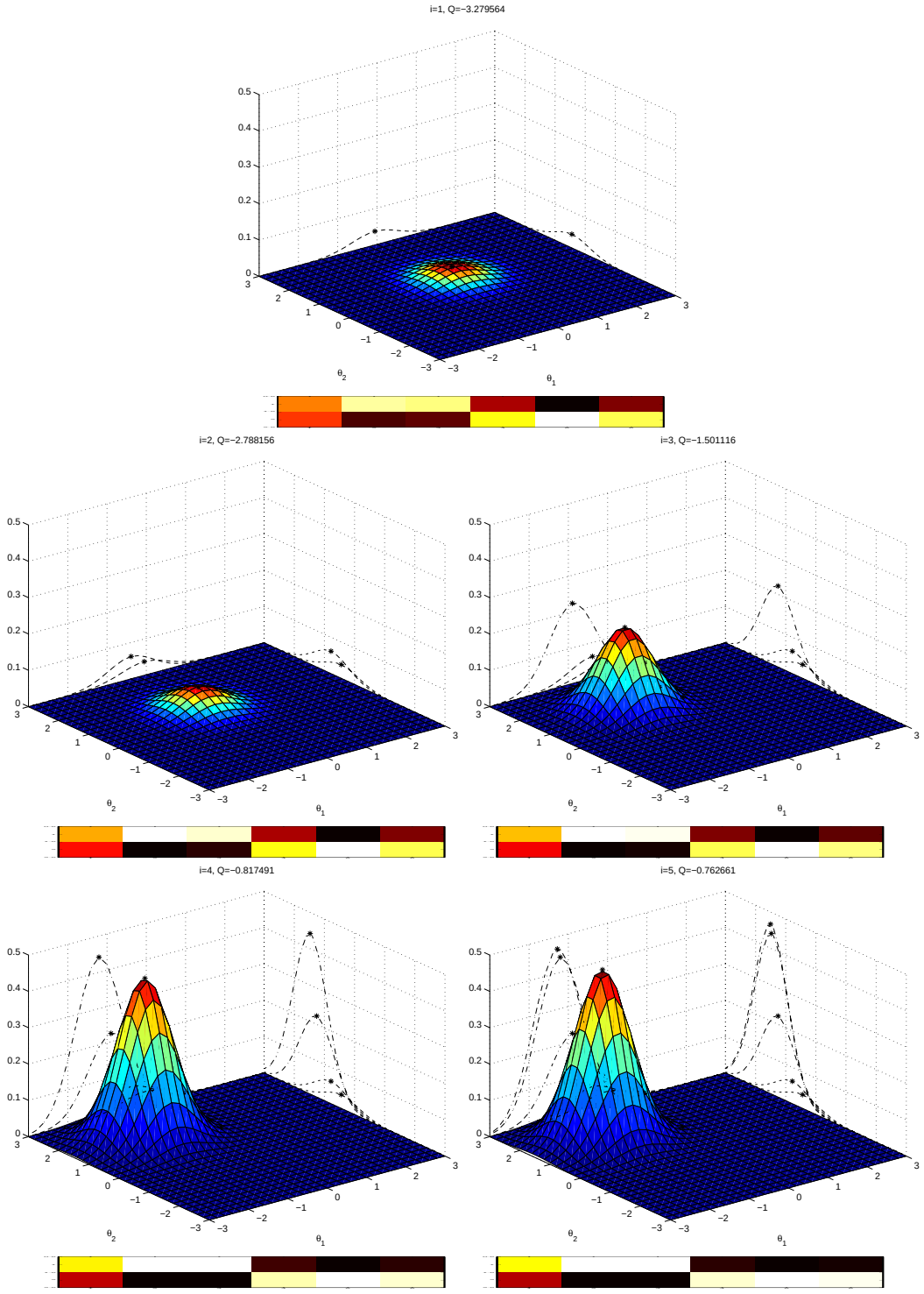


Figure 4.3: EM works by constructing successive lower bounds to a function, show above for the function in Figure 4.2. The dashed curves are projections of the 2D bound onto two axis-parallel planes that show how each iteration improves the bound. The bars under each panel show the corresponding “soft correspondence” (see text).

iterations. In the M-step, the lower bound is maximized (shown by a black asterisk in the figure), and the corresponding new estimate (θ_1, θ_2) is guaranteed to lie closer to the location of the nearest local maximum of the likelihood. Each next bound is an increasingly better approximation to the mode of the likelihood, until at convergence the bound touches the likelihood at the local maximum, and progress can no longer be made. This is shown in the last panel of Figure 4.3.

The same intuition underlies EM in the case of structure from motion. However, instead of two-dimensional, the state space will be vastly larger, as we are estimating the parameters of each feature $\mathbf{x}_j$ and all the motion parameters $\mathbf{m}_i$. As in the mixture example, we will compute a marginal probability table for the correspondence in each image, with n^2 entries: one for each measurement to feature association. The calculation of these probabilities in the E-step is more challenging than in the mixture example, however, which is why we will resort to an approximation by sampling (see below). In the M-step, we maximize the resulting bound, and make progress towards a local maximum in the space of structure and motion.

Appendix B provides a more detailed, mathematical derivation of EM, based on the lower-bound interpretation sketched above. The earliest paper on EM is (Hartley, 1958), but the seminal reference that formalized EM and provided a proof of convergence is the "DLR" paper by Dempster, Laird, and Rubin (Dempster et al., 1977). A recent book devoted entirely to EM and applications is (McLachlan and Krishnan, 1997), whereas (Tanner, 1996) is another popular and very useful reference.

4.4 An EM Approach to Correspondence

In the present application, the EM algorithm starts from an initial guess $\mathbf{\Theta}^0$ for structure and motion, and then iterates over the following steps:

1. **E-step:** Calculate the *expected log likelihood* $Q^t(\mathbf{\Theta})$ of $\mathbf{\Theta}$ given the data $\mathbf{U}$ and the hidden variables $\mathbf{J}$:

$$Q^t(\mathbf{\Theta}) = \langle \log P(\mathbf{U}|\mathbf{J}, \mathbf{\Theta}) \rangle \stackrel{\Delta}{=} \sum_{\mathbf{J} \in \mathcal{J}^n} f^t(\mathbf{J}) \log P(\mathbf{U}, \mathbf{J}|\mathbf{\Theta}) \tag{4.3}$$

   where the expectation is taken with respect to the posterior distribution $f^t(\mathbf{J}) \stackrel{\Delta}{=} P(\mathbf{J}|\mathbf{U}, \mathbf{\Theta}^t)$ over all possible assignments $\mathbf{J}$ given the data $\mathbf{U}$ and a current guess $\mathbf{\Theta}^t$ for structure and motion.

2. **M-step:** Find the maximum likelihood (ML) estimate $\mathbf{\Theta}^{t+1}$ for structure and motion, by maximizing $Q^t(\mathbf{\Theta})$:

$$\mathbf{\Theta}^{t+1} = \underset{\mathbf{\Theta}}{\text{argmax}} \; Q^t(\mathbf{\Theta}) \tag{4.4}$$

or, in case an informative prior is available, the maximum a posteriori (MAP) estimate $\mathbf{\Theta}^{t+1}$, by adding the log-prior $\log P(\mathbf{\Theta})$ to the objective function:

$$\mathbf{\Theta}^{t+1} = \underset{\mathbf{\Theta}}{\text{argmax}} \; \left[Q^t(\mathbf{\Theta}) + \log P(\mathbf{\Theta})\right] \tag{4.5}$$

4.4.1 The Expected Log-Likelihood $Q^t(\mathbf{\Theta})$

Because of the specific assumptions we can make in the structure from motion application (and other related computer vision problems), we can substantially simplify the computation of $Q^t(\mathbf{\Theta})$. The key to the efficiency of EM lies in the fact that, under certain assumptions, the expression (4.3) above contains many repeated terms, and can be rewritten as

$$Q^t(\mathbf{\Theta}) \equiv \sum_{i=1}^{m} \sum_{j=1}^{n} \sum_{k=1}^{K_i} f_{ijk}^t \log P(\mathbf{u}_{ik}|\mathbf{m}_i, \mathbf{x}_j) \tag{4.6}$$

where, as you might recall, the following symbols were defined before:

- $\mathbf{u}_{ik}$ is the k^{th} measurement in image i, with $i \in 1..m$ and $k \in 1..K_i$
- $\mathbf{m}_i$ are the motion parameters associated with image i
- $\mathbf{x}_j$ is the j^{th} structure feature, with $j \in 1..n$

and where f_{ijk}^t is the *marginal posterior probability* $P(\mathbf{j}_{ik} = j|\mathbf{U}, \mathbf{\Theta}^t)$ that measurement $\mathbf{u}_{ik}$ corresponds to feature $\mathbf{x}_j$, i.e. the probability that the correspondence indicator $\mathbf{j}_{ik}$ equals j. Thus, the marginal correspondence probability f_{ijk}^t is formally defined as

$$f_{ijk}^t \triangleq P(\mathbf{j}_{ik} = j|\mathbf{U}, \mathbf{\Theta}^t) = \sum_{\mathbf{J} \in \mathcal{J}^n} \delta(\mathbf{j}_{ik}, j) f(\mathbf{J}) \tag{4.7}$$

where δ is the Kronecker delta function, i.e. $\delta(\mathbf{j}_{ik}, j) = 1$ if $\mathbf{j}_{ik} = j$ and 0 otherwise. The intuitive explanation is as follows: through the likelihood term $\log P(\mathbf{u}_{ik}|\mathbf{m}_i, \mathbf{x}_j)$, each measurement $\mathbf{u}_{ik}$ should influence the estimation of the motion $\mathbf{m}_i$, and the feature $\mathbf{x}_j$ to which it *actually* corresponds. This is because $\log P(\mathbf{u}_{ik}|\mathbf{m}_i, \mathbf{x}_j)$ corresponds to, in the

typical SFM problem, the *reprojection error* associated with measurement $\mathbf{u}_{ik}$. However, in the present case we only have a probabilistic or "soft" assignment of measurements to features. The probability that measurement $\mathbf{u}_{ik}$ is assigned to feature $\mathbf{x}_j$ is exactly the quantity f_{ijk}^t. These probabilities are computed in the E-step, and we use them weight each associated reprojection error accordingly. Using this weighted objective function, the the motion parameters $\mathbf{m}_i$ and the structure parameters $\mathbf{x}_j$ are re-estimated in the M-step. The convergence proof of EM says that this will eventually converge to a maximum-likelihood estimate for both $\mathbf{M}$ and $\mathbf{X}$[1].

To show the validity of (4.6), let us first rewrite $\log P(\mathbf{U}, \mathbf{J}|\mathbf{\Theta})$ from equation (4.3) by applying the chain rule:

$$\log P(\mathbf{U}, \mathbf{J}|\mathbf{\Theta}) = \log P(\mathbf{U}|\mathbf{J}, \mathbf{\Theta}) + \log P(\mathbf{J}|\mathbf{\Theta})$$

Here $\log P(\mathbf{U}|\mathbf{J}, \mathbf{\Theta})$ corresponds to the total reprojection error, given a specific correspondence vector $\mathbf{J}$, and $\log P(\mathbf{J}|\mathbf{\Theta})$ is the log of a prior distribution over correspondences $\mathbf{J}$. The first term is exactly the one minimized in typical structure from motion algorithms: by varying $\mathbf{\Theta}$ we try to minimize the reprojection error. Less intuitive is that the prior term $\log P(\mathbf{J}|\mathbf{\Theta})$ can also influence the optimal estimate $\mathbf{\Theta}^*$ for structure and motion, depending on how occlusion and clutter are modeled (see Chapter 7).

However, below we make the assumption that the prior $P(\mathbf{J}|\mathbf{\Theta})$ does not depend on $\mathbf{\Theta}$, i.e. $P(\mathbf{J}|\mathbf{\Theta}) = P(\mathbf{J})$. In that case the only term of interest in $Q^t(\mathbf{\Theta})$ is the expected image likelihood $\langle \log P(\mathbf{U}|\mathbf{J}, \mathbf{\Theta}) \rangle$. Using the symbol "$\equiv$" to denote equality up to a constant, we can then simplify the expected log-likelihood (4.3) as follows:

$$Q^t(\mathbf{\Theta}) \equiv \langle \log P(\mathbf{U}|\mathbf{J}, \mathbf{\Theta}) \rangle = \sum_{\mathbf{J} \in \mathcal{J}^n} f^t(\mathbf{J}) \log P(\mathbf{U}|\mathbf{J}, \mathbf{\Theta}) \tag{4.8}$$

Note that the assumption $P(\mathbf{J}|\mathbf{\Theta}) = P(\mathbf{J})$ is no longer valid if a sophisticated model is used to model occlusion and clutter (see Chapter 7).

The second assumption we make is that of conditional independence of the measurements $\mathbf{u}_{ik}$. First, assume that the respective images $\mathbf{U}_i$ are conditionally independent of each other *given* the structure and motion parameters $\mathbf{\Theta}$ and the correspondence vectors $\mathbf{j}_i$. In that case, the image likelihood function factors over the different images:

$$P(\mathbf{U}|\mathbf{J}, \mathbf{\Theta}) = \prod_{i=1}^{m} P(\mathbf{U}_i|\mathbf{j}_i, \mathbf{m}_i, \mathbf{X}) \tag{4.9}$$

[1] In fact, the weighted objective function is exactly the lower-bound computed in the E-step and maximized in the M-step

In addition, if the individual image measurements $\mathbf{u}_{ik}$ are conditionally independent of each other given $\mathbf{j}_{ik}$, $\mathbf{m}_i$, and the structure $\mathbf{X}$, we get:

$$P(\mathbf{U}_i|\mathbf{j}_i, \mathbf{m}_i, \mathbf{X}) = \prod_{k=1}^{K_i} P(\mathbf{u}_{ik}|\mathbf{j}_{ik}, \mathbf{m}_i, \mathbf{X}) \tag{4.10}$$

Invoking these conditional independence assumptions (equations 4.9 and 4.10), we obtain the factored expression:

$$Q^t(\mathbf{\Theta}) \equiv \sum_{\mathbf{J}\in\mathcal{J}^n} f^t(\mathbf{J}) \sum_{i=1}^{m}\sum_{k=1}^{K_i} \log P(\mathbf{u}_{ik}|\mathbf{j}_{ik}, \mathbf{m}_i, \mathbf{X}) \tag{4.11}$$

Using a standard trick from the EM literature we can express this by means of the marginal probabilities f_{ijk}^t:

$$Q^t(\mathbf{\Theta}) \equiv \sum_{i=1}^{m}\sum_{j=0}^{n}\sum_{k=1}^{K_i} f_{ijk}^t \log P(\mathbf{u}_{ik}|j, \mathbf{m}_i, \mathbf{X}) \tag{4.12}$$

the correctness of which is most easily noted by plugging in the definition of the f_{ijk}^t (definition 4.7 on page 56) into the above, which yields back (4.11).

Finally, we can eliminate spurious measurements from consideration. On the assumption that the likelihood $P(\mathbf{u}_{ik}|\mathbf{j}_{ik} = 0)$ of spurious measurements does not depend on the structure and motion $\mathbf{\Theta}$, the terms for which $j = 0$ in (4.12) are constant with respect to $\mathbf{\Theta}$. $Q^t(\mathbf{\Theta})$ can thus be written as a sum of non-spurious likelihood terms only:

$$Q^t(\mathbf{\Theta}) \equiv \sum_{i=1}^{m}\sum_{j=1}^{n}\sum_{k=1}^{K_i} f_{ijk}^t \log P(\mathbf{u}_{ik}|\mathbf{m}_i, \mathbf{x}_j)$$

The difference is that (a) the summation over j is now from 1 to n rather than from 0, and (b) we have used the fact that, for $j \neq 0$, we can write $P(\mathbf{u}_{ik}|j, \mathbf{m}_i, \mathbf{X}) = P(\mathbf{u}_{ik}|\mathbf{m}_i, \mathbf{x}_j)$, further simplifying the likelihood terms so that they are a function of one specific structure element only.

Note that the form (4.6) does *not* depend on the assumption of Gaussian noise, but rather on the conditional independence of the image measurements. Note also that a similar trick cannot be applied to the "naive" expression (4.2) for the posterior, as the latter is a sum of probabilities, not *log*-likelihoods.

4.4.2 The M-step and Virtual Measurements

In this section we show that, in the most common case, the M-step can be implemented in a simple and intuitive way. Recall that in the M-step, we re-estimate the structure and motion by minimizing the expected log-likelihood $Q^t(\mathbf{\Theta})$, i.e. equation 4.6 on page 56. When a Gaussian noise model is used, $Q^t(\mathbf{\Theta})$ can be rewritten *such that the M-step amounts to solving a structure from motion problem of the same size as before*, but using as input a newly synthesized set of virtual measurements, created in the E-step. The concept of using synthetic measurements is not new. It is also used in the tracking literature, where EM is used to perform track smoothing (Avitzour, 1992; Streit and Luginbuhl, 1994).

Consider the common case where the measurement model for $\mathbf{u}_{ik}$ can be written as the application of a (possibly non-linear) *measurement function* $\mathbf{h}(.,.)$ plus additive, zero-mean Gaussian noise with covariance matrix $\mathbf{R}_{ik}$. In that case, the conditional probability density for a single measurement is:

$$P(\mathbf{u}_{ik}|\mathbf{m}_i, \mathbf{x}_j) = \frac{1}{\sqrt{|2\pi\mathbf{R}_{ik}|}} \exp\left[-\frac{1}{2}(\mathbf{u}_{ik} - \mathbf{h}_{ij})^T \mathbf{R}_{ik}^{-1}(\mathbf{u}_{ik} - \mathbf{h}_{ij})\right] \tag{4.13}$$

where we assume $j \neq 0$ and we define $\mathbf{h}_{ij} \stackrel{\Delta}{=} \mathbf{h}(\mathbf{m}_i, \mathbf{x}_j)$ for notational convenience.

The main point to be made in this section is this: it can be shown by simple algebraic manipulation that in that case $Q^t(\mathbf{\Theta})$ (equation 4.6) can be written as the sum of a constant that does not depend on $\mathbf{\Theta}$, and a new re-projection error of n features in m images

$$Q^t(\mathbf{\Theta}) \equiv -\frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{n}(\mathbf{v}_{ij}^t - \mathbf{h}_{ij})^T \mathbf{R}_{ij}^{-1}(\mathbf{v}_{ij}^t - \mathbf{h}_{ij}) \tag{4.14}$$

where the *virtual measurements* $\mathbf{v}_{ij}^t$ are defined as

$$\mathbf{v}_{ij}^t \stackrel{\Delta}{=} \mathbf{R}_{ij}\sum_{k=1}^{K_i} f_{ijk}^t \mathbf{R}_{ik}^{-1}\mathbf{u}_{ik} \tag{4.15}$$

with the *virtual measurement covariance* $\mathbf{R}_{ij}$ defined by

$$\mathbf{R}_{ij}^{-1} \stackrel{\Delta}{=} \sum_{k=1}^{K_i} f_{ijk}^t \mathbf{R}_{ik}^{-1} \tag{4.16}$$

Thus, each virtual measurement $\mathbf{v}_{ij}^t$ is simply a weighted average of the original measurements $\mathbf{u}_{ik}$ in the image. Intuitively, $\mathbf{R}_{ik}^{-1}$ is a measure for how much information is given by

the measurement $\mathbf{u}_{ik}$, and $\mathbf{v}_{ij}^t$ is a quantity in which the original measurements contribute according to their information content. The inverse virtual measurement covariance $\mathbf{R}_{ij}^{-1}$ encodes how much information each virtual measurement $\mathbf{v}_{ij}^t$ contributes in the estimation of the unknown structure and motion $\mathbf{\Theta}$.

The proof of the equivalence of (4.6) and (4.14) involves only algebraic manipulation and can be found in Appendix C. More important is the intuitive interpretation that stems from it, recapitulated below.

4.4.3 An Intuitive Interpretation of EM

The important point is that the M-step objective function (4.14) above, arrived at by assuming *unknown* correspondence, is of exactly the same form as the objective function for the SFM problem with *known* correspondence. As a consequence, *any of the existing SFM methods, of which many are discussed in Section 2.4, can be used to implement the M-step.* This provides an intuitive interpretation for the overall algorithm:

1. **E-step:** Calculate the weights f_{ijk}^t from the distribution over assignments. Then, in each of the m images calculate n virtual measurements $\mathbf{v}_{ij}^t$.
2. **M-step:** Solve a conventional SFM problem using the virtual measurements as input.

In other words, the E-step synthesizes new measurement data, and the M-step is implemented using conventional SFM methods. What is left is to show how the E-step can be implemented.

Other geometric estimation problems in vision, such as 2D-2D, 2D-3D, and 3D-3D alignment, as well as (sparse) multi-view stereo, can all be handled as special cases of the structure from motion problem.

4.4.4 Isotropic Gaussian Noise

For i.i.d. isotropic Gaussian noise, i.e. where the noise is distributed in a radially symmetric way, we can further simplify the virtual measurements formulation (4.14). In that case, the covariance matrix $\mathbf{R}_{ik} = \sigma^2 I_{2\times 2}$, and the virtual measurement equations (4.15) and (4.16) simplify considerably

$$\mathbf{R}_{ij} \triangleq \frac{1}{\sum_{k=1}^{K_i} f_{ijk}^t} \sigma^2 I_{2\times 2}$$

$$\mathbf{v}_{ij}^t \triangleq \frac{1}{\sum_{k=1}^{K_i} f_{ijk}^t} \sum_{k=1}^{K_i} f_{ijk}^t \mathbf{u}_{ik}$$

i.e. the virtual measurements are simple weighted averages of the original measurements. The expected log-likelihood is then a weighted sum of squared errors:

$$Q^t(\mathbf{\Theta}) \equiv -\frac{1}{2\sigma^2} \sum_{i=1}^{m} \sum_{j=1}^{n} \left(\sum_{k=1}^{K_i} f_{ijk}^t \right) \|\mathbf{v}_{ij}^t - \mathbf{h}_{ij}\|^2$$

4.4.5 Occlusion and Spurious Measurements

The derivation of the virtual features and variances also accommodates occluded features and spurious measurements. The influence of occlusion can be understood as follows: for features $\mathbf{x}_j$ that are have a high probability of being occluded in image i the total probability mass $\sum_{k=1}^{K_i} f_{ijk}^t$ of being associated with any of the measurements $\mathbf{u}_{ik}$ will be low. In that case, the corresponding squared error term $\|\mathbf{v}_{ij}^t - \mathbf{h}_{ij}\|^2$ will not be important in the calculation of $Q^t(\mathbf{\Theta})$. Likewise, if a measurement $\mathbf{u}_{ik}$ has a high probability of being spurious, its influence on the re-estimation process of $\mathbf{\Theta}$ is diminished, as its contribution to the virtual measurements will be diminished.

In the special case that there are no spurious measurements we have $\sum_{k=1}^{K_i} f_{ijk}^t = \sum_{k=0}^{K_i} f_{ijk}^t = 1$, and we can further simplify by dropping the normalization factors:

$$\mathbf{v}_{ij}^t = \sum_{k=1}^{n} f_{ijk}^t \mathbf{u}_{ik} \quad \text{and} \quad \mathbf{R}_{ij} = \mathbf{R}_{ik} = \sigma^2 I_{2\times 2}$$

and the log-likelihood (4.14) becomes simply the sum of squared re-projection errors with respect to the virtual measurements:

$$Q^t(\mathbf{\Theta}) \equiv -\frac{1}{2\sigma^2} \sum_{i=1}^{m} \sum_{j=1}^{n} \|\mathbf{v}_{ij}^t - \mathbf{h}_{ij}\|^2$$

4.4.6 Markov Chain Monte Carlo and the E-step

The previous section showed that, when given the virtual measurements, the M-step can be implemented using any known SFM approach. As a consequence, we need only concern ourselves with the implementation of the E-step. In particular, we need to calculate the marginal probabilities $f_{ijk}^t = P(\mathbf{j}_{ik} = j | \mathbf{U}, \mathbf{\Theta}^t)$ needed to calculate the virtual measurements $\mathbf{v}_{ij}^t$ and covariance $\mathbf{R}_{ij}$.

Unfortunately, due to the *mutual exclusion* constraint an analytic expression for the sufficient statistics f_{ijk}^t is hard to obtain. Assuming conditional independence of the assignments $\mathbf{j}_i$ in each image, we can factor $f^t(\mathbf{J})$ as:

$$f^t(\mathbf{J}) = P(\mathbf{J}|\mathbf{U}, \mathbf{\Theta}^t) = \prod_{i=1}^{m} P(\mathbf{j}_i|\mathbf{U}_i, \mathbf{\Theta}^t)$$

where $\mathbf{U}_i$ are the measurements in image i. Applying Bayes law, we have:

$$P(\mathbf{j}_i|\mathbf{U}_i, \mathbf{\Theta}^t) \propto P(\mathbf{j}_i|\mathbf{\Theta}^t) P(\mathbf{U}_i|\mathbf{j}_i, \mathbf{\Theta}^t) \tag{4.17}$$

The second factor, the likelihood of the image correspondence vector $\mathbf{j}_i$, can easily be evaluated once a specific measurement model is assumed. However, the prior probability $P(\mathbf{j}_i|\mathbf{\Theta}^t)$ of an assignment $\mathbf{j}_i$ encodes the knowledge we have about mutual exclusion: if a measurement $\mathbf{u}_{ik}$ has been assigned $\mathbf{j}_{ik} = j$, then no other measurement in the same image should be assigned the same feature point $\mathbf{x}_j$. While it is easy to *evaluate* the posterior probability $f_i^t(\mathbf{j}_i)$ for any given assignment $\mathbf{j}_i$ through (4.17), a closed form expression for f_{ijk}^t that incorporates this mutual exclusion constraint is not available.

The solution proposed in the next chapter, Chapter 5, is to approximate the E-step by *sampling* from the posterior probability distribution $f_i^t(\mathbf{j}_i)$ over valid assignments vectors $\mathbf{j}_i$. The use of sampling has the benefit of being able to approximate the correct E-step up to arbitrary resolution, taking into account all mutual exclusion constraints.

Formally this can be justified in the context of a *Monte Carlo EM* or MCEM, a version of the EM algorithm where the E-step is executed by a Monte-Carlo process (Tanner, 1996; McLachlan and Krishnan, 1997). For now, assume a sample $\{\mathbf{j}_i^r\}$ from the true distribution $f_i^t(\mathbf{j}_i)$ is available. To compute the virtual measurements in (4.15), we need to compute the marginal probabilities f_{ijk}^t. Approximating the marginal probabilities f_{ijk}^t when given a sample $\{\mathbf{j}_i^r\}$ is straightforward:

$$f_{ijk}^t \approx \frac{1}{R} \sum_{r=1}^{R} \delta(\mathbf{j}_{ik}^r, j) \tag{4.18}$$

Note that this can be done without explicitly storing the samples, by keeping running counts of how many times each measurement $\mathbf{u}_{ik}$ is assigned to feature j.

The detailed explanation of how one can obtain a sample from $f_i^t(\mathbf{j}_i)$ is postponed until Chapter 5. It is done using a Markov chain Monte Carlo (MCMC) sampling method, in particular the Metropolis-Hastings algorithm.

4.5 Summary of the Algorithm

The inputs to the algorithm are:

- The number of features n.
- The measurements $\mathbf{U} = \{\{\mathbf{u}_{ik} | k \in 1..K_i\} | i \in 1..m\}$, where m is the number of images and K_i is the number of measurements in each image.
- The measurement covariances $\mathbf{R}_{ik}$ for each measurement $\mathbf{u}_{ik}$.

Note that the only information we have about the measurements is the image in which they were recorded.

The output of the algorithm is:

- A locally optimal structure and motion estimate $\mathbf{\Theta}^* \stackrel{\Delta}{=} (\mathbf{M}^*, \mathbf{X}^*)$, a local maximizer of the posterior probability distribution $P(\mathbf{\Theta}|\mathbf{U})$.

The pseudo-code for the final algorithm is as follows:

1. Generate an initial structure and motion estimate $\mathbf{\Theta}^0$, e.g. at random.
2. Given $\mathbf{\Theta}^t$ and the data $\mathbf{U}$, run the Metropolis-Hastings sampler in each image (Chapter 5) to obtain approximate values for the weights f_{ijk}^t (equation 4.18).
3. Calculate the virtual measurements $\mathbf{v}_{ij}^t$ and covariances $\mathbf{R}_{ij}$ using equations (4.15) and (4.16).
4. Find the new estimate $\mathbf{\Theta}^{t+1}$ for structure and motion using the virtual measurements $\mathbf{v}_{ij}^t$ as data, and the virtual covariance matrices $\mathbf{R}_{ij}$ as their noise models. This can be done using any SFM method discussed in Section 2.4.
5. If not converged, return to step 2.

4.6 Dealing with Local Minima

One significant disadvantage of EM is that is only guaranteed to converge to a *local* maximum of the likelihood function, not to a global maximum. This is especially problematic in the structure from motion application, where bad initial estimates for structure and motion can be locked in by incorrect correspondences, and vice versa.

The main strategy to avoid local minima is the use of deterministic annealing, in conjunction with random restarts if the algorithm fails to converge.

In deterministic annealing we increase the noise parameter σ in early iterations, gradually decreasing it to its correct value. This has two beneficial consequences. First, the posterior distribution $f_i^t(\mathbf{j}_i)$ is less peaked when σ is high, allowing the MCMC sampler to explore the space of assignments $\mathbf{j}_i$ more easily. Second, the expected log-likelihood $Q^t(\mathbf{\Theta})$ is smoother and has fewer local maxima for higher values of σ.

If the algorithm still does not converge, we can restart it with different initial conditions. It is easy to detect when a local minimum is reached based on the expected value of the residual, as it obeys a known χ^2 distribution. If this occurs, the algorithm is restarted with different initial conditions, until eventually successful.

Chapter 5

Sampling Weighted Assignments

This chapter explains how Markov chain Monte Carlo sampling can be used to approximate a distribution over correspondence assignments. This can then be used to approximate the E-step in the MCEM-based approach to correspondence discussed previously in Chapter 4.

In this chapter, the following assumptions will be made to simplify the problem:

1. There are no occlusions, i.e. all features are seen exactly once in all images.
2. There are no spurious measurements.

This has the following implications:

- The number of measurements in all images is equal to n, i.e., $K_i = n, \forall i$.
- A valid correspondence vector $\mathbf{J}$ now consists of m permutations of the indices $1..n$. In other words, each of the vectors $\mathbf{j}_i$ defines an *assignment* from the measurements $\mathbf{u}_{ik}$ to the features $\mathbf{x}_j$.

Clearly, the assumption that there is no occlusion or clutter is restrictive. However, there might be applications in which these assumptions hold, e.g. because occlusion of features does not occur and the feature extraction process is easy, or made easy by instrumenting the environment. In addition, modeling visibility issues can be quite involved, which would obscure the exposition below. Because of this, a detailed discussion of visibility modeling is postponed until Chapter 7, where the assumptions above will be relaxed.

5.1 Mutual Exclusion and the E-step

Recall that we are interested in estimating the structure and motion $\mathbf{\Theta}$ given only the measurements $\mathbf{U}$ but not the correspondence $\mathbf{J}$. In the previous chapter, the EM algorithm was proposed as a tractable way to find a local maximum of the posterior distribution $P(\mathbf{\Theta}|\mathbf{U})$, and it was shown that, when given the virtual measurements, the M-step can be implemented using any known structure from motion algorithm. As a consequence, we need only concern ourselves with the implementation of the E-step. In particular, we need to calculate the marginal probabilities

$$f_{ijk}^t \stackrel{\Delta}{=} P(\mathbf{j}_{ik} = j|\mathbf{U}, \mathbf{\Theta}^t) = \sum_{\mathbf{J}\in\mathcal{J}^n} \delta(\mathbf{j}_{ik}, j) f^t(\mathbf{J}) \tag{5.1}$$

i.e. the probability that the measurement $\mathbf{u}_{ik}$ corresponds to feature $\mathbf{x}_j$, given the estimate $\mathbf{\Theta}^t$ calculated in the previous M-step. These marginal probabilities can then be used to calculate the virtual measurements $\mathbf{v}_{ij}^t$ and virtual covariances $\mathbf{R}_{ij}$ using equations (4.15) and (4.16). Since this all that is needed to re-estimate $\mathbf{\Theta}$ (yielding $\mathbf{\Theta}^{t+1}$), the f_{ijk}^t play the role of *sufficient statistics* for the E-step.

Unfortunately, due to the mutual exclusion constraint an analytic expression for the sufficient statistics f_{ijk}^t is hard to obtain. To see this, first note that the posterior distribution $f^t(\mathbf{J})$ over correspondences can be factored over the different images:

$$f^t(\mathbf{J}) \stackrel{\Delta}{=} P(\mathbf{J}|\mathbf{U}, \mathbf{\Theta}^t) \propto \prod_{i=1}^{m} P(\mathbf{j}_i) P(\mathbf{U}_i|\mathbf{j}_i, \mathbf{m}_i^t, \mathbf{X}^t)$$

where $\mathbf{U}_i$ are the measurements in image i. Here we applied Bayes law and we tacitly assumed that the correspondences in the different images are *a priori* independent of each other and of the geometry $\mathbf{\Theta}^t$. In Chapter 7 we will see that this is not always a valid assumption, but in the absence of occlusion and clutter this assumption holds. As a consequence, the marginal probabilities f_{ijk}^t associated with measurements $\mathbf{u}_{ik}$ in the i^{th} image only depend on the posterior distribution $f^t(\mathbf{j}_i)$ for the image correspondence vector $\mathbf{j}_i$ (and we need no longer consider the joint posterior $PJ(\mathbf{J})$):

$$f_{ijk}^t \stackrel{\Delta}{=} P(\mathbf{j}_{ik} = j|\mathbf{U}_i, \mathbf{m}_i^t, \mathbf{X}^t) = \sum_{\mathbf{j}_i\in\mathcal{J}_i^n} \delta(\mathbf{j}_{ik}, j) f_i^t(\mathbf{j}_i) \tag{5.2}$$

where

$$f_i^t(\mathbf{j}_i) \stackrel{\Delta}{=} P(\mathbf{j}_i|\mathbf{U}_i, \mathbf{m}_i^t, \mathbf{X}^t) \propto P(\mathbf{j}_i) P(\mathbf{U}_i|\mathbf{j}_i, \mathbf{m}_i^t, \mathbf{X}^t) \tag{5.3}$$

It is here that the mutual exclusion constraint rears its ugly head. The second factor in (5.3), the likelihood $P(\mathbf{U}_i|\mathbf{j}_i, \mathbf{\Theta}^t)$, can easily be evaluated once a specific measurement model is assumed. However, the prior probability $P(\mathbf{j}_i)$ of a correspondence vector $\mathbf{j}_i$ encodes the knowledge we have about the structure from motion domain: if a measurement $\mathbf{u}_{ik}$ has been assigned $\mathbf{j}_{ik} = j$, then no other measurement in the same image should be assigned the same feature point $\mathbf{x}_j$. In other words, if we assume that *valid* correspondence vectors are all equally likely, the prior probability of $\mathbf{j}_i$ is

$$P(\mathbf{j}_i) = \begin{cases} \frac{1}{n!} & \text{if } \mathbf{j}_i \text{ is a valid assignment} \\ 0 & \text{otherwise} \end{cases} \tag{5.4}$$

While it is easy to evaluate the posterior probability $f(\mathbf{j}_i)$ for any *given correspondence vector* $\mathbf{j}_i$ through (5.3), this *global* constraint makes it difficult to express analytically either the posterior f_i or the marginals f_{ijk}.

5.2 Ways to Approximate the E-step

In this section I survey the different ways in which the marginal probabilities f_{ijk}^t can be approximated, given that there is no easy analytical expression available.

One can of course exactly compute the factors f_{ijk}^t through brute force enumeration of all possible correspondences, as done for example in the target tracking literature in (Avitzour, 1992). This is only feasible, however, in the case that n is relatively small, e.g. $n \leq 5$. This assumption may hold when tracking relatively few objects, but it typically does not hold in feature-based computer vision applications. If occlusion and clutter are not allowed, there are $n!$ possible correspondences in each image. The number of candidate correspondences grows even faster with n if occlusion and clutter are modeled, as will be discussed in Chapter 7.

An alternative is to approximate the E-step by neglecting mutual exclusion constraints altogether, as done in (Durbin and Willshaw, 1987; Durbin et al., 1989; Szeliski, 1989; Streit and Luginbuhl, 1994; Wells, 1997). In this case, the f_{ijk}^t factors are simply a function of the Mahalanobis distance between the predicted and actual measurements. An application where this holds, for example, is estimating the parameters of a mixture distribution (e.g. in clustering applications). This approach is not feasible in the context of structure from motion, as mutual exclusion provides an important constraint on the allowable solutions. If this information is not used, degenerate solutions are almost always obtained.

Finally, another approximate E-step can be obtained by a *mean-field* approximation. If the correspondence between features and measurements is arranged in a binary adjacency matrix, it satisfies two way constraints on the rows and columns. Specifically, they add up to one, creating a *doubly stochastic* matrix. The intuition behind a mean-field approximation is to construct a continuous-valued matrix that obeys these two way constraints, and that is taken to approximate the *mean field* of the binary assignment variables, in other words: the marginal probabilities f_{ijk}^t. These continuous variables are then optimized over as to best approximate the true mean field, using methods that can be traced back to statistical physics. Early work on this approach was done in (Yuille et al., 1991; Kosowsky and Yuille, 1994; Rangarajan and Mjolsness, 1994), leading to the "invisible hand" algorithm (named after an analogy with economic theories), and the "softassign" algorithm. Later work on the softassign algorithm (Gold and Rangarajan, 1996; Rangarajan et al., 1997; Gold et al., 1998; Chui and Rangarajan, 2000) introduced a graduated convexity strategy to avoid local minima (as is done below, as well), and implemented the algorithm in terms of Sinkhorn's algorithm to obtain doubly stochastic matrices (Sinkhorn, 1964). In the tracking literature, a mean-field approximation to a Markov random field (MRF) model of the data-association problem is presented in (Molnar and Modestino, 1998).

The brute-force method is intractable, and both the "mixture" and the mean-field approximations to the E-step cannot accurately model the mutual exclusion constraint. In the next section, another well known way of approximating a distribution (and hence its marginals) is proposed for the correspondence domain: sampling.

5.3 Markov Chain Monte Carlo and the E-step

The solution proposed in this dissertation is to approximate the E-step is to *sample* from the posterior probability distribution $f_i^t(\mathbf{j}_i)$ over valid assignments vectors $\mathbf{j}_i$. The use of sampling has the benefit of being able to approximate the correct E-step up to arbitrary resolution, taking into account all mutual exclusion constraints. Indeed, any other constraint on the range of allowable correspondences can be readily accommodated, e.g. ordering constraints in stereo.

Assume a sample $\{\mathbf{j}_i^r | r \in 1..R\}$ from the distribution $f_i^t(\mathbf{j}_i)$ is available. To compute the virtual measurements in (4.15), we need to compute the marginal probabilities f_{ijk}^t. As already discussed in section 4.4.6, a Monte Carlo estimate for the marginal probabilities

can easily be obtained by:

$$f_{ijk}^t \approx \frac{1}{R}\sum_{r=1}^{R}\delta(\mathbf{j}_{ik}^r, j) = \frac{1}{R}C_{ijk} \tag{5.5}$$

where R is the number of samples and the $C_{ijk} \triangleq \sum_{r=1}^{R}\delta(\mathbf{j}_{ik}^r, j)$ are defined to be the cumulative counts for each of the possible associations. Recall that δ is the Kronecker delta, with $\delta(\mathbf{j}_{ik}^r, j) = 1$ iff $\mathbf{j}_{ik}^r = 1$. Note that the estimate (5.5) can be computed without explicitly storing the samples, by incrementally updating the running counts C_{ijk} of how many times each measurement $\mathbf{u}_{ik}$ is assigned to feature j.

To sample from arbitrary distributions we can use the Metropolis-Hastings algorithm, a Markov Chain Monte Carlo method (MCMC) (Neal, 1993; Gilks et al., 1996; Doucet et al., 2001). In the present case, the *target distribution* of the sampler is the posterior distribution $f_i^t(\mathbf{j}_i)$ over correspondence vectors $\mathbf{j}_i$ in image i. Formally, this can be justified in the context of a *Monte Carlo EM* or MCEM, a version of the EM algorithm where the E-step is executed by a Monte-Carlo process (Tanner, 1996; McLachlan and Krishnan, 1997). Independently, MCMC has also been applied to data-association in (Pasula et al., 1999), albeit in a different context. Gibbs sampling, an alternative MCMC sampling method, has been applied to the data-association in tracking (Bergman and Doucet, 2000), but their method requires a brute-force enumeration over all possible associations in a single time step.

MCMC methods can be used to obtain approximate values for expectations over distributions that defy easy analytical solutions. All MCMC methods work in a similar way: they generate a sequence of *states*, in our case the correspondences $\mathbf{j}_i$, with the property that the collection of generated correspondence vectors $\mathbf{j}_i^r$ approximates a sample from the target distribution $f_i^t(\mathbf{j}_i)$. To accomplish this, a *Markov chain* is defined over the space of correspondence vectors $\mathbf{j}_i$, i.e., a transition probability matrix is specified that gives the probability of transitioning from any given correspondence vector $\mathbf{j}_i$ to any other. The transition probabilities are set up in a very specific way, however, such that the *stationary distribution* of the Markov chain is exactly the target distribution $f_i^t(\mathbf{j}_i)$. This guarantees that, if we run the chain for a sufficiently long time and then start recording states, these states constitute a sample from the target distribution. Note that while neighboring samples in the sequence are strongly correlated, the sample taken as a whole will be a true sample from the distribution after the sampler has converged.

The Metropolis-Hastings (MH) algorithm (Hastings, 1970; Metropolis et al., 1953) is one way to simulate a Markov chain with the correct stationary distribution, without explicitly

building the full transition probability matrix (which would be an intractable, given the combinatorial nature of the space). In our case, we use it to generate a sequence of R samples $\mathbf{j}_i^r$ from the target distribution $f_i^t(\mathbf{j}_i)$. The pseudo-code for the MH algorithm is as follows (adapted from (Gilks et al., 1996)):

1. Start with a valid initial correspondence vector $\mathbf{j}_i^0$.
2. Propose a new correspondence vector using the *proposal density* $g(\mathbf{j}_i'; \mathbf{j}_i^r)$.
3. Calculate the *acceptance ratio*
$$a = \frac{f_i^t(\mathbf{j}_i')}{f_i^t(\mathbf{j}_i^r)} \frac{g(\mathbf{j}_i^r; \mathbf{j}_i')}{g(\mathbf{j}_i'; \mathbf{j}_i^r)} \tag{5.6}$$
where $f_i^t(\mathbf{j}_i)$ is the *target distribution.*
4. **If** $a >= 1$ then accept $\mathbf{j}_i'$, i.e., we set $\mathbf{j}_i^{r+1} = \mathbf{j}_i'$.
Otherwise, accept $\mathbf{j}_i'$ with probability $\min(1, a)$. If the proposal is rejected, then we keep the previous sample, i.e., we set $\mathbf{j}_i^{r+1} = \mathbf{j}_i^r$.

Intuitively, step 2 proposes "moves" in state space, generated according to a probability distribution $g(\mathbf{j}_i'; \mathbf{j}_i^r)$ which is fixed in time but can depend on the current state $\mathbf{j}_i^r$. The calculation of a and the acceptance mechanism in steps 3 and 4 have the effect of modifying the transition probabilities of the chain such that its stationary distribution is exactly $f_i^t(\mathbf{J})$.

The MH algorithm easily allows incorporating the mutual exclusion constraint: if a correspondence vector $\mathbf{j}_i'$ is proposed that violates the constraint, the acceptance ratio is simply 0, and the move is not accepted. Alternatively, and this is more efficient, one could take care never to propose such a move.

5.4 Correspondences as Matchings

It is convenient to look at the correspondence problem in each image in isolation, and think of it in terms of *weighted bipartite graph matching.* By abstracting away from the structure from motion problem, we can concentrate on sampling from weighted assignments distributed according to a Gibbs distribution. This point of view is beneficial, as weighted matchings are well-studied constructs in combinatorial optimization (Papadimitriou and Steiglitz, 1982; Bertsekas, 1991; Cook et al., 1998). Abstracting away from the problem at

hand will allow us to more easily apply insights from the extensive literature on matchings. In addition, it has the benefit of unburdening the notation somewhat.

Consider the bipartite graph $G = (U, V, E)$ in image i where the vertices U correspond to the image measurements, i.e., $u_k \stackrel{\Delta}{=} \mathbf{u}_{ik}$, and the vertices V are identified with the features, i.e., $v_j \stackrel{\Delta}{=} \mathbf{x}_j$. Both k and j range from 1 to n, i.e., $|U| = |V| = n$. Finally, the graph is fully connected by the set of edges $E = U \times V$, and we associate the following *edge weight* with each edge $e = (u_k, v_j)$:

$$w(u_k, v_j) \stackrel{\Delta}{=} -\log P(\mathbf{u}_{ik}|\mathbf{j}_{ik}, \mathbf{m}_i, \mathbf{x}_{\mathbf{j}_{ik}}) \tag{5.7}$$

Definition A *matching* is defined as a subset M of the edges E, such that each vertex is incident to at most one edge. An *assignment* is defined as a perfect matching: a set of n edges such that every vertex is incident to exactly one edge.

Given these definitions, it is easily seen that every assignment vector $\mathbf{j}_i$ corresponds to an assignment in the bipartite graph G, so we use the same symbol to denote both entities. Furthermore, we use the notation $\mathbf{j}_i(u)$ to denote the match of a vertex u, i.e., $\mathbf{j}_i(u_k) = v_j$ iff $\mathbf{j}_{ik} = j$. Recalling equation (5.3), it is easily seen that *for valid assignments* $\mathbf{j}_i \in \mathcal{P}_i^n$, the posterior probability $f_i(\mathbf{j}_i)$ can be expressed in terms of the edge weights as follows:

$$f(\mathbf{j}_i) \propto \exp\left[\sum_{k=1}^{n} \log P(\mathbf{u}_{ik}|\mathbf{j}_{ik}, \mathbf{m}_i, \mathbf{x}_{\mathbf{j}_{ik}})\right] \propto e^{-w(\mathbf{j}_i)} \tag{5.8}$$

where the *weight* $w(\mathbf{j}_i)$ of an assignment is defined as

$$w(\mathbf{j}_i) = \sum_{k=1}^{n} w(u_k, \mathbf{j}_i(u_k))$$

Expression (5.8) has the form of a Gibbs distribution, where $w(\mathbf{j}_i)$ plays the role of an energy term: assignments with higher weight (energy) are less likely, assignments with lower weight (energy) are more likely.

Thus, the Gibbs distribution provides the link between weighted assignments on the one hand, and the posterior probability of the associated correspondence on the other hand. Clearly, this is no coincidence, as the weights are exactly defined as the log-likelihoods (i.e., reprojection errors!) of the associated correspondence assignments. Keeping this connection in mind helps a great deal in understanding the overall MCEM approach.

5.5 An Efficient Sampler

The previous section showed that the problem of sampling from the assignment vectors $\mathbf{j}_i$ in the structure and motion problem is equivalent to sampling from weighted assignments in the bipartite graph G, where the target distribution is given by the Gibbs distribution (5.8). Below we temporarily abstract away from the application at hand (structure from motion and derived geometric estimation problems) problem and think solely in terms of weighted assignments J in a single image.

In this section I show that the Metropolis-Hastings method can be made to very effectively sample from weighted assignments. The convergence of the Metropolis-Hastings algorithm depends crucially on the proposal density g. We need a proposal strategy that leads to a rapidly mixing Markov chain, i.e., one that converges quickly to the stationary distribution. Below we discuss three different proposal strategies, each of which induces a Markov chain with increasingly better convergence properties.

5.5.1 Flip Proposals

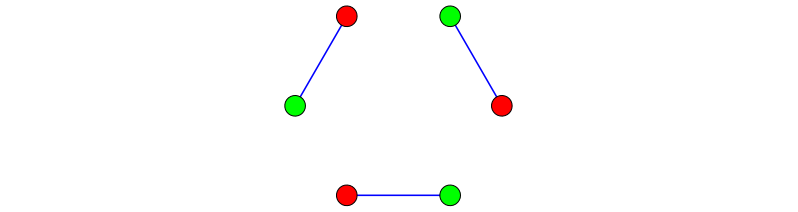
The simplest way to propose a new assignment J' from a current assignment J is simply to swap the assignment of two randomly chosen vertices u:

1. Pick two matched edges (u_1, v_1) and (u_2, v_2) at random.
2. Swap their assignments, i.e., set $J'(u_1) \leftarrow v_2$ and $J'(u_2) \leftarrow v_1$

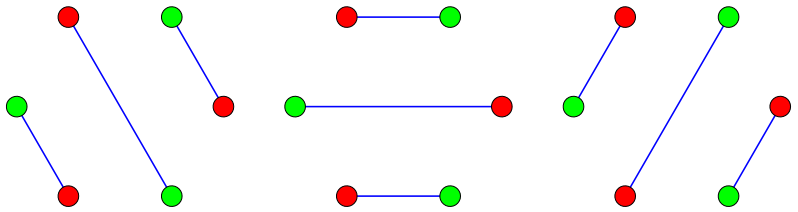
To calculate the ratio a, note that the proposal ratio $\frac{g(J;J')}{g(J';J)} = 1$. Thus, the acceptance ratio a is equal to the probability ratio, given by

$$a = \frac{f(J')}{f(J)} = \exp\left[w(u_1, v_1) + w(u_2, v_2) - w(u_1, v_2) - w(u_2, v_1)\right]$$

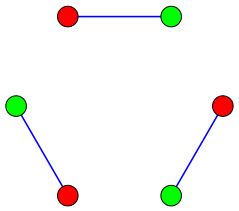
Even though this "flip proposal" strategy is attractive from a computational point of view, it has the severe disadvantage of leading to slowly mixing chains in many instances. To see this, consider the arrangement with $n = 3$ in Figure 5.1. The regular arrangement of the vertices on the circle means that there are two equally optimal assignments, (a) and (e). The probability distribution over the assignments is given in Table 5.1: as expected configurations (a) and (e) contain most of the probability mass, whereas (b-d) are much



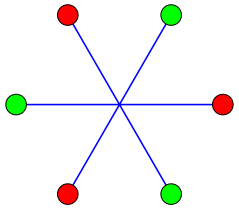
(a)



(b) (c) (d)



(e)

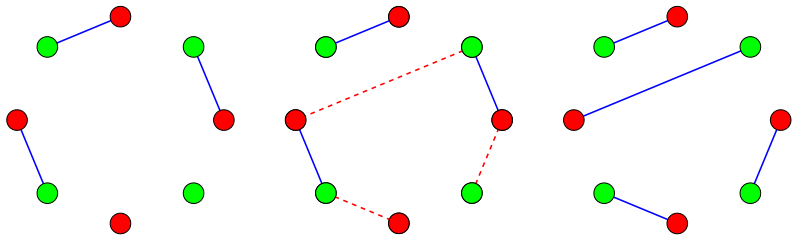


(f)

Figure 5.1: An ambiguous assignment problem with $n = 3$. All vertices lie on a circle with radius R. See text for explanation.

a	b	c	d	e	f
49.994	0.004	0.004	0.004	49.994	0.000

Table 5.1: The probability distribution (in percent) over the assignments in Figure 5.1, according to the Gibbs distribution with defined by isotropic Gaussian noise with standard deviation $\sigma = 0.4R$ (with R the radius).



(a) (b) (c)

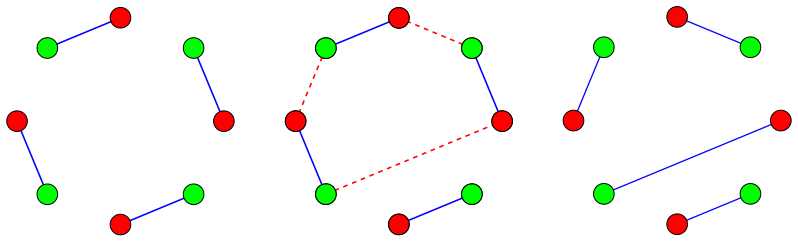
Figure 5.2: Augmenting paths. (a) Original, partial matching. (b) An augmenting path, alternating between free and matched edges. (c) The resulting matching after augmenting the matching in (a) with the path in (b) .

less likely, and (f) is very improbable. The figure illustrates a major problem with "flip proposals": there is no way to move from (a) to (e) via flip proposals without passing through one of the unlikely states (b-d). An MCMC sampler that proposes only such moves can stay stuck in the modes (a) or (e) for a long time.

5.5.2 Augmenting Paths and Alternating Cycles

In order to improve the convergence properties of the chain, we use the idea of randomly generating an *augmenting path*, a construct that plays a central role in deterministic algorithms to find the optimal weighted assignment (Bertsekas, 1991; Cook et al., 1998; Papadimitriou and Steiglitz, 1982). The intuition behind an augmenting path is simple: it is a way to resolve conflicts when proposing a new assignment for some random vertex in U. When sampling, an idea for a proposal density is to randomly pick a vertex u and change its assignment, but as this can lead to a conflict, we propose to use a similar mechanism to resolve the conflict recursively.

We now explain augmenting paths following (Kozen, 1991). Assume we have a partial



(a) (b) (c)

Figure 5.3: (a) Original assignment. (b) An alternating cycle implementing a k-swap, with k=3 in this example. (c) Newly obtained assignment.

matching M. An example is given in Figure 5.2 (a). Now pick an unmatched vertex u, and propose to match it up with v. We indicate this by traversing the free edge (u, v). If v is free, we can simply add this edge to the matching M. However, if v is not free we cancel its current assignment by traversing the *matched* edge (v, u'). We then recurse, until a free vertex in V is reached, tracing out the *augmenting path* p. One such a path is shown in Figure 5.2 (b). Now the matching can be *augmented* to M' by swapping the matched and the free edges in p. This *augmentation* operation is written as $M' = M \oplus p$, where $\oplus$ is the symmetric difference operator on sets

$$A \oplus B = (A \cup B) - (A \cap B) = (A - B) \cup (B - A)$$

For the example, the resulting matching is shown in Figure 5.2 (c).

Algorithms to find optimal matchings start with an empty matching, and then perform a series of augmentations until an optimal matching is obtained (Kozen, 1991). For sampling purposes alternating *cycles* are of interest, because they implement k-swaps. An example is shown for $n = 4$ in Figure 5.3. In contrast to the optimal algorithms, when sampling we start out with a perfect matching (an assignment), and want to propose a move to a different (also perfect) matching. We can do this by proposing the matching $J' = J \oplus C$, where C is an alternating cycle. This has the effect of permuting a subset of the assignments. Permutations that leave no element untouched are called *derangements,* and hence any alternating cycle implements a derangement of a subset of the assignments.

5.5.3 Proposing Moves by "Chain Flipping"

Recall that the goal is to sample from assignments **J** using the Metropolis-Hastings algorithm. We now advance a new strategy to generate proposed moves, through an algorithm that we call "chain flipping" (CF). The algorithm is based on randomly generating an alternating cycle according to the following algorithm:

1. Pick a random vertex u in U
2. Choose a match v in V by traversing the edge $e = (u, v)$ according to the transition probabilities
$$q(u, v) \triangleq \frac{\exp(-w(u, v))}{\sum_v \exp(-w(u, v))} \tag{5.9}$$
which accords higher probability to edges $e = (u, v)$ with lower weight.
3. Traverse the matched edge (v, u') to undo the former match.
4. Continue with 2 until a cycle is formed.
5. Erase the transient part to get an alternating cycle C.

This algorithm simulates a Markov chain MC defined on the bipartite graph G and terminates the simulation when a cycle is detected. The resulting alternating cycle C is used to propose a new assignment $J' = J \oplus C$, i.e., we "flip" the assignments on the alternating cycle or "chain" of alternating edges.

We also need to calculate the acceptance ratio a. As it happens, we have

$$a_{CF} = \frac{f(J')}{f(J)} \frac{g(J; J')}{g(J'; J)} = 1 \tag{5.10}$$

To prove this, note that by (5.8) and (5.9) the probability ratio is given by

$$\frac{f(J')}{f(J)} = \frac{e^{-w(J')}}{e^{-w(J)}} = \prod_{u \in C} \frac{q(u, J'(u))}{q(u, J(u))} \tag{5.11}$$

The proposal density $g(J'; J)$ is equal to the probability of proposing a cycle C that yields J' from J, which is given by:

$$g(J'; J) = \left(\prod_{(u,v) \in p} q(u, J'(u)) \right) \sum_T P_{MC}(T) \tag{5.12}$$

where the sum is over all transient paths T that end on the cycle C, and $P_{MC}(T)$ is the probability of one such transient. The probability $g(J; J')$ of proposing J starting from J' is similarly obtained, and substituting both together with (5.11) into (5.10) yields the surprising result $a = 1$.

A distinct advantage of the CF algorithm is that, as with the Gibbs sampler (Gilks et al., 1996), every proposed move is always accepted. The n^2 transition probabilities $q(u, v)$ are also fixed and can be easily pre-computed. A major disadvantage, however, is that many of the generated paths do not actually change the current assignment, making the chain slower than it could be. This is because in step 2 there is nothing that prevents us from choosing a matched edge, leading to a trivial cycle, and in steady state matched edges are exactly those with high transition probabilities.

5.5.4 "Smart Chain Flipping"

An obvious modification to the CF algorithm, and one that leads to very effective sampling, is to make it impossible to traverse through a matched edge when generating the proposal paths. This ensures that every proposed move does indeed change the assignment, *if* it is accepted. However, now the ratio a can be less than 1, causing some moves to be rejected.

Forcing the chosen edges to be free can be accomplished by modifying the transition probabilities $q(u, v)$. We denote the new transition probabilities as $q^{\mathbf{J}}(u, v)$, as they depend on the current assignment J, and define them as follows:

$$q^{\mathbf{J}}(u,v) \triangleq \begin{cases} \frac{\exp(-w(u,v))}{\sum_{v \neq J(u)} \exp(-w(u,v))} & \text{if } v \neq J(u) \\ 0 & \text{if } v = J(u) \end{cases}$$

i.e., we disallow the transition through a matched edge. We can rewrite this in terms of the transition probabilities $q(u, v)$ defined earlier in (5.9), as follows

$$q^{\mathbf{J}}(u,v) = \begin{cases} \frac{q(u,v)}{1-q(u,J(u))} & \text{if } v \neq J(u) \\ 0 & \text{if } v = J(u) \end{cases}$$

Note that *these depend on the current assignment J*, but in an implementation their explicit calculation can be avoided by appropriately modifying the cumulative distribution function of q at run-time.

This proposal strategy, which we call "smart chain flipping" (SMART), generates more exploratory moves than the CF algorithm, but at the expense of rejecting some of the moves. It can be easily verified that we now have

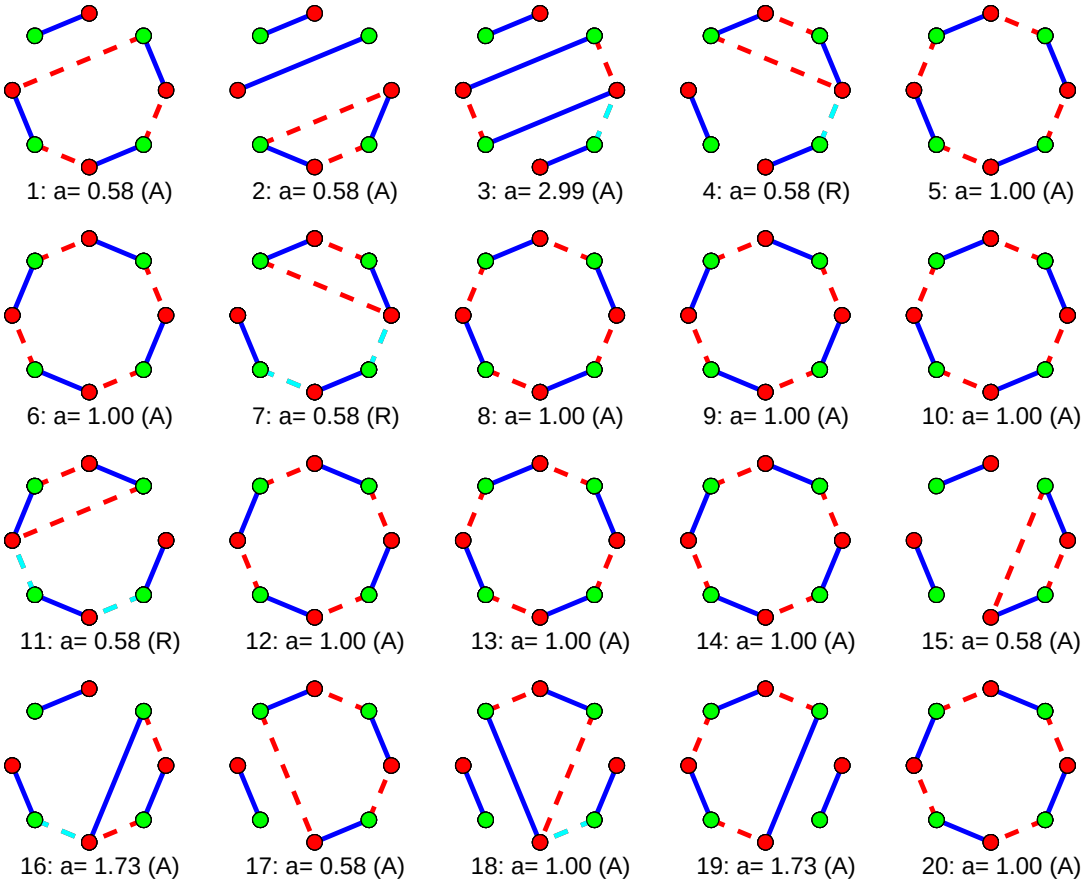
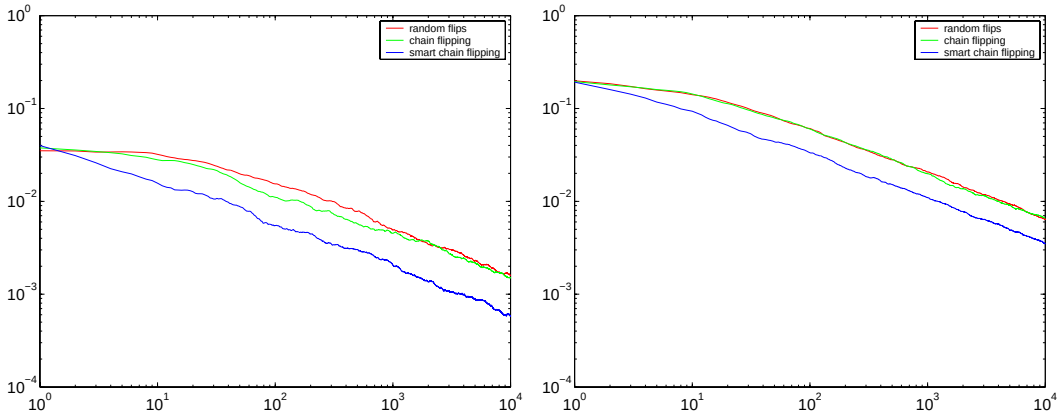


Figure 5.4: 20 iterations of an MCMC sampler with the "smart chain flipping" proposals. The current matches are shown as solid blue edges, the proposed matches as dashed red edges, and the transient part as dashed cyan edges. The acceptance ratio a is shown, as well as whether the move was accepted (A) or rejected (R).



(a) Low Temperature (b) High Temperature

Figure 5.5: Log-log plot comparing the mean absolute error (y-axis) versus number of samples (x-axis) for the 3 different proposal distributions: random flips, chain flipping, and smart chain flipping. (a) For a 'sharp' distribution with low annealing parameter $\sigma = 0.2$, and (b) for a high value of $\sigma = 0.6$.

$$a_{SMART} = \prod_{u \in C} \frac{1 - q(u, J(u))}{1 - q(u, J'(u))}$$

In Figure 5.4 we have shown 20 iterations of a Metropolis-Hastings sampler using the SMART proposals, and also show the value of a and whether the move was accepted (A) or rejected (R).

5.5.5 Results for Efficient Sampling

Experimental results support the intuition that "smart chain flipping" leads to rapidly mixing chains. In order to assess the relative performance of the three different samplers I have discussed above, I generated 1000 synthetic weighted assignment instances with $n = 5$, and ran each sampler for 10000 iterations on each example. There was no need to wait until the stationary distribution was reached, as the initial assignment was drawn from the exact distribution to start with, which is possible for examples with small n.

Figure 5.5 shows a log-log plot of the average absolute error (averaged over all examples) for one of the marginal statistics (expression 5.5 on page 69) as compared to the true value

(definition 5.2 on page 66). This was done for two different values of the annealing parameters σ, which determines the smoothness of the distribution. As can be seen from the figure, the "smart chain flipping" proposal is an order of magnitude better than the two other samplers, i.e. it reaches the same level of accuracy in far fewer iterations. For lower temperatures, i.e. sharper distributions, the difference is more pronounced. For higher temperatures, the errors are larger on average (as the sampler needs to explore a larger typical set), and the difference is less pronounced. It can also be seen that the difference between the random flip and (non-smart) chain flipping proposals is negligible.

Another approach to assess the convergence of the sampler is discussed in (Gelman, 1996): we can plot the time series for a single summary statistic in multiple, concurrently run MCMC simulations. Convergence can be assumed if all time series converge to the same value for the statistic. Displays such as this also give a qualitative understanding of the behavior of the different strategies, as we discuss in more detail below.

For Figure 5.6, we sample from a distribution over assignments with $n = 4$, for the configuration of features and observations as shown in Figure 5.4. It is clear from the latter figure that there are two globally optimal assignments, leading to a strongly bimodal distribution. In Figure 5.6 we show the convergence of each of the three proposal strategies discussed above, respectively from top to bottom: "flip proposals", "chain flipping", and "smart chain flipping". For each strategy, we show the results for a relatively smooth distribution ($\sigma = 0.9R$, shown at left), and a relatively peaked distribution ($\sigma = 0.5R$, shown at right). The summary statistic used is the proportion of samples that assigns observation 1 to feature 1, estimated by the average

$$\hat{J}_{11} \triangleq \frac{1}{T} \sum_{t} \delta(J^t(1), 1)$$

In the case of the low value for σ, this value is expected to be equal to 0.5, and smaller for higher values of σ. In all cases, the sampler was run for 1100 iterations, the first 100 of which were discarded as a transient.

We draw the following inferences from these figures:

- "Flip proposals" are very slowly mixing and get stuck on high probability assignments, especially for peaked distributions (low σ). This is evident from Figure 5.6 (b).
- "Chain flipping" leads to better mixing, but from the Figure 5.6 (c) and (d) it is clear that there are long stretches where the assignment is not changed much if at all.

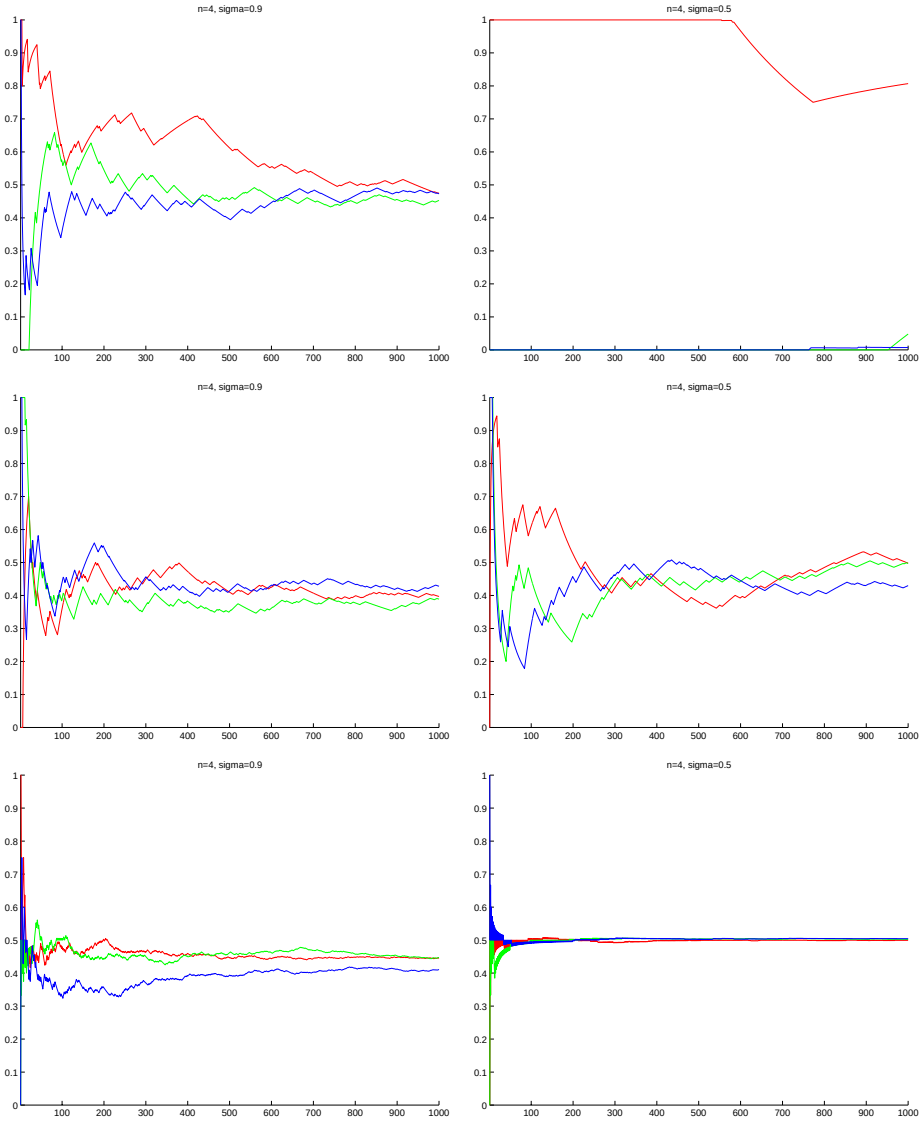


Figure 5.6: Assessing the convergence for the different proposal strategies. See text for explanation. (top) "Flip proposals", (middle) "chain flipping", (bottom) "Smart chain flipping". On the left, $\sigma = 0.9R$, on the right $\sigma = 0.5R$. The configuration that is being sampled over is the same as in Figure 5.4.

- Much better performance is obtained using “smart chain flipping”, especially for the peaked distribution on the right. The convergence to the bimodal distribution is almost immediate when compared to the other strategies. Convergence is somewhat slower for a high value of σ, as there are many more probable states that take some time to be visited often enough.

Chapter 6

Results with no Occlusion or Clutter

In this chapter I demonstrate that the Monte Carlo EM approach does indeed provide a practical way to approximate the optimal solution of multi-view geometric estimation problems with unknown correspondence. All the results shown in this chapter (as in the entire dissertation) concern the structure from motion (SFM) problem. As discussed in Section 4.1 on page 49, SFM can be regarded as a superset of many geometric estimation problems, and is also the most challenging of these problems in many respects.

The results shown below are for problems in which there is no occlusion or clutter, i.e. satisfying the assumptions made in Chapter 5. Whereas the MCEM algorithm was derived in Chapter 4 with no such restriction, the sampler from the previous chapter was designed to sample over the space of *assignments* in each image i, i.e. over matchings between n measurements $\mathbf{u}_{ik}$ and n 3D features $\mathbf{x}_j$, where both k and j range from 1 to n. In the next chapter, Chapter 7, we discuss probabilistic models for occlusion and clutter, and results for problems with occluded features or spurious measurements are presented in Chapter 8.

This chapter is divided into two sections. Section 6.1 illustrates the MCEM approach using the "cube" sequence, which I have used frequently in papers and talks to explain the approach in a user-friendly way. The next section, Section 6.2, shows additional results on real image sequences, both to establish that the approach is feasible and to illustrate its qualitative behavior on different image sets.

6.1 The MCEM Approach

6.1.1 Inputs to The Algorithm

The MCEM approach is illustrated below using a data-set derived from the images shown in Figure 6.1. There are 11 images of a calibration cube with texture on the sides. The images were taken under controlled conditions in the CMU calibrated imaging lab (CIL). They were taken as a sequence, but in order to illustrate that the MCEM approach does not need this information, the sequence information is disregarded in this example. In general, the correspondence matching and structure recovery can be made considerably easier if it is known in which temporal (or spatial) order the images were taken. In particular, this can be done in a straightforward manner by using a prior on the motion $\mathbf{M}$, if this were desired (in fact, an example of such a prior is discussed in detail in Section 8.1 on page 144).

The inputs to the MCEM algorithm are 50 measurements in each image, manually obtained by clicking on the same interesting features in all 11 images, for a total of 550 measurements. Figure 6.2 shows the measurements thus obtained for 6 out of the 11 images. As part of this manual process, the ground truth correspondence between the images was recorded, and is used below to present the output of the algorithm in a comprehensible manner. Naturally, the ground truth is not used by the MCEM algorithm itself.

6.1.2 Structure from Motion without Correspondence

To initialize the algorithm, the initial structure and motion estimate $\mathbf{\Theta}^0$ was generated as follows:

- The 50 features $\mathbf{x}_j$ were initialized randomly in a normally distributed cloud around the origin, with standard deviation $\sigma = 0.1$.
- The 11 cameras were all placed at location $t = (0, 0, -5)^T$, facing the origin.

The MCEM algorithm gradually recovers the 3D-structure of the cube and the location of each of the features, as shown in Figure 6.3. The recovered structure at each iteration is visualized by drawing colored polygons that correspond to the faces and/or salient features on the surface of the cube. Note that to do this, the ground truth correspondence is used to guess the most likely identity of each structure point (needed to draw the polygons). This

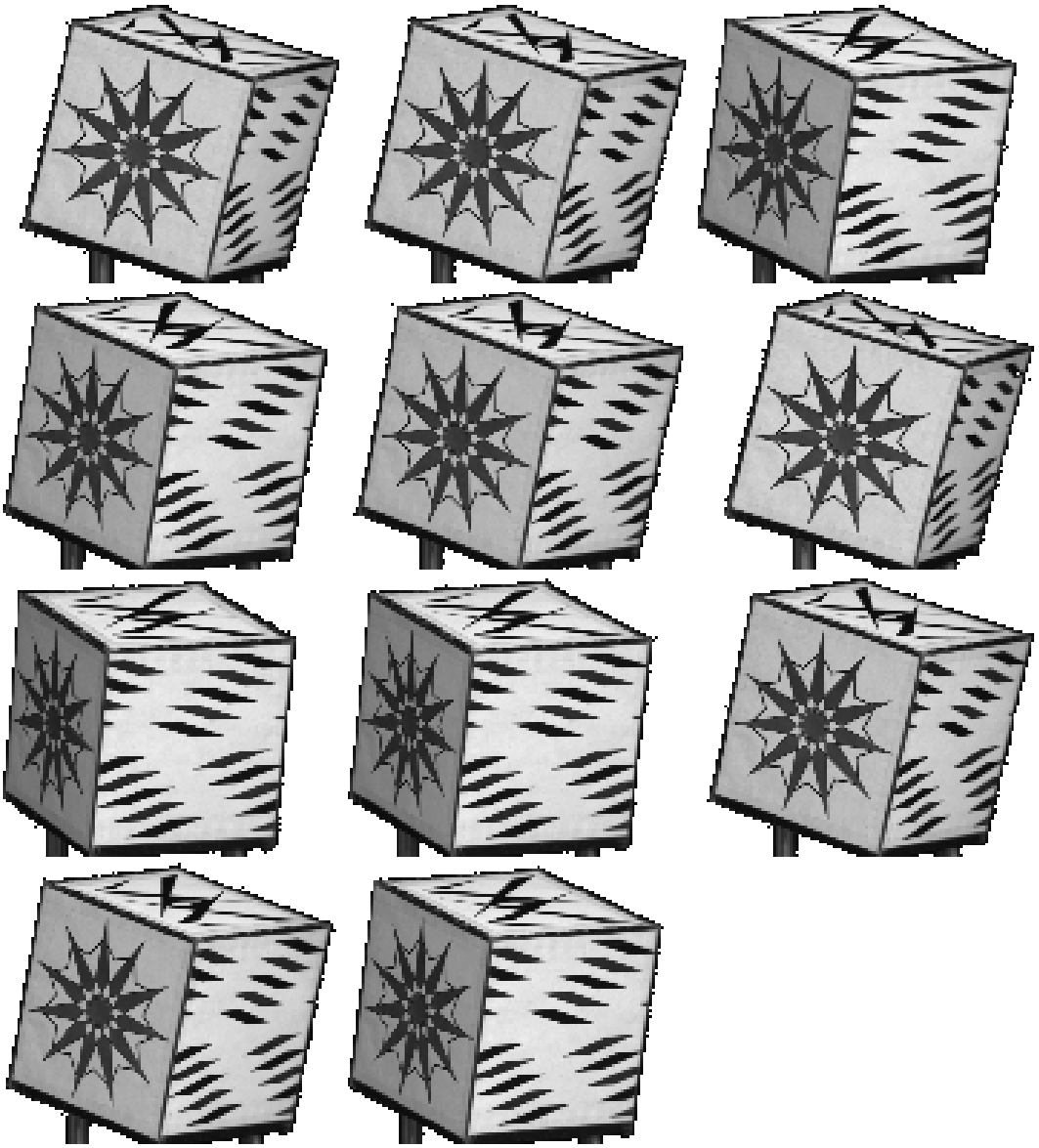
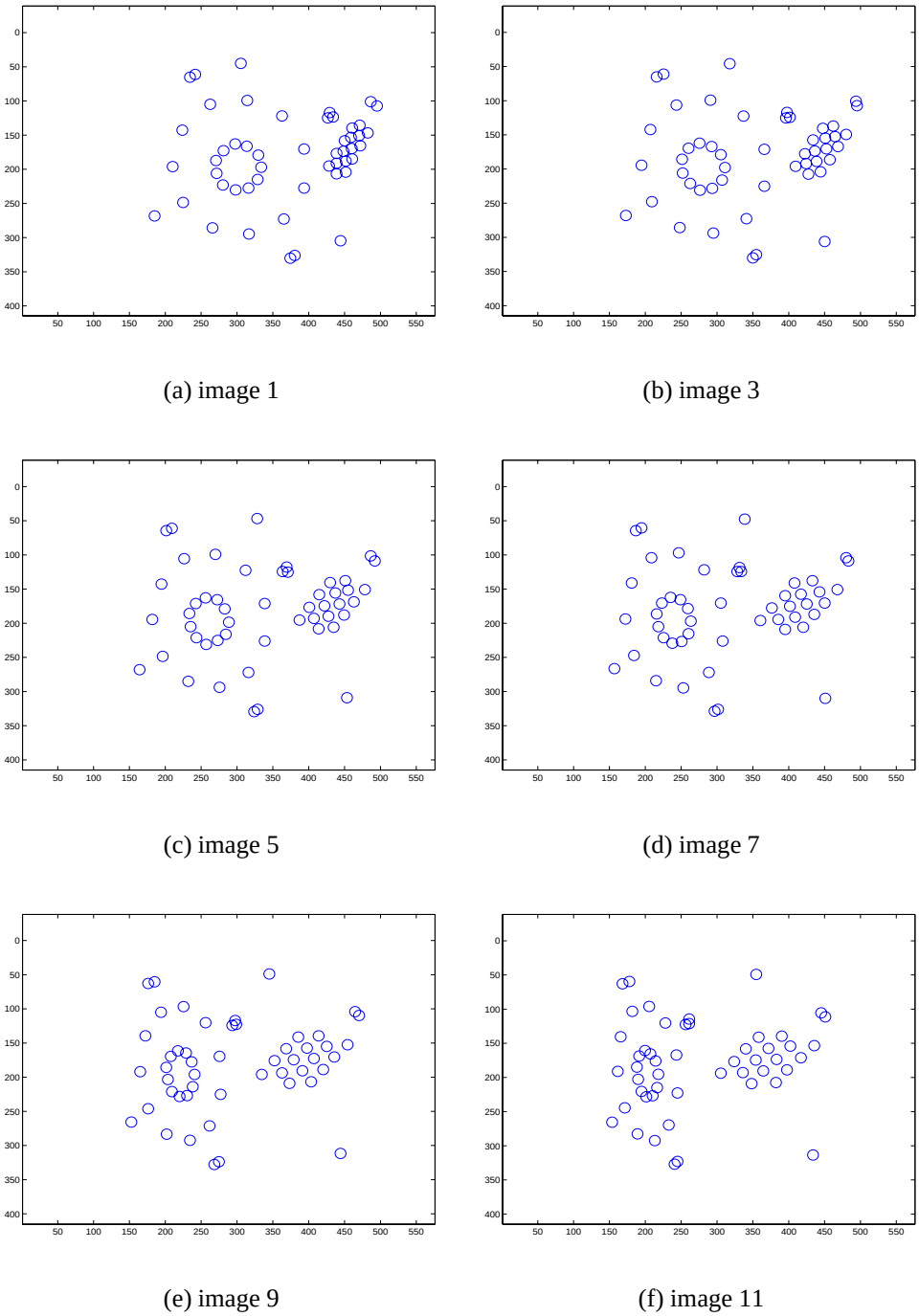


Figure 6.1: The 11 original “cube” input images.



(a) image 1

(b) image 3

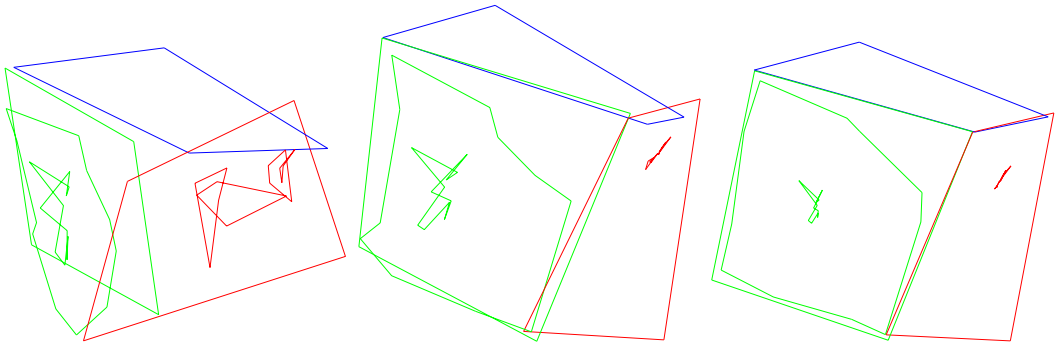
(c) image 5

(d) image 7

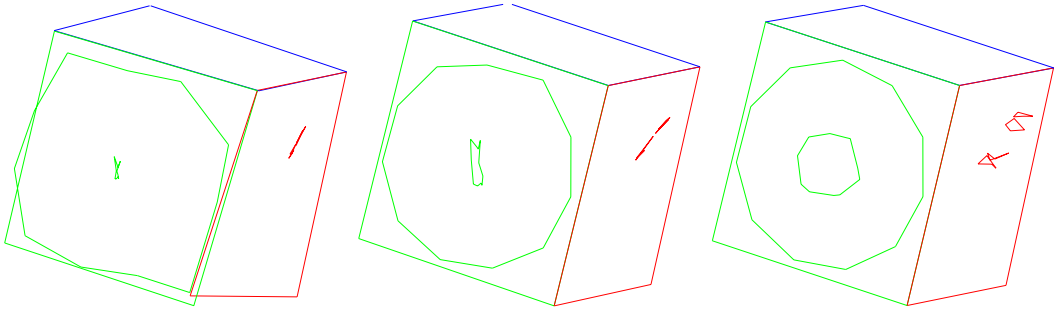
(e) image 9

(f) image 11

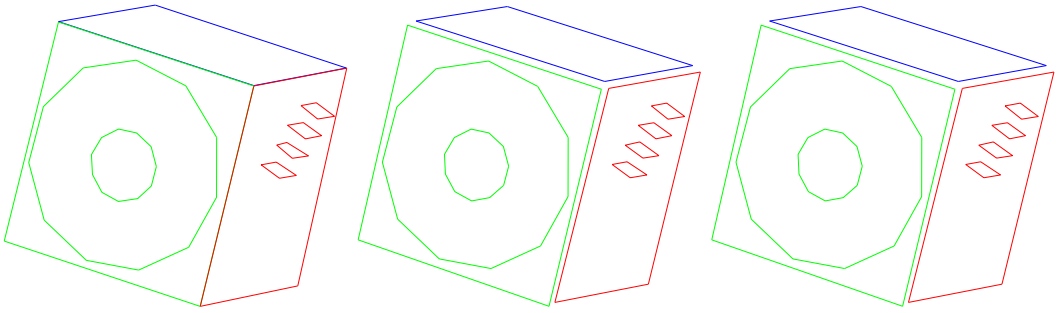
Figure 6.2: Measurements in 6 (out of 11) “cube” images.



(a) it 1 (b) it 5 (c) it 9



(d) it 13 (e) it 17 (f) it 21



(g) it 25 (h) it 29 (i) it 30

Figure 6.3: The structure estimate at successive iterations of the algorithm for the “cube” images.

is necessary as the order of the structure points can be scrambled by a random permutation, even if the "correct" correspondence between images is nearly recovered.

There are two important facts to note from Figure 6.3. First, even after the first iteration, the recovered structure shown in panel (a) is recognizable, and this *without the benefit of a known correspondence, and starting from a random initial estimate.* Second, as a consequence of the deterministic annealing strategy used to avoid local minima, the gross structure is recovered first, whereas small details are recovered more gradually as the annealing factor is decreased. Both these points will be discussed in more detail, but it is instructive to first take a more detailed look at the E-step.

In the example, the annealing schedule is linear with the initial annealing factor equal to $44 \times \sigma$, where σ is the noise standard deviation estimated at 1 pixel.

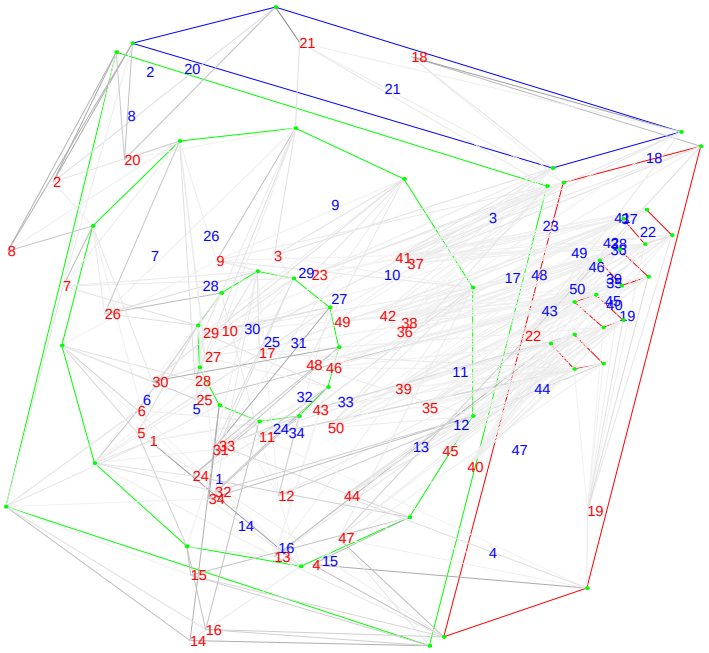
6.1.3 The E-step

Figure 6.4 illustrates the E-step and the computation of the virtual measurements. Recall that the EM algorithm alternates between an expectation step (E-step) and a maximization step (M-step). In each E-step we compute (or estimate using sampling, in this case), for each measurement $\mathbf{u}_{ik}$, the marginal probability f_{ijk}^t that it actually corresponds to feature $\mathbf{x}_j$, conditioned on the current structure and motion estimate $\mathbf{\Theta}^t$.

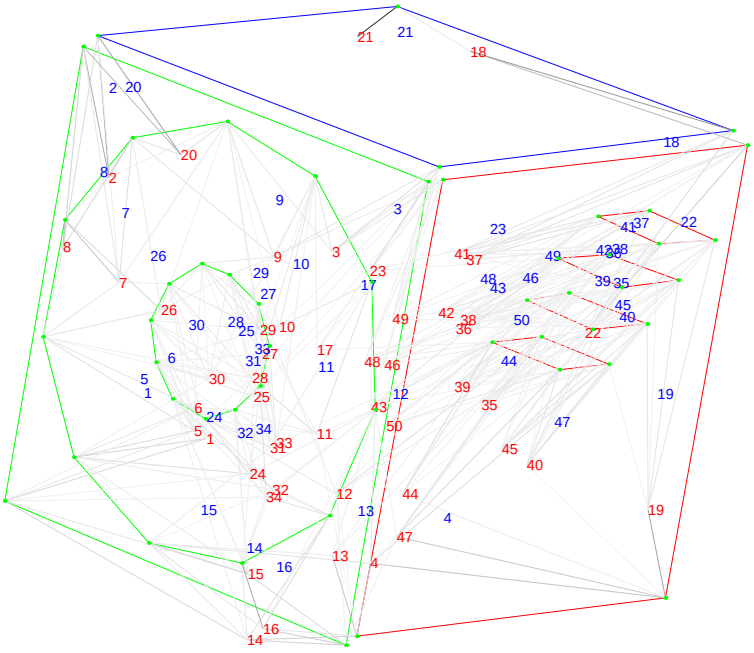
As explained in Chapter 5, this is done by first projecting the estimated structure $\mathbf{X}^t$ into each image according to the estimated motion parameters $\mathbf{m}_i$. These projected features $\mathbf{h}(\mathbf{m}_i, \mathbf{x}_j)$ are shown in Figure 6.4 as red numbers. Note that, because in the first iteration the structure estimate is random and all the cameras are at the same position, the projected features in both images 1 and 6 are in exactly the same location. In both images, the measurements are visualized using the same mesh as used in Figure 6.3, i.e. the measurements are the vertices of the colored mesh.

The Monte Carlo E-step proceeds by sampling over assignments between the projected features (red numbers) and the measurements (vertices of the mesh). The resulting estimated marginal probabilities are shown as grayscale edges between the feature projections and the measurements, where a darker edge means a more probable association.

The virtual measurements (shown as blue numbers) are then computed as weighted averages of the original measurements, with weights corresponding to the grayscale edges in Figure 6.4. There is exactly one virtual measurement $\mathbf{v}_{ij}^t$ in each image for each feature $\mathbf{x}_j$, and the weights used to compute the $\mathbf{v}_{ij}^t$ are those that connect the feature $\mathbf{x}_j$ with the



(a) image 1



(b) image 6

Figure 6.4: The calculation of virtual measurements (the E-step) in the first iteration, for images 1 and 6. See text for a detailed explanation.

measurements $\mathbf{u}_{ik}$ in image i. To illustrate this, consider the projected feature $\mathbf{h}(\mathbf{m}_1, \mathbf{x}_{19})$ which, by chance, ended up towards the top of the image (the red number 19 in Figure 6.4a). After running the sampler, feature $\mathbf{x}_{19}$ is estimated to be most strongly associated with measurements on the right-hand side of the image, corresponding to the cube vertices of the top-left corners (both in back and in front, but the association is stronger with those of the back corner). The virtual measurement $\mathbf{v}_{1,19}$ is formed as the weighted average of *all* measurements, but weighted according to the estimated correspondence probabilities f_{ijk}^t. Since the measurements on the right are much more strongly associated with feature $\mathbf{x}_{19}$, the virtual measurement $\mathbf{v}_{1,19}$ appears on the right, as well. It is shown as the blue number 19 in the figure.

Also apparent from the figure is the effect of the mutual exclusion constraint. For example, the projection of feature $\mathbf{x}_{19}$ is closer to the measurements on the left. However, these associations are made less probable because the projections of features $\mathbf{x}_{34}$, $\mathbf{x}_{27}$, $\mathbf{x}_{31}$, and $\mathbf{x}_7$, respectively, monopolize the probability mass allocated to the measurements on the left. This can be best understood in terms of the sampling mechanism: if an assignment is proposed in which $\mathbf{x}_{19}$ is associated with the left measurements, the other feature projections are forced (because of mutual exclusion) to make less probable associations elsewhere. Because of that, the acceptance ratio will be much smaller, i.e. such a proposal will most likely be rejected. *This is the essential difference with the E-step in mixture estimation,* in which the marginal probabilities are based on distance only. In the present case, shorter edges can actually be less probable.

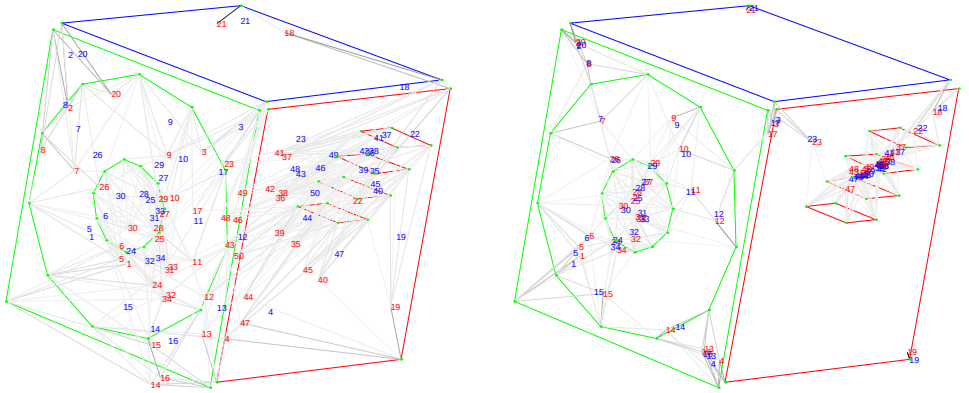
Finally, looking at both image 1 and 6 in combination, some intuition can be gained as to why gross structure is recovered in the first iteration. Because of mutual exclusion, as explained above, the virtual measurements $\mathbf{v}_{ij}^t$ can end up far from their associated projected features $\mathbf{h}(\mathbf{m}_i, \mathbf{x}_j)$. In addition, the images are not that different, as in this sequence, the separation between the camera viewpoints is not that large. Both factors combine to cause virtual measurements to be more or less consistently distributed towards the left or the right, or anywhere else in the image. Since the virtual measurements $\mathbf{v}_{ij}^t$ are used as input to bundle adjustment, the resulting structure and motion estimate $\mathbf{\Theta}^{t+1}$ does also have some consistency to it. However, because the initial structure estimate is completely random, there is of course no preservation of the order of the feature indices j. Even if, in a very unlikely case, all virtual measurements would end up being strongly associated with one measurement only, and the same one at that in all the different images, the indices j would still be scrambled by an arbitrary permutation.

6.1.4 Marginal Probabilities over Time

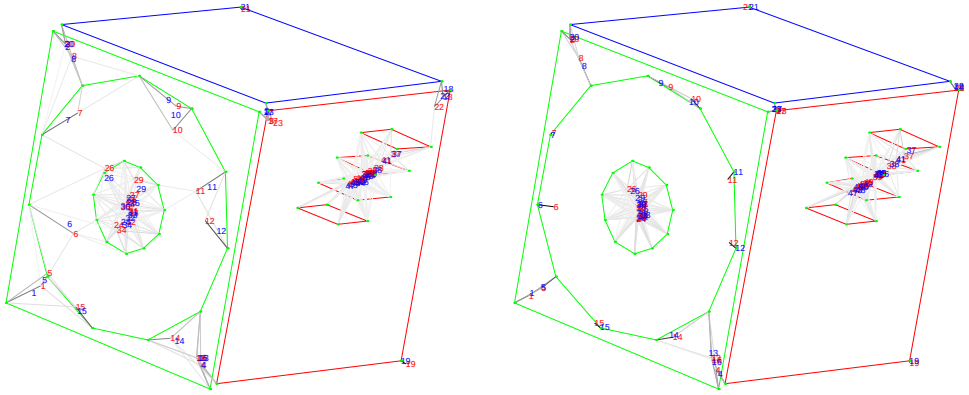
The behavior of the algorithm can further be illustrated by looking at how the marginal probabilities or *soft correspondences* f_{ijk}^t (and hence the virtual measurements $\mathbf{v}_{ij}^t$) change over time, which is illustrated in Figure 6.5 for one image. First, this figure clearly illustrates the effect of annealing: in the early iterations, measurements that are close together are grouped together. This is because for a high annealing factor, small differences in distance between projected features and measurements will not affect the marginal probabilities greatly. In later iterations, they separate out, and smaller structural features begin to appear in the structure estimate. Second, in later iterations, when the correct correspondence is close to being the only one with any probability mass, the virtual measurements almost coincide with the actual measurements. This is because one association dominates all other associations for a given feature $\mathbf{x}_j$, and hence the computation of the associated virtual measurement $\mathbf{v}_{ij}^t$ is dominated by the contribution of only one measurement.

A more concise and very insightful way to monitor the changing marginal probabilities is by displaying them as doubly stochastic matrices, as shown in Figures 6.6 and 6.7. For each image, the marginal probabilities f_{ijk}^t can arranged in an $n \times n$ matrix, since there are n measurements $\mathbf{u}_{ik}$ in each image and n features $\mathbf{x}_j$. This will be a *doubly stochastic matrix*, meaning that both rows and columns will sum to one. In each iteration, 11 of these matrices are computed, and they are graphically represented in figures 6.6 and 6.7 as a set of 11 stacked images. The n columns correspond to the n features $\mathbf{x}_j$, whereas the $m \times n$ rows correspond to the measurements $\mathbf{u}_{ik}$. The darker a pixel (k, j) is in subimage i (corresponding to image i), the higher the probability f_{ijk}^t. The probabilities change at every iteration, as (a) the E-step is conditioned on a changing structure and motion estimate $\mathbf{\Theta}^t$, computed in the M-step, and (b) because of deterministic annealing (see below).

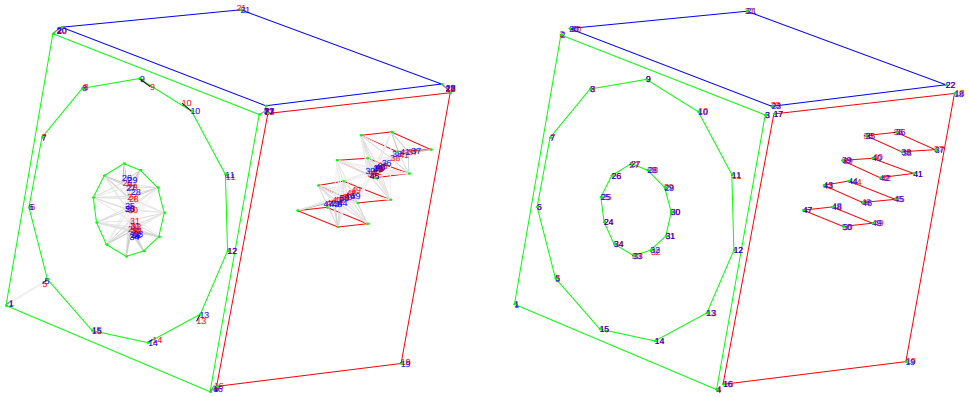
There are two important things to notice. First, the marginal probabilities converge to permutation matrices associated with the correct (ground-truth) correspondence. The matrices are presented in such a way that the ground truth corresponds to a stack of 11 identity matrices. This can be done only because we actually have the ground truth correspondence, and hence we can rearrange the columns and rows of the matrices to make it so. As can be seen in the figures, the marginal probabilities gradually converge to the identity matrices. In reality, the marginal probabilities corresponding to incorrect correspondences are not entirely zero. Even in the last iteration, when the algorithm has converged, a non-zero probability is estimated for "incorrect" correspondences. The word "incorrect" is in quotes, as from the point of view taken here, there are no correct or incorrect correspondences, only more



(a) it 1 (b) it 5

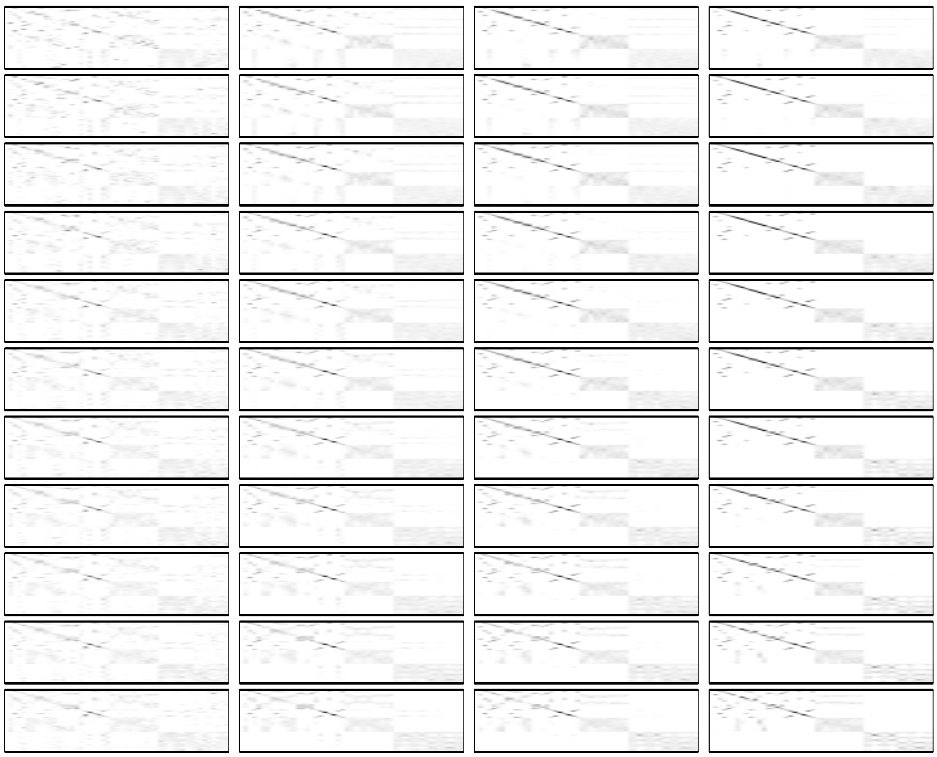


(c) it 9 (d) it 13



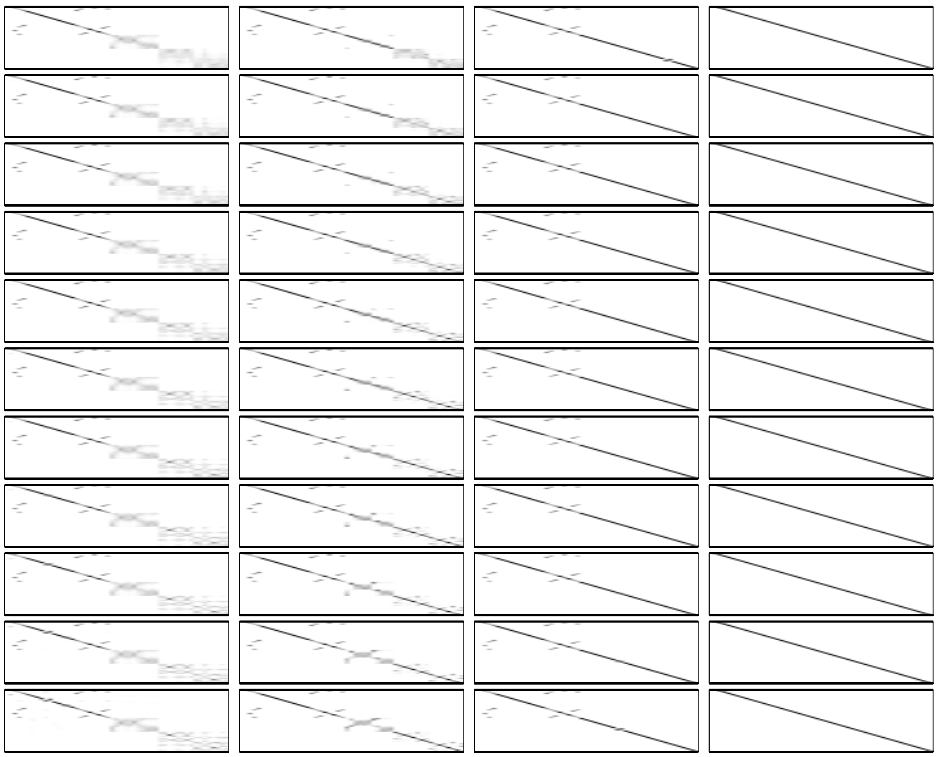
(e) it 17 (f) it 30

Figure 6.5: Marginal probabilities f_{ijk}^t and virtual measurements $\mathbf{v}_{ij}^t$ over time (image 6).



(a) it 1 (b) it 5 (c) it 9 (d) it 13

Figure 6.6: Marginal probabilities computed in the E-step.



(a) it 17 (b) it 21 (c) it 25 (d) it 30

Figure 6.7: Marginal probabilities (cont'd).

and less probable ones. Recall that we are only obtaining a point estimate for structure and motion, and in order to do so a distribution over correspondences is computed, not a single, "correct" one.

Second, the effect of annealing can clearly be seen in the figures. The decreasing annealing factor has the effect of gradually "sharpening" the posterior probability distribution over correspondences $\mathbf{J}$, and hence also its marginals f_{ijk}^t. Looking at iteration 17 (Figure 6.7a), for example, we see that while the first 20 or so features (corresponding to structural features on a larger scale, e.g. the cube vertices) are already sharply associated in a consistent manner across all images, the marginal probabilities corresponding to the last 30 features appear as gray "blocks" in the matrix. If examined carefully, three blocks can be discerned, corresponding respectively to the small circle on the front of the cube, and two distinct groups associated with the decoration on the side of the cube. We can correlate that with Figure 6.5e, which shows the calculation of virtual measurements for the same iteration (iteration 17). Looking at both figures in combination it is easy to see the connection between the two different ways of presenting essentially the same information.

6.1.5 The M-step

In the M-step, the virtual measurements computed in the E-step are used to re-estimate the structure and motion Θ. A 3D representation of the evolving structure estimate, as shown in Figure 6.3, is not always the most insightful way to represent this. In addition, it requires us to specify a mesh, which is not always applicable. An alternative is to instead plot the evolution of the *projected* structure in image space, as shown in Figure 6.8. In the figure, in the top panel, the measurements are shown as circles, and for each feature $\mathbf{x}_j$ the evolution of its projection $\mathbf{h}(\mathbf{m}_i, \mathbf{x}_j)$ is shown as a trajectory. Because in this case the correct correspondence is recovered, all the trajectories endpoints coincide (almost) exactly with a measurement. The endpoint itself is shown as an asterisk in the figure. Finally, at the bottom, the measurements and the trajectory endpoints are superimposed on the original input image, which clearly shows that the algorithm has converged to a consistent solution. An overview across all images can be gained by looking at the evolution of the projected structure in many images at once, as shown in Figure 6.9.

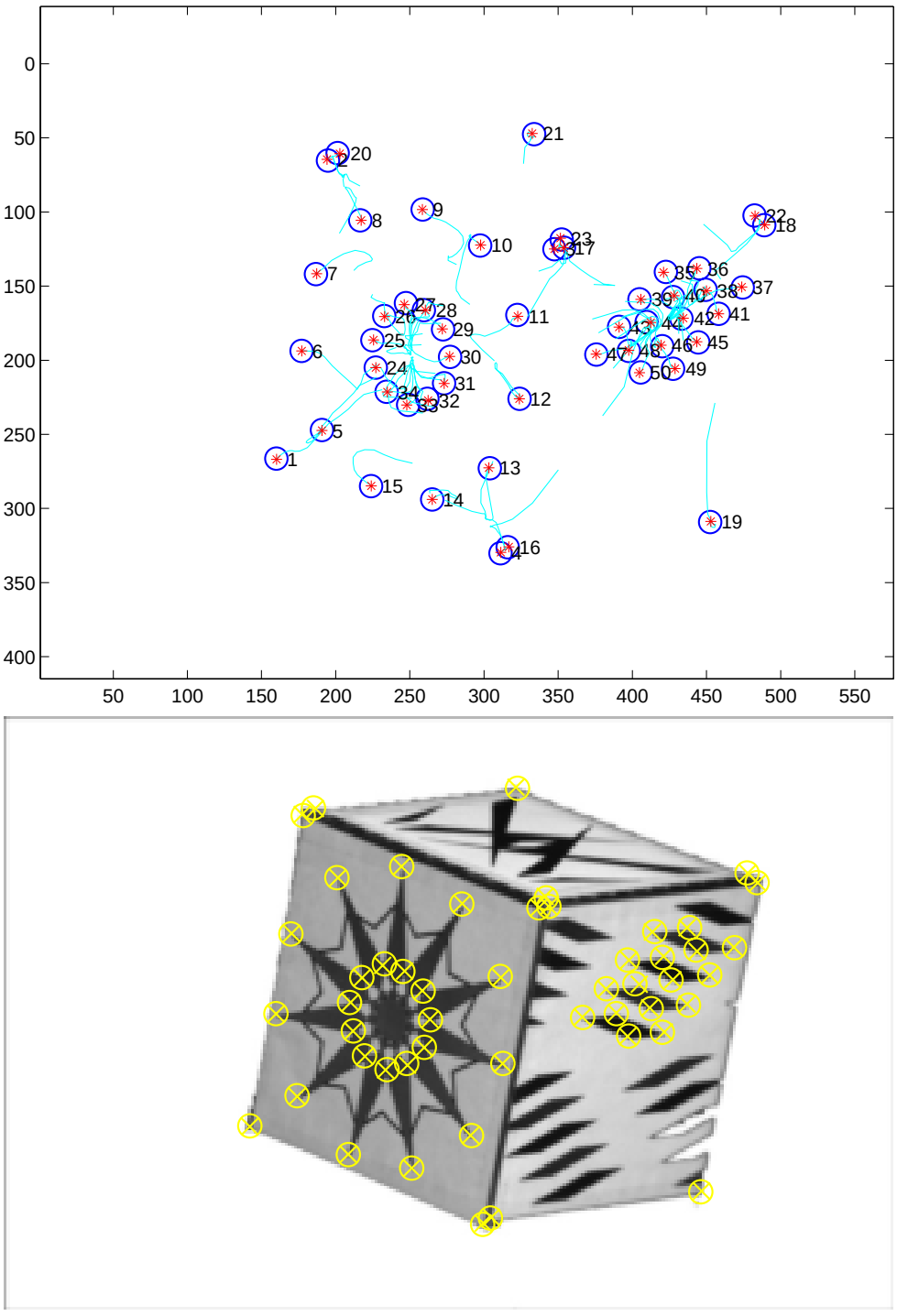
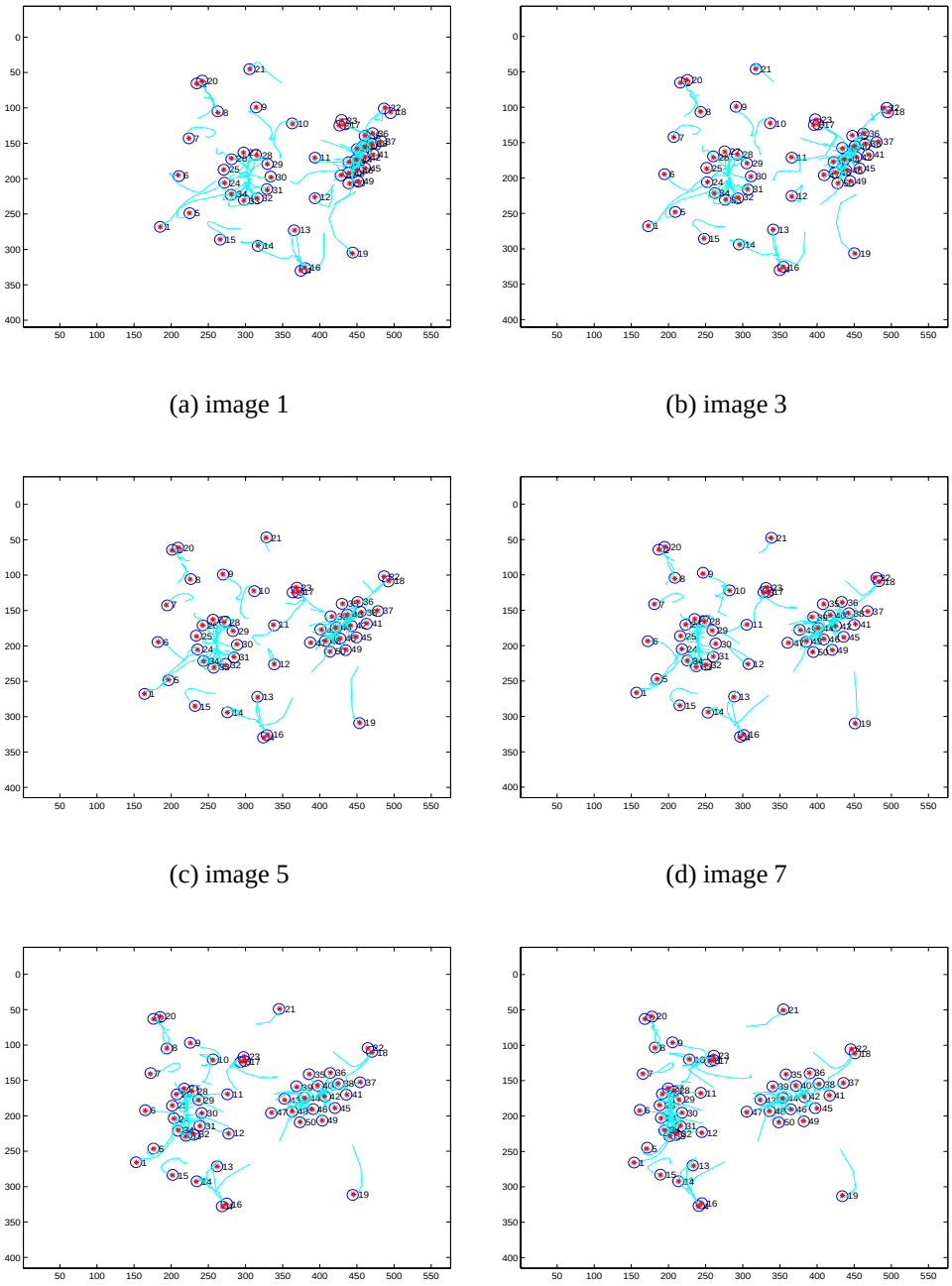


Figure 6.8: Top: plot of the projected features over time in one image. The last predicted location is marked with an asterisk. Measurements are shown as circles. Bottom: the last predicted location (x) and the measurements (o) superimposed on the original input image.



(a) image 1

(b) image 3

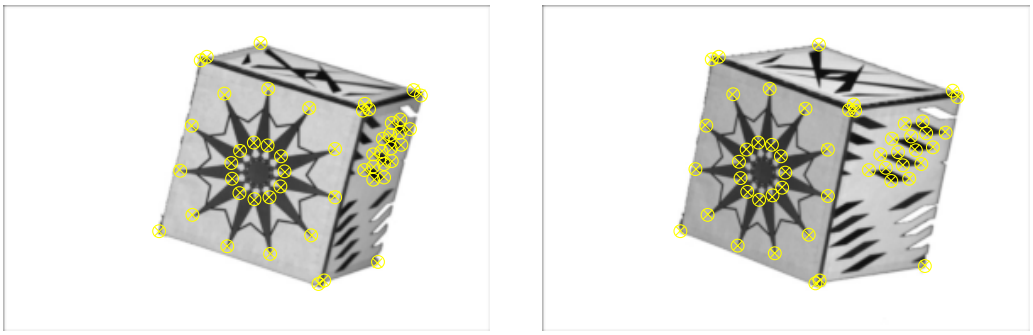
(c) image 5

(d) image 7

(e) image 9

(f) image 11

Figure 6.9: Plot of the predicted location for each of the features over time in each of the 6 “cube” images shown in Figures 6.2 and 6.10 (below).



(a) image 1

(b) image 5

(c) image 7

(d) image 11

Figure 6.10: 4 (out of 11) original “cube” input images. The last predicted location is marked with an asterisk. Measurements are shown as circles.

6.1.6 The Final Result

The quality and correctness of the final structure and motion estimate can be assessed by superimposing the measurements and the projected features corresponding to the converged estimate on the original input images. This representation is shown in Figure 6.10 for the case of the "cube" sequence, and it is used extensively below, as well.

6.2 More Results with Real Images

In this section, some more results are shown on different image sets. In order to appreciate what the algorithm uses as input, the raw 2D measurements are always shown first, without showing the actual input images. If the actual images are shown, the problem looks easy, as our visual cortex makes immediate sense of the scene. However, remember that the MCEM approach as presented above does not make use of any appearance information (at least not until Chapter 9), which is something we automatically do when we look at an image.

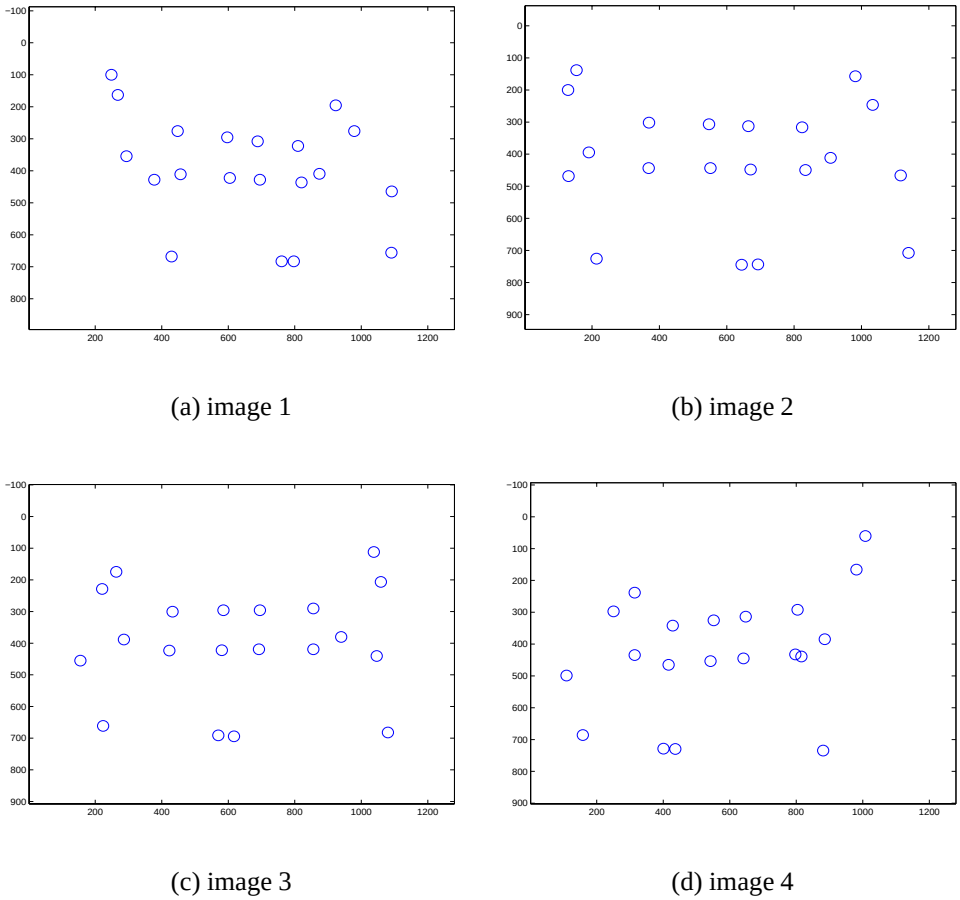
The results shown below were are all obtained under the assumption that there was no occlusion or clutter. In all cases, measurements were obtained in the same way as before, i.e. by a graduate student who manually clicked on salient features. This simulates a "perfect feature detector", which never reports any spurious measurements or misses a feature. Additionally, the features were chosen such that it is never occluded in any of the input images. As mentioned before, these assumptions will be relaxed in the next chapter.

6.2.1 Townhouse

The "townhouse" sequence illustrates that incorrect correspondences in the first iterations can be recovered from in later iterations. The image sequence consists of 4 images with 20 measurements in each image, shown in Figure 6.11.

In this case, the structure and motion were initialized in the same way as for the cube sequence, with all cameras looking to a normally distributed cloud of points. The evolution of the marginal probabilities over time, shown in Figure 6.12, illustrates that this initial estimate is quite good for this sequence, in which image viewpoints are indeed quite close. Indeed, the marginal probabilities in the very first iteration closely resemble stacked identity matrices, which correspond to the ground truth. The measurements incorrectly assigned in the first E-step do not have a large effect on the M-step, which in turn favors correct assignments in subsequent E-steps. By iteration 3 there is still confusion, but it is gradually cleared up as the annealing factor is decreased.

For this sequence, some generic prior knowledge about the motion parameters was used. The fact that the structure is initialized randomly does not matter that much: a good initial motion estimate is what matters more. In this case, the cameras were initialized at the same position, all upright, and looking at the same point in space. This is a good strategy for image sets with moderate motion between the images. Prior knowledge is also used in



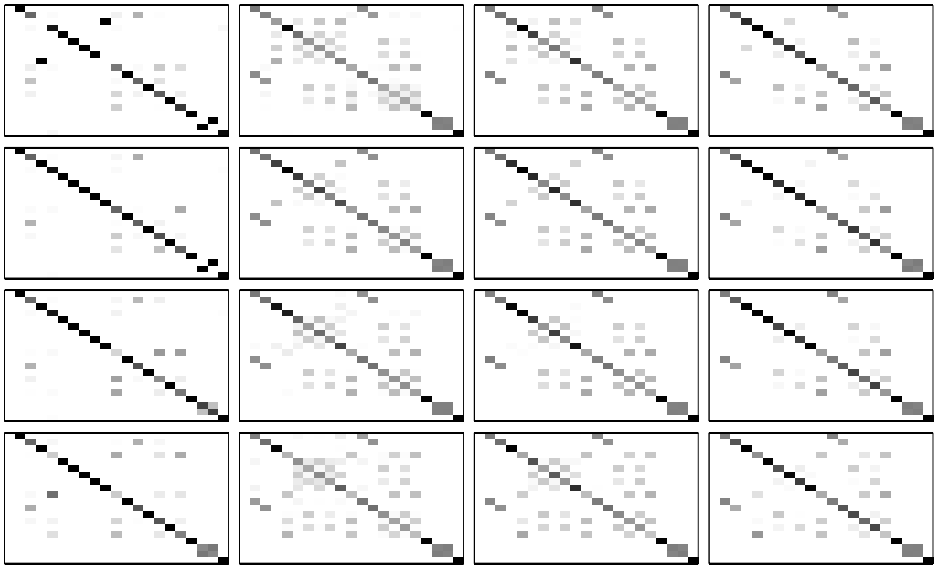
(a) image 1

(b) image 2

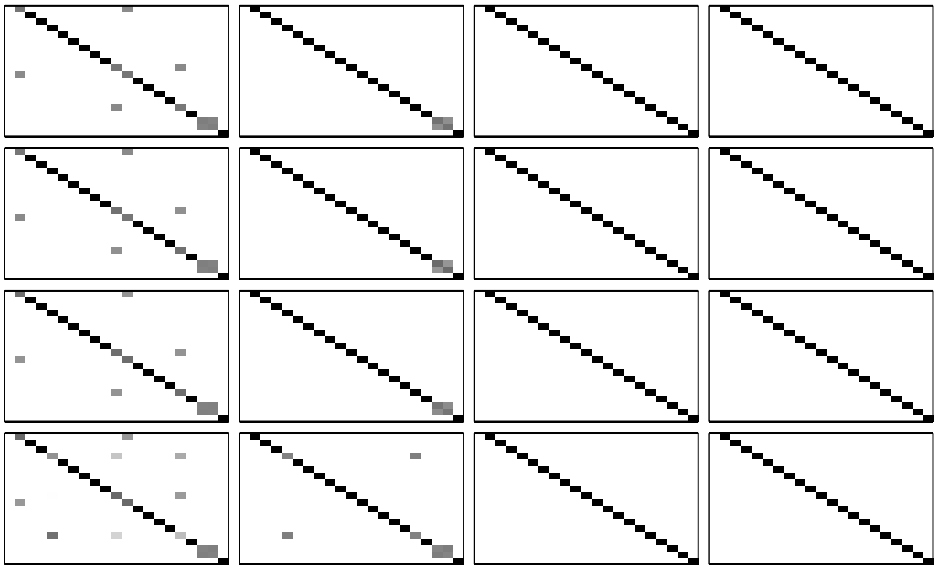
(c) image 3

(d) image 4

Figure 6.11: Measurements in the 4 “townhouse” input images.



(a) it 1 (b) it 3 (c) it 5 (d) it 7



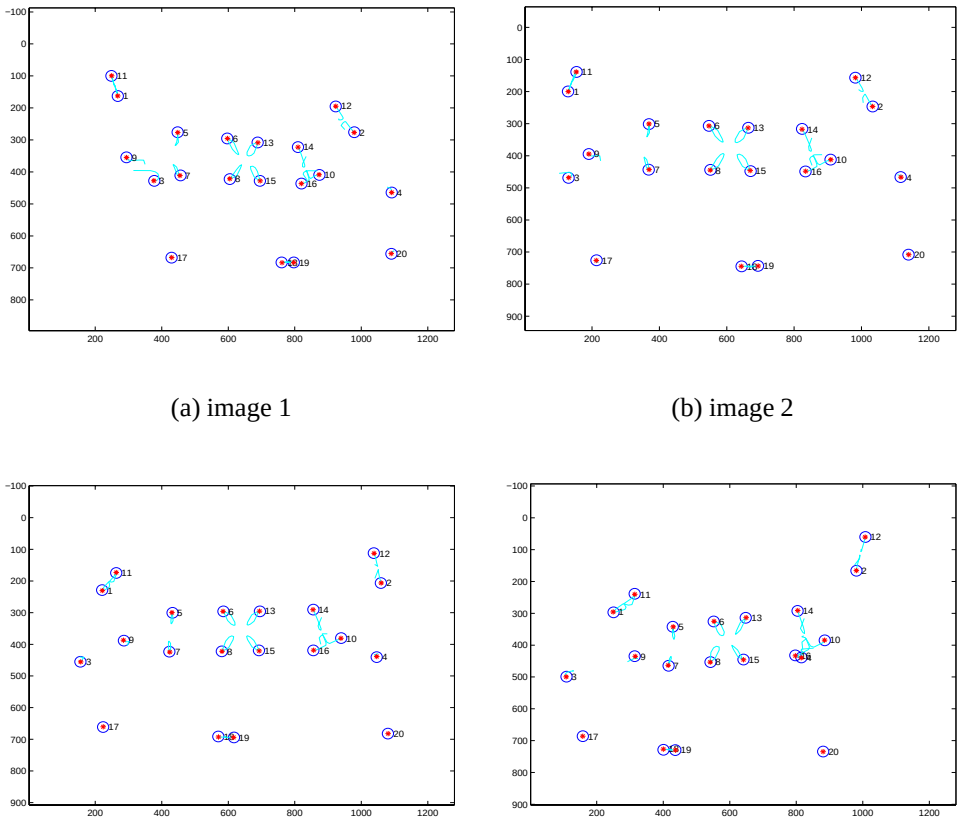
(e) it 9 (f) it 11 (g) it 13 (h) it 15

Figure 6.12: Marginal probabilities over time for the “townhouse” images series.

the M-step: in many cases, we know enough about how the sequence is taken that we can rule out many improbable values for the motion parameters. For example, in the majority of images taken with snapshot cameras, the orientation will be landscape. That fact is used here (and in all results below), to impose a strong prior on the roll parameter of the cameras (i.e. making non-zero values less probable). Pictures with portrait orientation could conceivably be handled automatically, as in many cases it is possible to guess the orientation from the images (e.g.bright sky belongs at the top).[1] A weaker prior is imposed on the vertical position of the camera: in this instance and also below, the prior biases the estimate towards all images taken from the same height.

The evolving structure estimate for the "townhouse" sequence is shown here using both trajectories of projected features, in Figure 6.13, and a 3D mesh representation, in Figure 6.14. In the latter figure, the camera position and orientation for the 4 cameras is shown, as well. Finally, the converged structure and motion estimate is illustrated by superimposing both the projected features and the measurements on the original input images, in Figure 6.15.

[1] In fact, research at the Kodak company addresses exactly this problem (personal communication with unnamed Kodak source).



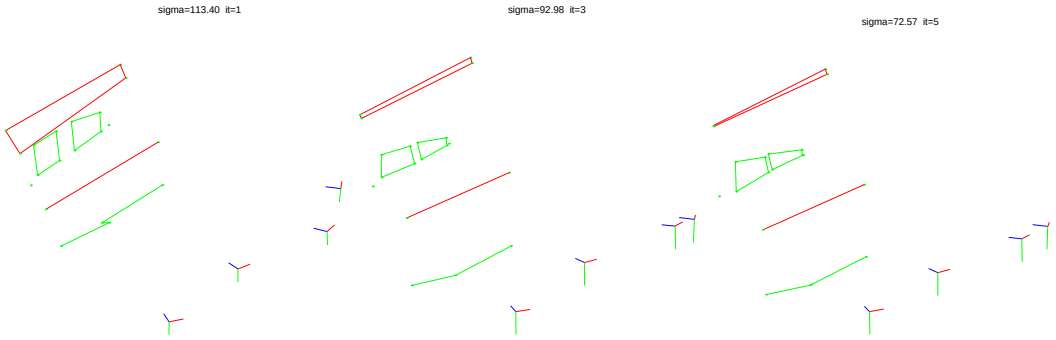
(a) image 1

(b) image 2

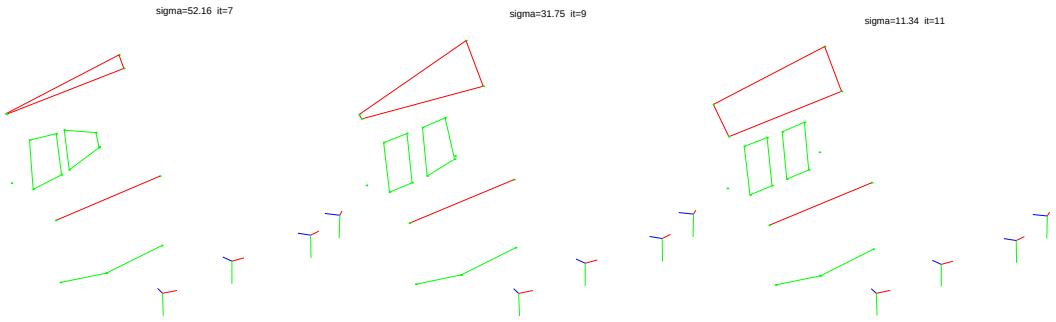
(c) image 3

(d) image 4

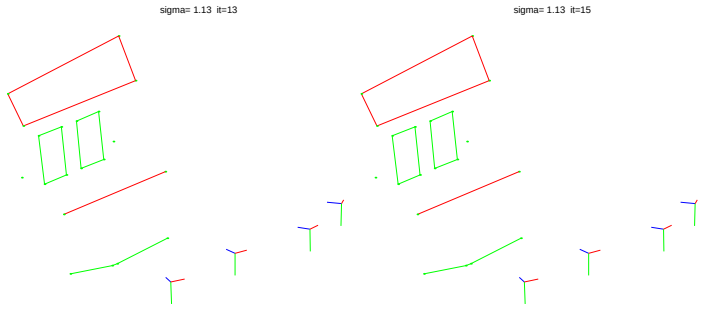
Figure 6.13: Plot of the predicted location for each of the features over time in the 4 “town-house” images. The last predicted location is marked with an asterisk. Measurements are shown as circles.



(a) it 1 (b) it 3 (c) it 5



(d) it 7 (e) it 9 (f) it 11



(g) it 13 (h) it 15

Figure 6.14: The structure estimate at successive iterations of the algorithm for the “town-house” image series.

(a) image 1

(b) image 2

(c) image 3

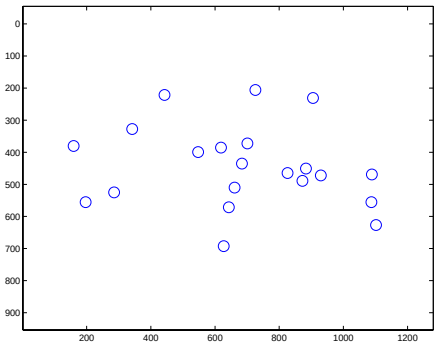
(d) image 4

Figure 6.15: The 4 original "townhouse" input images. The last predicted location is marked with an asterisk. Measurements are shown as circles.

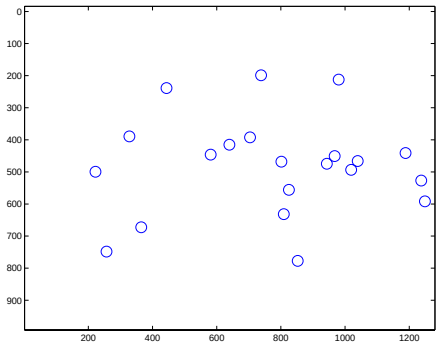
6.2.2 Mantle

A good initial estimate can lead to very fast convergence, as illustrated by the "mantle" image set. Again, 2D measurements are shown first in Figure 6.16, whereas the evolving marginal probabilities are shown as stacked stochastic matrices in Figure 6.17. By varying the number of iterations in successive runs, I found that the MCEM approach converged in as little as 10 iterations, provided a good initial estimate for structure and motion was available. In the "mantle" case this was obtained, as before, by initializing all the cameras at the same location and looking at the same point in space. However, in this case the structure was initialized on a plane at some arbitrary depth, using the measurements from an arbitrary image to create rays that were intersected with the plane. As a consequence the measurements and projected features coincide in that image (in this case image 3), as shown in Figure 6.18c. Since the other images were taken relatively nearby and with the same orientation (landscape), the displacements in the other images are also relatively small, which explains why the first E-step is so close to the ground truth.

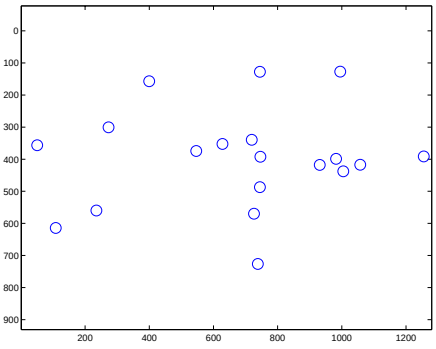
The input images with superimposed measurements and projected structure and motion estimate are shown in Figure 6.19.



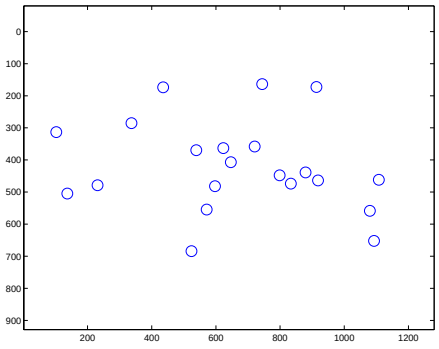
(a) image 1



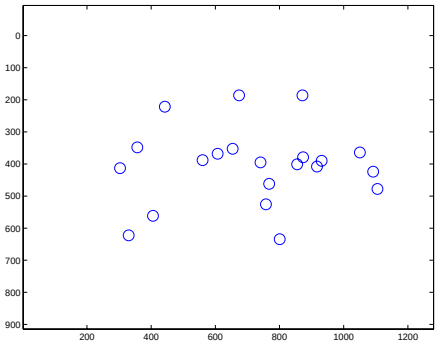
(b) image 2



(c) image 3

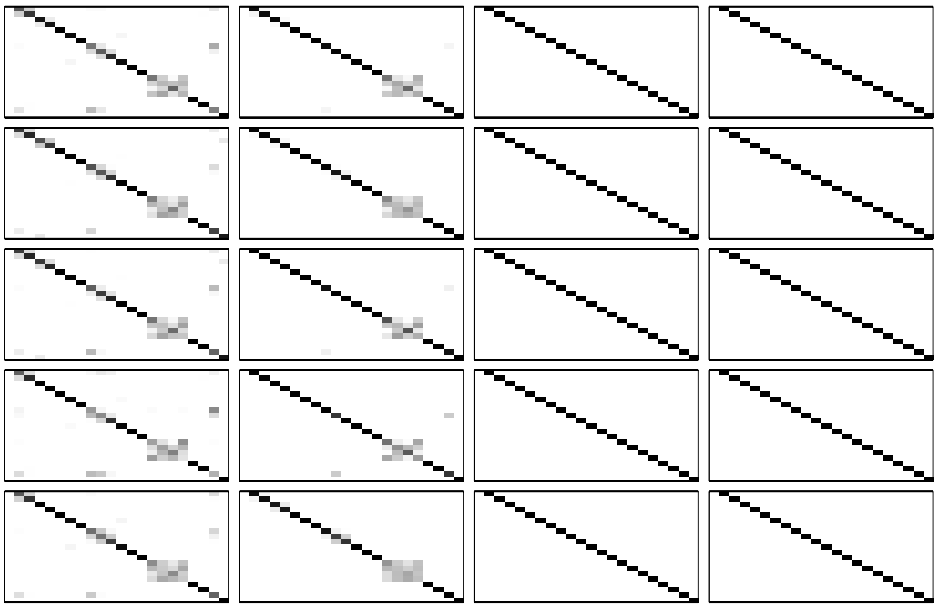


(d) image 4



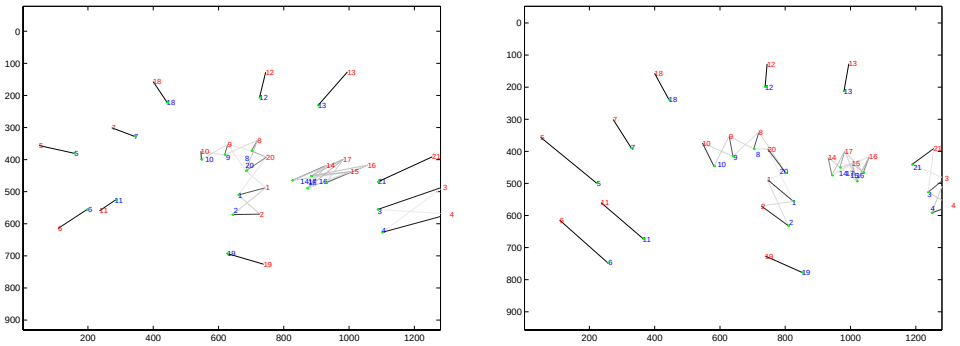
(e) image 5

Figure 6.16: Measurements in all 5 “mantle” input images.



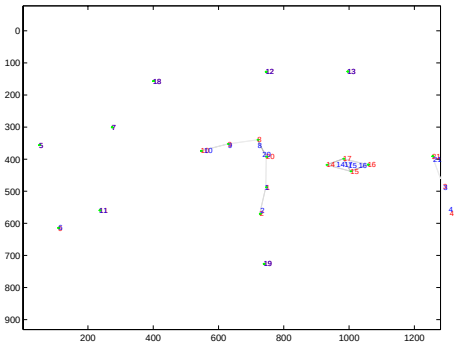
(a) it 1 (b) it 4 (c) it 7 (d) it 10

Figure 6.17: Marginal probabilities computed in the E-step (“mantle”).

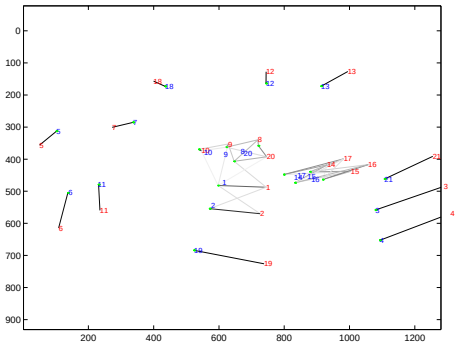


(a) image 1

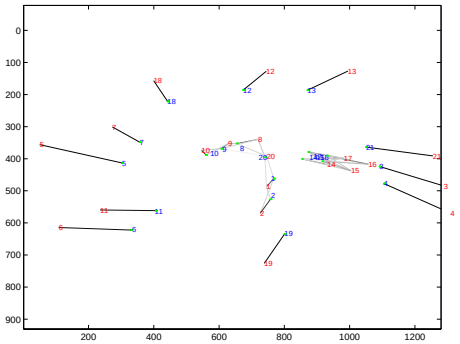
(b) image 2



(c) image 3



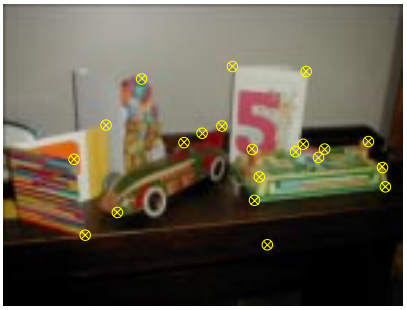
(d) image 4



(e) image 5

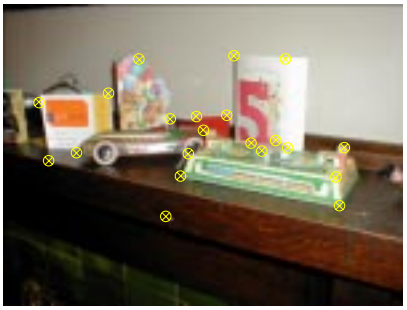
Figure 6.18: Projected features (red), marginal probabilities (grayscale edges), and virtual measurements (blue) in iteration 1 for the “mantle” image set. Note image 3 !

(a) image 1



(b) image 2

(c) image 3



(d) image 4

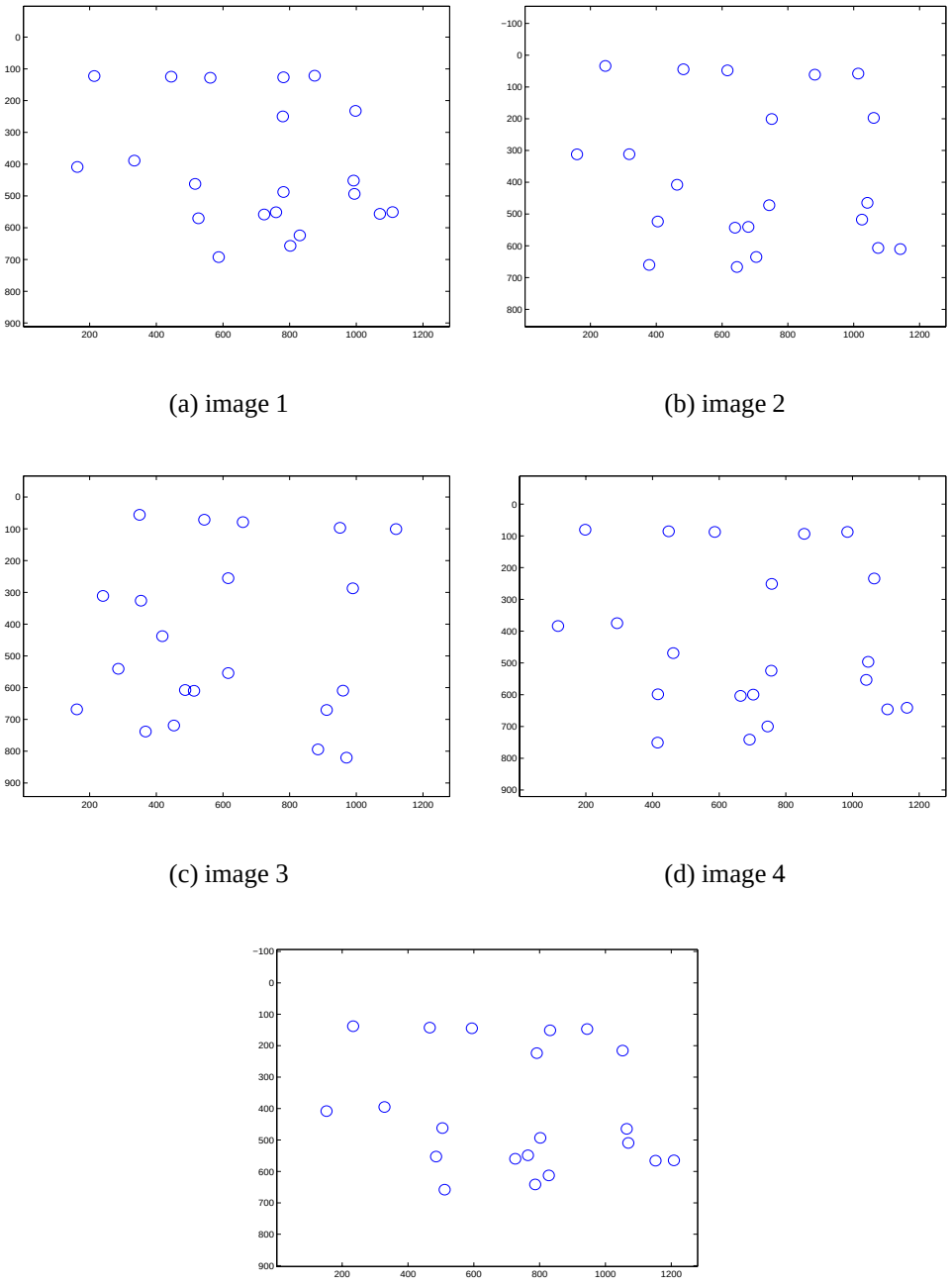
(e) image 5

Figure 6.19: The 5 original “mantle” input images. The last predicted location is marked with an asterisk. Measurements are shown as circles.

6.2.3 Desk

The same initialization method, initializing points on a plane using the measurements in one image, was applied to the "desk" image set, whose measurements are shown in Figure 6.20. The marginals estimated in the first iteration, shown in Figure 6.21a, have only partly consistent (soft) assignments. The scene is more difficult than the "mantle" scene, and takes longer to converge (25 iterations, in this case).

The more difficult searching executed by the EM algorithm (remember, EM does nothing but hill-climbing in likelihood space, using the lower-bounding mechanism) is illustrated most clearly in Figure 6.22, where the trajectories for the projected features are shown for the 5 images. The algorithm does finally converge however, and the end-result is shown, superimposed on the input images, in Figure 6.23.



(a) image 1

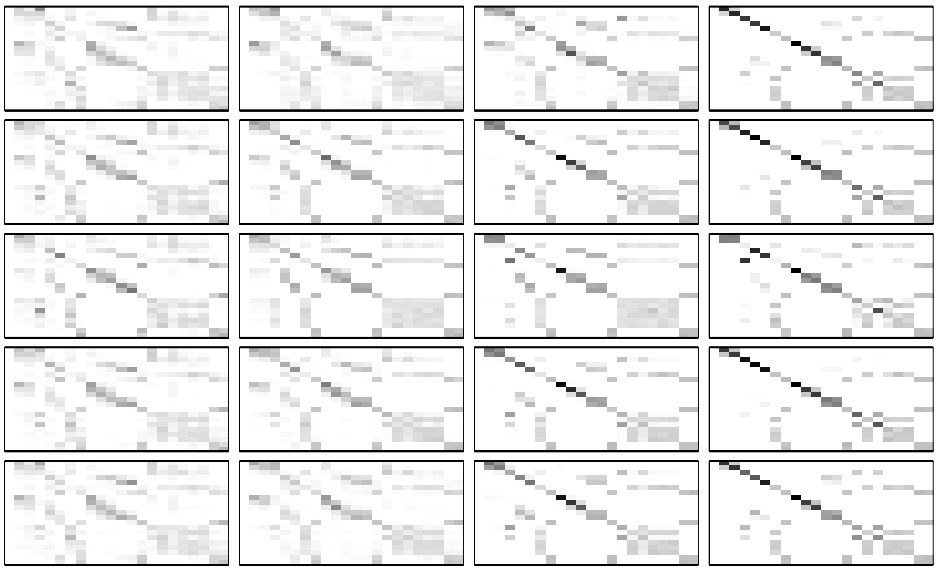
(b) image 2

(c) image 3

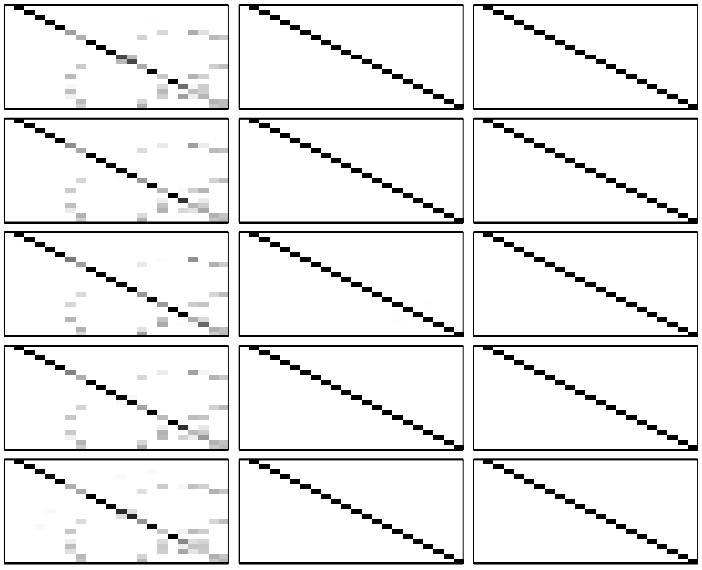
(d) image 4

(e) image 5

Figure 6.20: Measurements in the 5 “desk” input images.

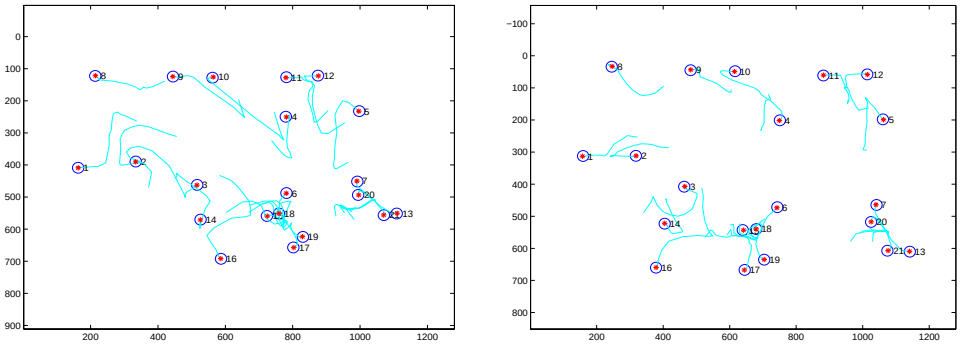


(a) it 1 (b) it 5 (c) it 9 (d) it 13



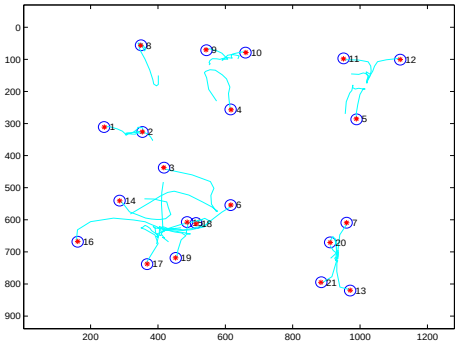
(e) it 17 (f) it 21 (g) it 25

Figure 6.21: Marginal probabilities computed in the E-step (“desk”).

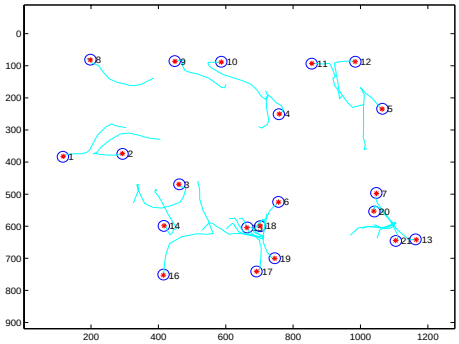


(a) image 1

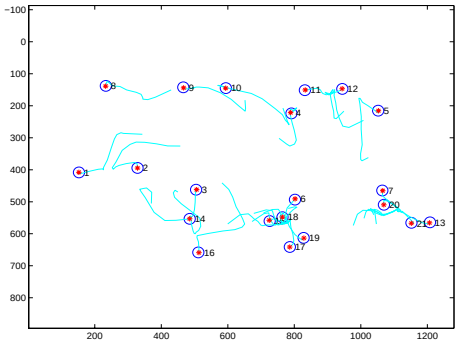
(b) image 2



(c) image 3

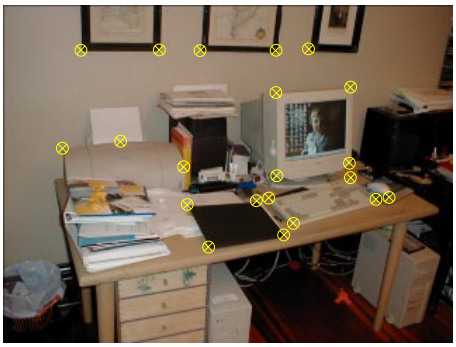


(d) image 4



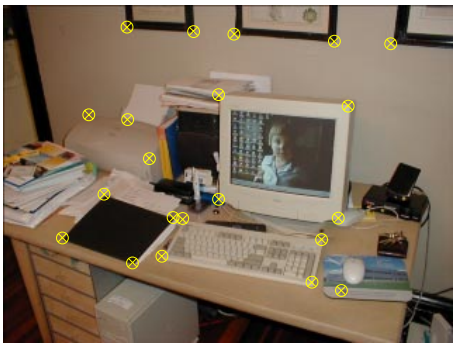
(e) image 5

Figure 6.22: Plot of the predicted location for each of the features over time in the 5 “desk” images. The last predicted location is marked with an asterisk. Measurements are shown as circles.

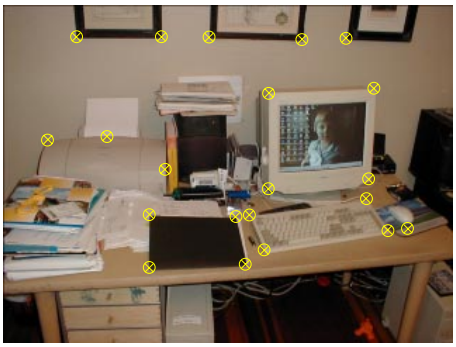


(a) image 1

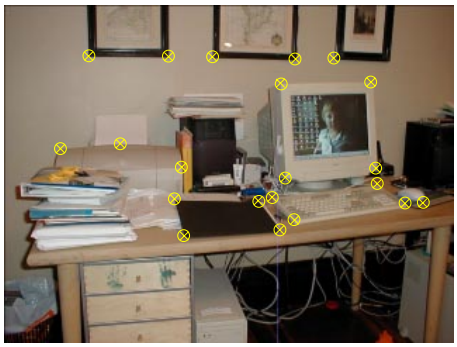
(b) image 2



(c) image 3



(d) image 4



(e) image 5

Figure 6.23: The 5 original “desk” input images. The last predicted location is marked with an asterisk. Measurements are shown as circles.

6.2.4 House

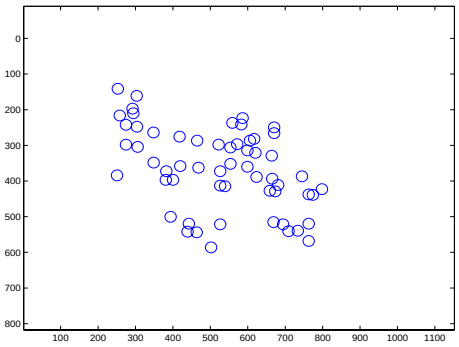
The EM algorithm can get stuck in local minima, and a typical way that this manifests itself is illustrated here using the "house" image set. This set consists of five images of the set of houses as the "townhouse" set shown before, but taken at a different time and from a wider range of viewpoints. As such, there is significant foreshortening due to the perspective projection. The measurements for this set are shown in Figure 6.24.

No matter how many times the algorithm is restarted with different initial conditions and/or annealing schedules (which also specifies the number of iterations the algorithm is run), there are remain small local mismatches in the final result. The evolution of the estimated 3D structure and the marginal probabilities for a typical run are shown in Figures 6.25 and 6.26, respectively. In this run, the algorithm was run for 50 iterations. Even though the gross structure is recovered very early and most detailed structure is recovered eventually, there are three features in the final structure estimate that are completely wrong. This can clearly be seen from the figures: looking at the 3D estimate in Figure 6.25f we see that the front roof and the rightmost window seem to have swapped vertices. Likewise, the marginal probabilities in the last iteration (Figure 6.26h) clearly show local mismatches in images 1,2, and 5.

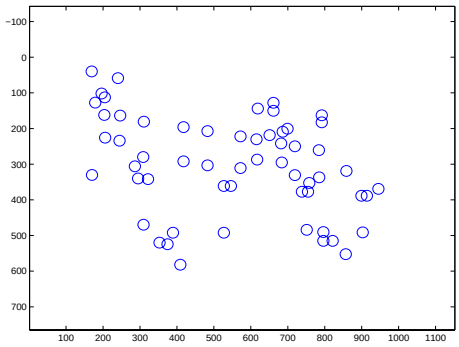
Because this is a typical manifestation of a local minimum, Figures 6.27 and 6.28 illustrate in more detail what exactly the mismatch is. From Figure 6.27, which shows the virtual measurements in the last iteration for images 1 and 5, it is clear that the problem is very local: the vast majority of the features, and also the motion, are estimated correctly. The problem seems to be limited to three features on the right hand-side of the images, where the gable of the roof and the rightmost window have switched position in the images. Figure 6.28 shows a close-up view of the problem area. The local minimum is this: the (wrongly) estimated structure for features $\mathbf{x}_7$, $\mathbf{x}_{39}$ and $\mathbf{x}_{40}$ is such that a three-way mismatch is by far the most likely correspondence. In turn, this correspondence causes the wrongly estimated structure estimate to persist. In other words, we are in a part of state space where the likelihood function is locally maximized, but we are in in fact not at the global maximum.

This particular run -with the local minimum problem- is further illustrated in Figures 6.29 and 6.30, respectively showing the trajectories of the projected features in image space, and the original input images with the final structure estimate superimposed on them.

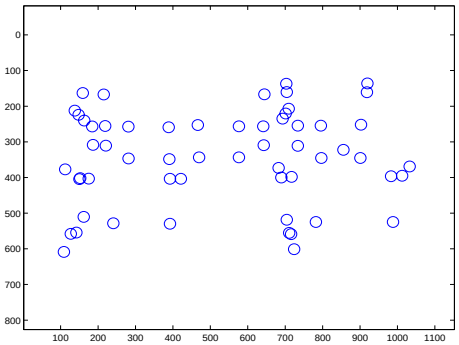
The final structure and motion estimate is very good, apart from the local mismatch between these three features. The motion estimate is only slightly biased by the local mismatch. In addition, the features for which the mismatch occurs is easily identified by looking at the



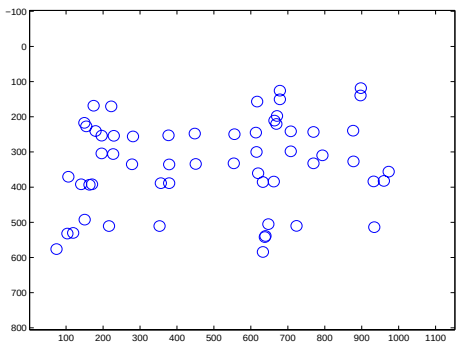
(a) image 1



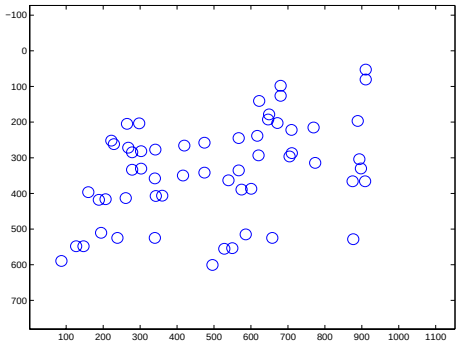
(b) image 2



(c) image 3

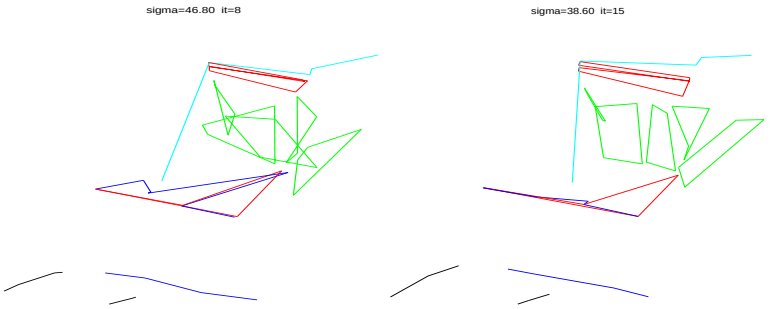


(d) image 4

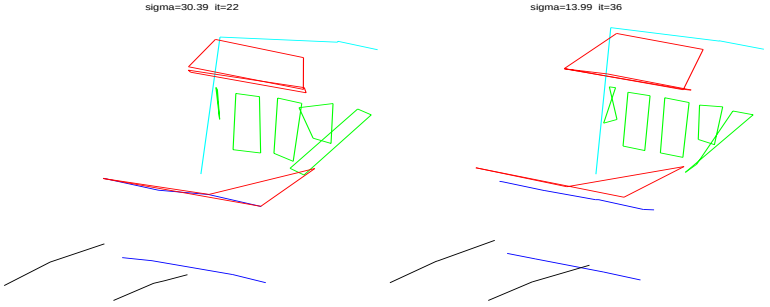


(e) image 5

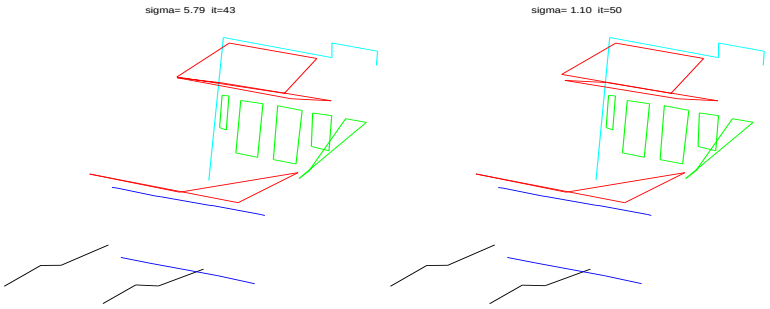
Figure 6.24: Measurements in the 5 “house” input images.



(a) it 8 (b) it 15

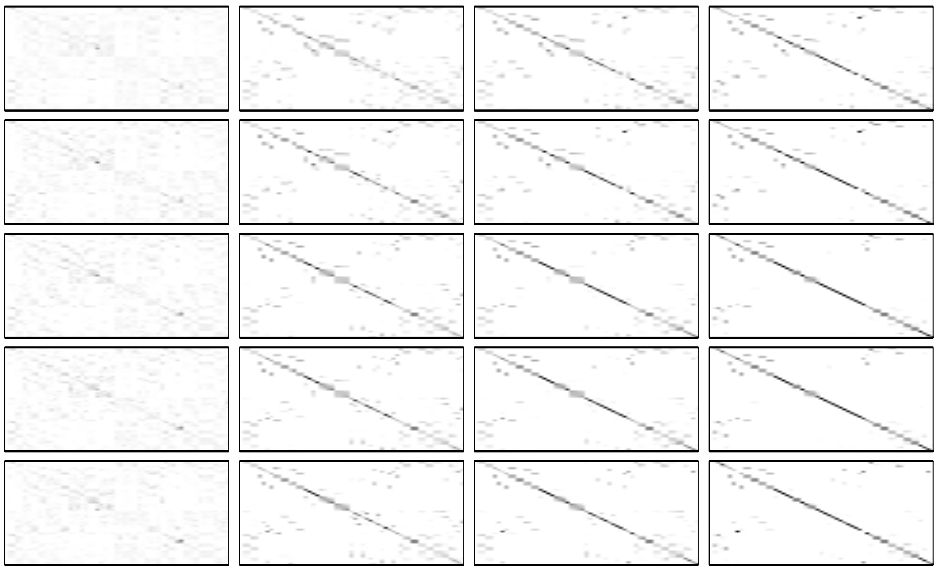


(c) it 22 (d) it 36

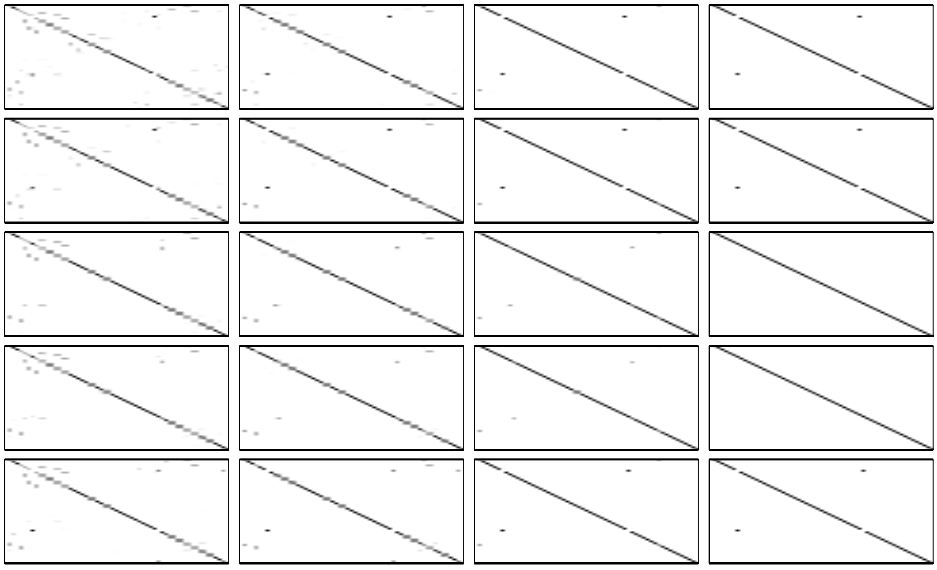


(e) it 43 (f) it 50

Figure 6.25: The structure estimate at successive iterations of the algorithm for the “house” image series.



(a) it 1 (b) it 8 (c) it 15 (d) it 22



(e) it 29 (f) it 36 (g) it 43 (h) it 50

Figure 6.26: Marginal probabilities computed in the E-step (“house”).

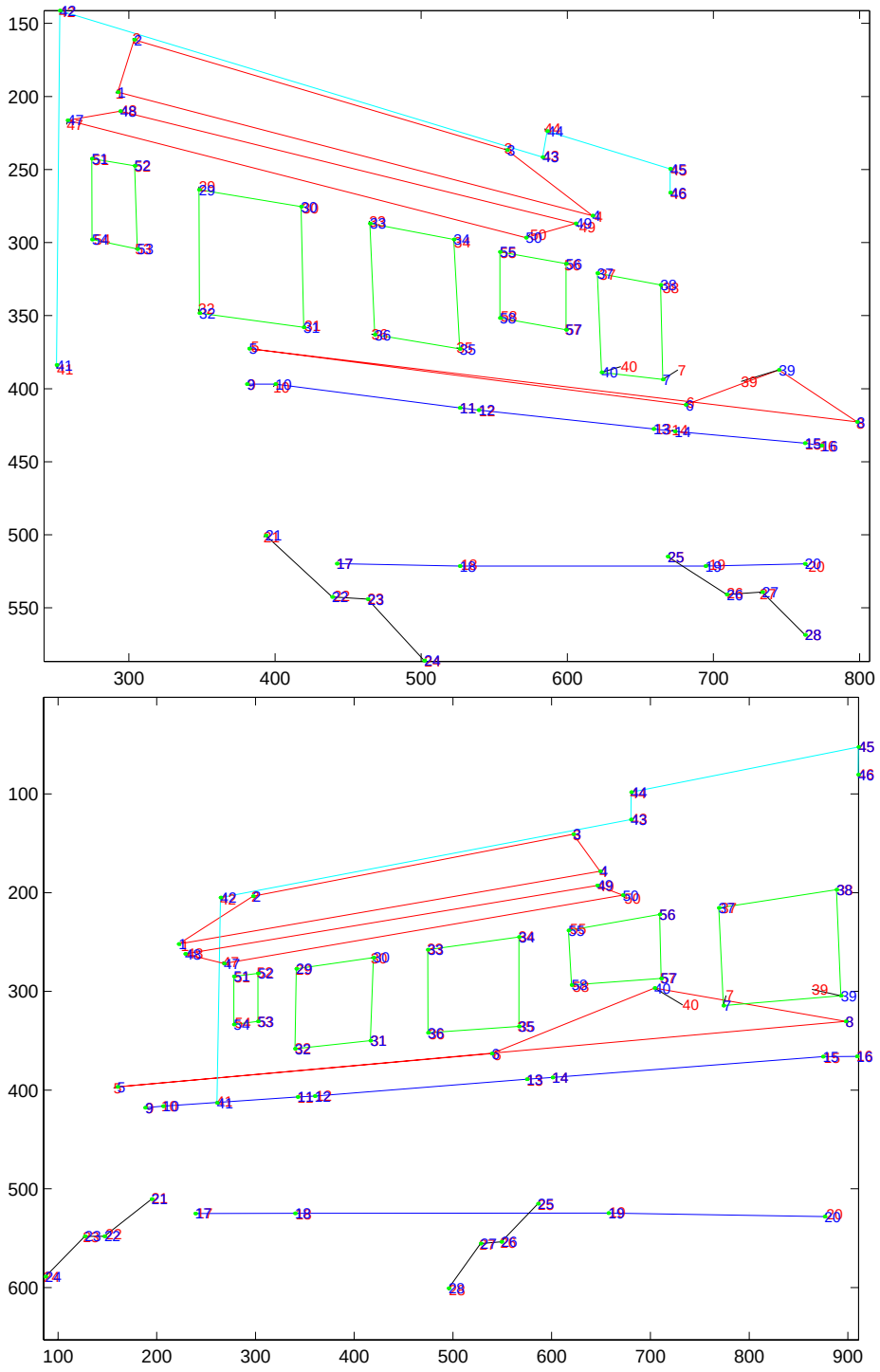


Figure 6.27: Projected features (red), marginal probabilities (grayscale edges), and virtual measurements (blue) in the last iteration for images 1 and 5. Note the three-way switch between features $\mathbf{x}_7$, $\mathbf{x}_{39}$ and $\mathbf{x}_{40}$.

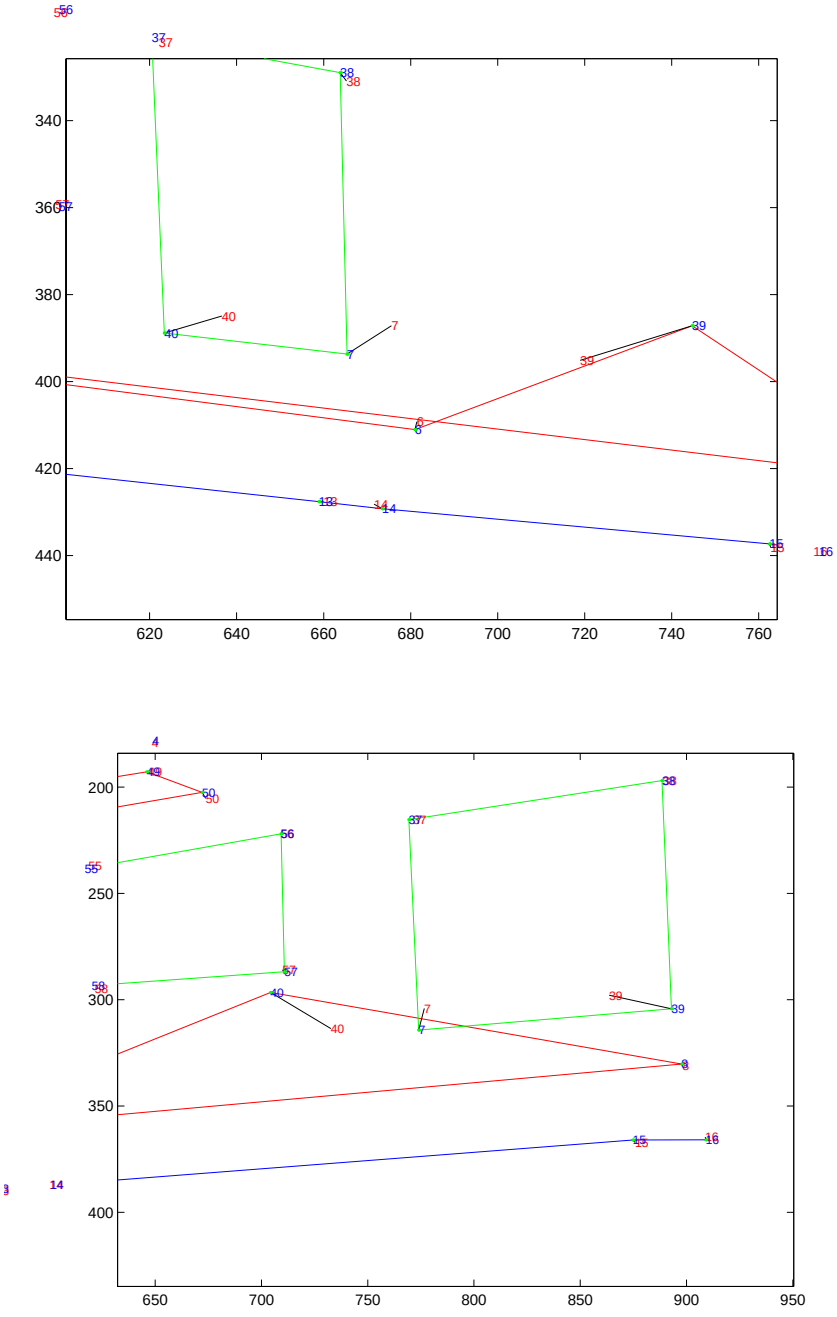
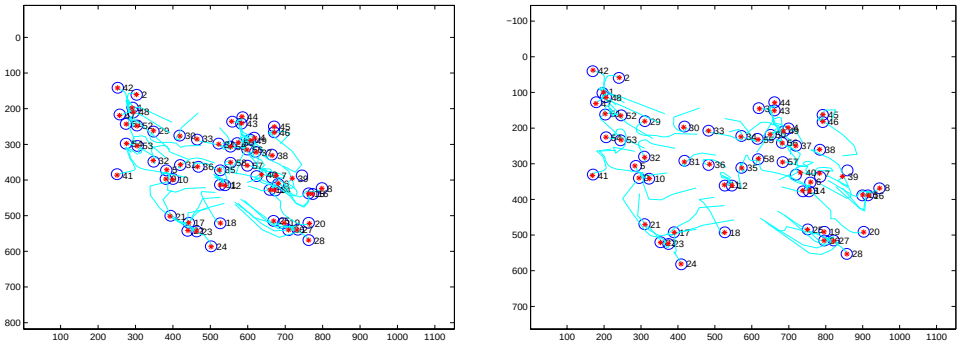
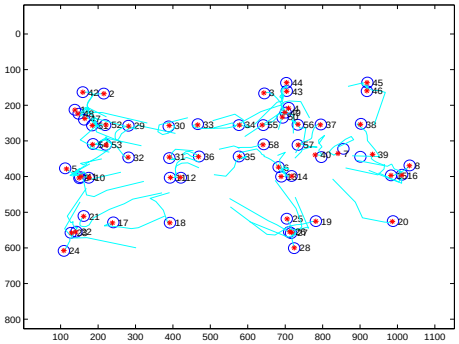


Figure 6.28: Version of Figure 6.27 that shows the three-way switch between features $\mathbf{x}_7$, $\mathbf{x}_{39}$ and $\mathbf{x}_{40}$ more clearly.

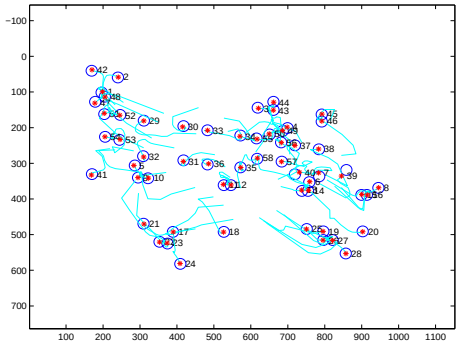


(a) image 1

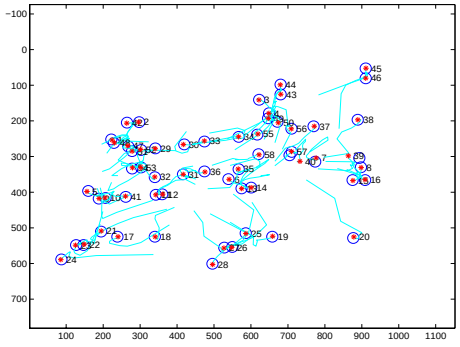
(b) image 2



(c) image 3



(d) image 2



(e) image 5

Figure 6.29: Plot of the predicted location for each of the features over time in the 5 “house” images. The last predicted location is marked with an asterisk. Measurements are shown as circles.

(a) image 1

(b) image 2

(c) image 3

(d) image 4

(e) image 5

Figure 6.30: The 5 original "house" input images. The last predicted location is marked with an asterisk. Measurements are shown as circles.

residuals for the corresponding virtual measurements. These two facts together make that the problem is actually easy to correct in a post-processing step: once the gross geometry is recovered, it is straightforward to correct local problems such as these.

A Local Mismatch Correction Scheme

One possible way to correct local mismatches such as the one illustrated above, relies on the fact that the motion estimate is overconstrained, even when mismatched features are omitted or discounted. Thus, a more accurate motion estimate can be obtained by either using a robust optimization scheme or simply omitting the mismatched features. After this is done we have the best possible motion estimate, given that we have not yet corrected the mismatch.

A possibly corrected structure estimate is then obtained by a RANSAC-like scheme. Since, when given the motion, the structure is determined by exactly two views, we can sample (or, if m is small, simply enumerate) from the possible image pairs, and select the pair that has the smallest reprojection error. This then, presumably, is a pair in which the mismatch does not occur (or, in the case of multiple problems, less mismatches occur).

Using the robustly estimated structure and motion $\widetilde{\Theta}$, we can run the E-step again, and re-estimate the structure and motion Θ in a last M-step. If the estimate $\widetilde{\Theta}$ was in the basin of attraction of the global maximum of the likelihood function, we are done. Otherwise, we can repeat the process.

The scheme, as described informally above, was implemented and works well to correct small, local mismatches, such as the one in this example.

Chapter 7

Occlusion and Clutter

In this chapter we extend the MCEM approach to handle occlusion and clutter. Self-occlusion of objects and occluding objects can mean that certain features are not visible in all of the images. In addition, the feature detector might miss some of the features, even if they are visible. Finally, a feature detector can generate spurious measurements, i.e., report a feature where there is none. All these processes can be modeled probabilistically. Clutter has been modeled before in the tracking literature. Occlusion has been studied less, and I explore some of the options before settling on a simple visibility model that allows for easy inference.

By allowing occlusion and clutter, the space of possible correspondence vectors $\mathbf{J}$ expands dramatically. Even if one assumes, as I do below, that the number of features n is known, the number of measurements K_i in the images can be different for each image and in general different from n. Only a subset of features might be visible in each image, and a subset of the measurements might be spurious. It is easily seen that enumerating all possible ways in which the subsets can be chosen and combined leads to a combinatorial explosion.

Fortunately, sampling over a larger space to approximate the E-step is no harder than sampling over the space of assignments, provided one can evaluate the probability of each correspondence vector up to a constant. It is shown in this chapter that for a simple visibility model, the posterior probability $f_i(\mathbf{j}_i)$ of a given correspondence vector $\mathbf{j}_i$ in one image is given by the following simple expression:

$$f_i(\mathbf{j}_i) \propto \alpha^{S_i} \exp\left[-w(\mathbf{j}_i)\right]$$

where α is a constant that depends on the amount of occlusion and clutter, and S_i is the

number of spurious measurements in image i. The second factor is simply the Gibbs distribution that favors measurements close to their predicted location under $\mathbf{j}_i$, as in Chapter 5 (equation 5.8 on page 71).

In this chapter we derive this expression and then show how the MCMC sampling algorithm can be adapted to sample over the larger space of correspondence vectors. In the following section, Section 7.1, we first examine the correspondence in one image. The knowledge we have about the amount of occlusion and clutter can be modeled with varying degrees of sophistication, which is discussed in Section 7.2. The result is that we can formulate a prior over correspondences, based on their degree of occlusion and clutter. When combined with a measurement model, we can finally derive the probability for any given correspondence vector $\mathbf{J}$, which is done in Section 7.3. Finally, sampling over these correspondence vectors is discussed in the last section, Section 7.4.

7.1 Correspondence in one Image

The key assumption underlying this chapter is that, given a specific degree of occlusion and clutter in one image, all correspondence assignments are otherwise equally likely. Intuitively, we expect a certain amount of occluded features and a certain amount of clutter, but correspondence assignments $\mathbf{J}$ that have too much or too little of each are less probable. That is expanded upon in the next section. However, it is clear that we have no reason *a priori* to favor some correspondences over others, if they have the *same* degree of clutter or occlusion. For the case when occlusion and clutter were not an issue, we assumed that all $n!$ possible image correspondence vectors $\mathbf{j}_i$ were equally likely. Similarly, and this is a key assumption, in the present case we assume that all correspondence vectors $\mathbf{j}_i$ are equally likely *a priori*, once we know (a) which features $\mathbf{x}_j$ are detected in image i and (b) how many spurious measurements there are.

To make this assumption more explicit, we define two key random variables that help quantify the degree of occlusion and clutter. Specifically, suppose the number of measurements in image i is equal to K_i, then we define

- $D_i \triangleq$ the number features detected in image i, where $0 \leq D_i \leq n$.
- $S_i \triangleq$ the number of spurious measurements in image i, where $0 \leq S_i \leq K_i$.

Clearly, we have $K_i = D_i + S_i$: a measurement is either spurious or corresponds to a detected feature.

However, in order for two correspondence vectors to be equally likely a priori, they also need to have exactly the *same* features $\mathbf{x}_j$ detected in each. Indeed, one can imagine a probabilistic model for occlusion that accords a higher probability of being occluded to a specific feature (see below, in Section 7.2.2). To model this, we need to know exactly which features were detected. Thus, let us define for each image i the *image detection vector* $\mathbf{d}_i$, where $i \in 1..m$. Each $\mathbf{d}_i$ is defined to be an n-dimensional vector of booleans d_{ij}, indicating for each feature $\mathbf{x}_j$ whether it was detected in image i or not:

$$\mathbf{d}_i \triangleq \{d_{ij} | j \in 1..n\}$$

We also define the aggregate detection vector $\mathbf{D}$ that spans all images:

$$\mathbf{D} \triangleq \{\mathbf{d}_i | i \in 1..m\}$$

Finally, we can now specify the prior probability of a given assignment vector $\mathbf{j}_i$, given the detection vector $\mathbf{d}_i$ and the number of spurious measurements S_i, as:

$$P(\mathbf{j}_i | \mathbf{d}_i, S_i) = Comp(\mathbf{j}_i, \mathbf{d}_i) \frac{1}{N_{\mathbf{J}}^{S_i, D_i}}$$

where $Comp(\mathbf{j}_i, \mathbf{d}_i)$ is an indicator function denoting whether $\mathbf{j}_i$ is compatible with $\mathbf{d}_i$, and $N_{\mathbf{J}}^{S_i, D_i}$ is the number of compatible assignment vectors containing S_i zeros and D_i detected feature indices. Since there are $\binom{S_i + D_i}{S_i}$ ways to choose the arrangement of the zeros, and for each arrangement we have $D_i!$ different ways of permuting the detected feature indices, we have

$$N_{\mathbf{J}}^{S_i, D_i} = \binom{S_i + D_i}{S_i} D_i! = \frac{(S_i + D_i)!}{S_i!} = \frac{K_i!}{S_i!}$$

Thus, we get

$$P(\mathbf{j}_i | \mathbf{d}_i, S_i) = Comp(\mathbf{j}_i, \mathbf{d}_i) \frac{S_i!}{(S_i + D_i)!} = Comp(\mathbf{j}_i, \mathbf{d}_i) \frac{S_i!}{K_i!} \tag{7.1}$$

Note that, if one would want to *generate* random assignment vectors $\mathbf{j}_i$, one needs the pattern of detected features $\mathbf{d}_i$: simply knowing the number of detected features D_i is not sufficient.

7.2 Detection, Visibility and Clutter

In this section we calculate how probable it is a priori that a specific image correspondence vector $\mathbf{j}_i$ is observed. It will be shown that, for a simple visibility model, this prior probability $P(\mathbf{j}_i)$ is given by

$$P(\mathbf{j}_i) = \frac{e^{-\lambda}}{(S_i + D_i)!}\lambda^{S_i} q^{D_i}(1-q)^{n-D_i} \tag{7.2}$$

where λ is the expected number of spurious measurements, and q is the the combined visibility-detection probability. Both these quantities are defined and discussed below. Expression 7.2 can then be combined with a measurement model in order to obtain a posterior probability over image correspondence vectors.

Given the key assumption made in the previous section, we can reduce the calculation of the prior $P(\mathbf{j}_i)$ to calculating the probability of a specific set of detected features $\mathbf{D}$ in combination with a specific number of spurious measurements S_i in each image. Note however that, in general, we can no longer treat the images in isolation. Indeed, we get the following expression for the conditional prior $P(\mathbf{J}|\mathbf{M}, \mathbf{X})$ over correspondence vectors $\mathbf{J}$:

$$P(\mathbf{J}|\mathbf{M}, \mathbf{X}) = P(\mathbf{D}, S_1, .., S_m|\mathbf{M}, \mathbf{X}) \prod_{i=1}^{m} P(\mathbf{j}_i|\mathbf{d}_i, S_i) \tag{7.3}$$

We have not yet made any assumptions that allow us to decompose the first factor, the prior on detection and number of spurious features $P(\mathbf{D}, S_1, .., S_m|\mathbf{M}, \mathbf{X})$, over the images. In fact, several realistic models are possible where this cannot be done. The final expression (7.2) is only valid for a specific, simple visibility model that disregards possible correlation between neighboring images.

In general, it is reasonable to model the occlusion and detection process separately from clutter, and model clutter as independent of the structure imaged. This is formalized by the following assumptions:

- Detection of features is independent of clutter. While in high-clutter situations it might be harder to pick out which measurements $\mathbf{u}_{ik}$ correspond to real features, the number or identity of detected features is not affected. In terms of probability distributions, this is expressed as

$$P(\mathbf{D}, S_1, .., S_m|\mathbf{M}, \mathbf{X}) = P(\mathbf{D}|\mathbf{M}, \mathbf{X})P(S_1, .., S_m|\mathbf{M}, \mathbf{X}) \tag{7.4}$$

- The number of spurious features in the images is independent of the structure $\mathbf{X}$. However, we keep the possible dependence on motion $\mathbf{M}$ explicit for now, so we can model correlation between the S_i in images taken closely together. In terms of probability distributions, this second assumption implies

$$P(S_1, .., S_m|\mathbf{M}, \mathbf{X}) = P(S_1, .., S_m|\mathbf{M}) \tag{7.5}$$

Under these assumptions, we get the following expression for the prior probability $P(\mathbf{J}|\mathbf{M}, \mathbf{X})$ on aggregate correspondence vectors $\mathbf{J}$ (by substituting (7.4) and (7.5) into (7.3)):

$$P(\mathbf{J}|\mathbf{M}, \mathbf{X}) = P(\mathbf{D}|\mathbf{M}, \mathbf{X})P(S_1, .., S_m|\mathbf{M}) \prod_{i=1}^{m} P(\mathbf{j}_i|\mathbf{d}_i, S_i)$$

Below we first examine the detection process $P(\mathbf{D}|\mathbf{M}, \mathbf{X})$ in further detail, then the clutter $P(S_1, .., S_m|\mathbf{M})$.

7.2.1 Detection

The question of whether a feature $\mathbf{x}_j$ is measured in image i, i.e., the value of d_{ij}, can be regarded as the answer to two separate questions: (a) is the feature $\mathbf{x}_j$ actually *visible* in image i, and (b) in the case it is visible, is it then actually detected by the measurement process? The latter question reflects the fact that feature detection algorithms are in general not infallible.

We model *visibility* $\mathbf{v}_i$ in a given image i as an n-dimensional Boolean vector

$$\mathbf{v}_i \triangleq \{v_{ij}|j \in 1..n\}$$

where each bit v_{ij} indicates whether feature $\mathbf{x}_j$ is visible in the image positioned at $\mathbf{m}_i$. We define the *aggregate visibility vector* $\mathbf{V}$ as the collection of all $\mathbf{v}_i$:

$$\mathbf{V} \triangleq \{\mathbf{v}_i|i \in 1..m\}$$

For simplicity we model each visible feature to have a fixed probability δ of being detected when visible. This yields the following conditional probability of detection d_{ij} when *given* visibility v_{ij}, in table format:

v_{ij}	d_{ij}	$P(d_{ij}\|v_{ij})$
1	1	δ
1	0	$1-\delta$
0	1	0
0	0	1

Since v_{ij} and d_{ij} are both boolean variables this can be written compactly as:

$$P(d_{ij}|v_{ij}) = \delta^{v_{ij}d_{ij}}(1-\delta)^{v_{ij}(1-d_{ij})}0^{(1-v_{ij})d_{ij}} \quad (7.6)$$

$$= \delta^{v_{ij}d_{ij}}(1-\delta)^{v_{ij}(1-d_{ij})}v_{ij}^{d_{ij}} \quad (7.7)$$

where the second equality can be easily verified using a truth table. The second form (7.7) is convenient to obtain the probability of a specific detection vector $\mathbf{d}_i$ given a visibility vector $\mathbf{v}_i$:

$$P(\mathbf{d}_i|\mathbf{v}_i) = \delta^{D_i}(1-\delta)^{V_i-D_i}\prod_{j=1}^{n} v_{ij}^{d_{ij}} \quad (7.8)$$

where D_i denotes the number of detected features, and V_i is the number of visible features in image i. The rightmost product above indicates whether $\mathbf{d}_i$ is compatible with $\mathbf{v}_i$ in terms of visibility: it is zero *iff* there exists a feature $\mathbf{x}_j$ for which $v_{ij} = 0$ and $d_{ij} = 1$. In other words, invisible features are assumed undetectable.

There might be imaging situations where a more sophisticated detection model is called for. For example, one application involves reconstructing the shape of an asteroid, where the detected features are craters on the asteroid's surface. Crater-shaped features are less likely to be detected on the shadow side of the asteroid, and this could be modeled by conditioning the probability of detection on the imaging situation $\mathbf{\Theta}$. An alternative solution is to include this effect in the calculation of visibility $\mathbf{v}_i$, and by convention reserve the detection process to effects that do not depend on $\mathbf{\Theta}$. This is the approach we take here.

The probability of a given detection vector $\mathbf{D}$ given $\mathbf{M}$ and $\mathbf{X}$ is then obtained by summing over all possible visibility configurations $\mathbf{V}$:

$$P(\mathbf{D}|\mathbf{M},\mathbf{X}) = \sum_{\mathbf{V}} P(\mathbf{V}|\mathbf{M},\mathbf{X})\prod_{i=1}^{m} P(\mathbf{d}_i|\mathbf{v}_i) \quad (7.9)$$

with $P(\mathbf{d}_i|\mathbf{v}_i)$ defined as above in equation (7.8). What is left is to model the probability of a visibility vector $\mathbf{V}$ by means of the conditional prior $P(\mathbf{V}|\mathbf{M},\mathbf{X})$, which is done below.

7.2.2 Visibility

There are several ways to model visibility, with varying degrees of sophistication: (a) using an MRF, (b) assuming conditional independence, (c) assuming a fixed probability of being visible. Below we mostly use the latter (simplest) model, keeping in mind that any of the more sophisticated models can be used if warranted. Note that since the model for

visibility is part of the prior it is often not critical that it is accurate. Once we condition on the measurements $\mathbf{U}_i$, the prior is most likely swamped out in the calculation of the posterior.

Below I discuss each of the three visibility models. For the two simpler models we can simplify the expression (7.9) for the prior $P(\mathbf{D}|\mathbf{M},\mathbf{X})$ on detection.

Using a Markov Random Field

Using a Markov random field (MRF) (as e.g. in (MacCormick and Blake, 1998))

$$P(\mathbf{V}|\mathbf{M},\mathbf{X}) = \frac{1}{Z}\exp\left[-\sum_c U(\mathbf{V}_c|\mathbf{M},\mathbf{X})\right]$$

where c are the cliques of a suitably defined neighborhood system, U is an MRF potential function, and Z is a normalization constant. This allows modeling intuitive knowledge such as the fact that features tend to be either both visible or both invisible in neighboring images, or that features close together are likely to be occluded together.

Conditionally Independent Visibility

A simplification is to assume that, given $\mathbf{M}$ and $\mathbf{X}$, the visibility values v_{ij} of individual features are conditionally independent, leading to

$$P(\mathbf{V}|\mathbf{M},\mathbf{X}) = \prod_{i,j} P(v_{ij}|\mathbf{m}_i,\mathbf{x}_j) \tag{7.10}$$

If we substitute (7.10) and (7.7) into (7.9) and simplify we obtain a particularly simple expression for the prior probability of a specific detection vector $\mathbf{D}$:

$$\begin{aligned}
P(\mathbf{D}|\mathbf{M},\mathbf{X}) &= \sum_{\mathbf{V}}\prod_{i,j} P(v_{ij}|\mathbf{m}_i,\mathbf{x}_j)\delta^{v_{ij}d_{ij}}(1-\delta)^{v_{ij}(1-d_{ij})}v_{ij}^{d_{ij}} && (7.11)\\
&= \prod_{i,j}\sum_{v_{ij}=0}^{1} P(v_{ij}|\mathbf{m}_i,\mathbf{x}_j)\delta^{v_{ij}d_{ij}}(1-\delta)^{v_{ij}(1-d_{ij})}v_{ij}^{d_{ij}} && (7.12)\\
&= \prod_{d_{ij}=1} P_{ij}\delta \prod_{d_{ij}=0}\left(P_{ij}(1-\delta)+(1-P_{ij})\right) && (7.13)\\
&= \prod_{d_{ij}=1} P_{ij}\delta \prod_{d_{ij}=0}\left(1-P_{ij}\delta\right) && (7.14)
\end{aligned}$$

where we defined $P_{ij} \stackrel{\Delta}{=} P(v_{ij} = 1|\mathbf{m}_i, \mathbf{x}_j)$ for notational convenience.

As a practical example consider the case of a 2D robot mapping application, where the $\mathbf{x}_j$ are 2D landmark locations and $\mathbf{m}_i$ robot location. Here a reasonable model is to assume that the visibility v_{ij} of a feature $\mathbf{x}_j$ depends only on its distance $d(\mathbf{x}_j, \mathbf{m}_i)$ to $\mathbf{m}_i$:

$$P(v_{ij}|\mathbf{m}_i, \mathbf{x}_j) = P(v_{ij}|d(\mathbf{x}_j, \mathbf{m}_i))$$

Fixed Probability of Visibility

Even simpler is to drop the dependence on $\mathbf{M}$ and $\mathbf{X}$ altogether, and have a fixed probability ν of being visible for each feature $\mathbf{x}_j$

$$P(v_{ij}|\mathbf{m}_i, \mathbf{x}_j) = \nu \tag{7.15}$$

so that, with V_i the number of features visible in image i, the probability of a given image visibility vector $\mathbf{v}_i$ becomes

$$P(\mathbf{v}_i|\mathbf{M}, \mathbf{X}) = P(\mathbf{v}_i|\nu) = \nu^{V_i}(1-\nu)^{n-V_i} \tag{7.16}$$

and

$$P(\mathbf{V}|\mathbf{M}, \mathbf{X}) = P(\mathbf{V}|\nu) = \prod_{i=1}^{m} P(\mathbf{v}_i|\nu)$$

Note that we can condition the occlusion probability ν on the type of environment or object that is being observed. It can be set by hand, estimated from data, or even included in $\mathbf{\Theta}$ as a parameter to be estimated by EM.

Substituting (7.15) in (7.14) yields the following expression for a specific detection vector $\mathbf{D}$:

$$\begin{aligned} P(\mathbf{D}|\mathbf{M}, \mathbf{X}) &= \prod_{d_{ij}=1} \nu\delta \prod_{d_{ij}=0} (1-\nu\delta) \\ &= \prod_i P(\mathbf{d}_i|\mathbf{M}, \mathbf{X}) \end{aligned}$$

with

$$P(\mathbf{d}_i|\mathbf{M}, \mathbf{X}) = (\nu\delta)^{D_i}(1-\nu\delta)^{n-D_i} = q^{D_i}(1-q)^{n-D_i} \tag{7.17}$$

where we defined q as the combined visibility-detection probability $q \stackrel{\Delta}{=} \nu\delta$. This result is of course obvious in retrospect: it is simply the probability that D_i detected features in image i were visible *and* detected, multiplied with the probability of the remaining $n - D_i$ features to be either occluded or undetected.

7.2.3 Clutter

In this section we look at the process of clutter, modeled by $P(S_1, .., S_m|\mathbf{M})$. Note that here we are only concerned with the prior probability of the *number* of spurious measurements S_i in each image i. Reasoning about the location of these clutter measurements cannot be done without reference to $\mathbf{U}$, i.e., this shows up in the calculation of the posterior.

Again we have a choice to model this using models of varying complexity:

- Using a Markov random field that models the fact that the number of spurious measurements might be correlated between neighboring images.
- Using a simpler model that neglects this possible dependence (and any possible dependence on $\mathbf{M}$), and regards clutter as independent in all images:

$$P(S_1, .., S_m|\mathbf{M}) = \prod_{i=1}^{m} P(S_i) \tag{7.18}$$

Below the simpler model is used, with the distribution over S_i governed by a *Poisson process with intensity* γ

$$P(S_i; A, \gamma) = \frac{(A\gamma)^{S_i} e^{-A\gamma}}{S_i!}$$

where A is the image area. Poisson processes are the standard way of modeling clutter in the tracking literature, see e.g. (Popoli and Blackman, 1999). The intensity γ is to be interpreted as the expected number of spurious measurements per unit area. In many cases it is easier to directly specify the expected number of spurious measurements

$$\lambda \triangleq E\{S_i\} = A\gamma$$

and the prior is written as

$$P(S_i; \lambda) = \frac{\lambda^{S_i} e^{-\lambda}}{S_i!} \tag{7.19}$$

7.2.4 A Prior on Correspondence

Putting all these results together, we can now formulate a prior on correspondence vectors $\mathbf{J}$. Assuming conditional independence of visibility (7.10) and clutter (7.18) the prior factors over the different images as

$$P(\mathbf{J}|\mathbf{M}, \mathbf{X}) = \prod_{i=1}^{m} P(\mathbf{j}_i|\mathbf{m}_i, \mathbf{X})$$

with the following generic prior on image correspondence vectors $\mathbf{j}_i$:

$$P(\mathbf{j}_i|\mathbf{m}_i, \mathbf{X}) = P(\mathbf{d}_i|\mathbf{m}_i, \mathbf{X})P(S_i)P(\mathbf{j}_i|\mathbf{d}_i, S_i)$$

Note that this does *not* hold in case a Markov random field is used for either visibility or clutter. In that case the prior cannot be easily factored.

Substituting expression (7.1) an (7.19) respectively for the correspondence and clutter priors, and simplifying, we obtain

$$\begin{aligned} P(\mathbf{j}_i|\mathbf{m}_i, \mathbf{X}) &= P(\mathbf{d}_i|\mathbf{m}_i, \mathbf{X}) \left(\frac{\lambda^{S_i} e^{-\lambda}}{S_i!}\right) \left(Comp(\mathbf{j}_i, \mathbf{d}_i)\frac{S_i!}{K_i!}\right) \\ &= \frac{e^{-\lambda}}{K_i!}\lambda^{S_i} P(\mathbf{d}_i|\mathbf{m}_i, \mathbf{X}) \\ &= \frac{e^{-\lambda}}{(S_i + D_i)!}\lambda^{S_i} P(\mathbf{d}_i|\mathbf{m}_i, \mathbf{X}) \end{aligned}$$

where $Comp(\mathbf{j}_i, \mathbf{d}_i) = 1$ is assumed, as $\mathbf{d}_i$ above is *computed* from $\mathbf{j}_i$.

Finally, if the simple visibility model (7.15) is used, we can substitute (7.17) for the detection probability $P(\mathbf{d}_i|\mathbf{m}_i, \mathbf{X})$ and we obtain the following final expression for the prior on image correspondence vectors $\mathbf{j}_i$:

$$P(\mathbf{j}_i|\mathbf{m}_i, \mathbf{X}) = P(\mathbf{j}_i|n) = \frac{e^{-\lambda}}{(S_i + D_i)!}\lambda^{S_i} q^{D_i}(1-q)^{n-D_i} \tag{7.20}$$

where $0 \leq D_i \leq n$ for all valid configurations, as before $q \triangleq \nu\delta$ is the combined visibility-detection probability, and both S_i and D_i can be readily computed from $\mathbf{j}_i$. Note that for this model (the simplest visibility model) the dependence on $\mathbf{m}_i$ disappears, and the dependence on $\mathbf{X}$ is only through the number of features n.

As a sanity check, we can calculate the expected number of spurious and detected measurements. It is easily seen that, under the distribution (7.20) the expected number of spurious measurements $E[S_i] = \lambda$, the expected number of detected features $E[D_i] = qn$, and by linearity of expectation $E[K_i] = E[S_i + D_i] = \lambda + qn$.

Boundary Cases

It is both instructive and of interest to examine some specific values for q and λ more closely. The case where $q = 0$, i.e., no features are ever detected, is not of practical

interest. However, for $q = 1$, the case where every feature is visible and reliably detected, we have $D_i = n$ and $K_i \geq n$. In this case the prior becomes

$$P(\mathbf{j}_i|n, q = 1) = \frac{e^{-\lambda}}{(S_i + n)!}\lambda^{S_i}$$

If $\lambda = 0$, i.e., there are no spurious features, the number of measurements K_i is equal to D_i, the number of detected measurements. Furthermore, we have $K_i = D_i \leq n$. The prior becomes

$$P(\mathbf{j}_i|n, \lambda = 0) = \frac{1}{D_i!}q^{D_i}(1 - q)^{n-D_i}$$

Finally, when both $\lambda = 0$ and $q = 1$ we have the familiar case where all features are visible in all images, and there are no spurious measurements. In this case $K_i = D_i = n$, $S_i = 0$, and the prior reverts to

$$P(\mathbf{j}_i|n, q = 1, \lambda = 0) = \frac{1}{n!}$$

7.3 The Probability of Correspondence Vectors

To sample over correspondence vectors $\mathbf{J}$, we need to evaluate their probability. Now that a prior $P(\mathbf{J}|\mathbf{M}, \mathbf{X})$ over correspondence vectors $\mathbf{J}$ is available, we can use Bayes law to calculate the posterior probability $P(\mathbf{J}|\mathbf{U}, \mathbf{M}, \mathbf{X})$:

$$P(\mathbf{J}|\mathbf{U}, \mathbf{M}, \mathbf{X}) \propto P(\mathbf{U}|\mathbf{J}, \mathbf{M}, \mathbf{X})P(\mathbf{J}|\mathbf{M}, \mathbf{X})$$

As shown below, the likelihood will be of the form

$$P(\mathbf{U}|\mathbf{J}, \mathbf{M}, \mathbf{X}) = \prod_i A^{-S_i} e^{-w(\mathbf{j}_i)} \tag{7.21}$$

where A is the image area. This expression, when combined with the simple visibility prior (7.20), yields a very simple form for the posterior:

$$P(\mathbf{J}|\mathbf{U}, \mathbf{M}, \mathbf{X}) \propto \prod_i \alpha^{S_i} \exp\left[-w(\mathbf{j}_i)\right] \tag{7.22}$$

where α is defined in terms of γ and q from Section 7.2:

$$\alpha \triangleq \gamma\frac{1 - q}{q}$$

The rest of this section is divided into two subsections: Section 7.3.1 details the expression for the likelihood (7.21), where-after in Section 7.3.2 the final expression for the posterior (7.22) is derived.

7.3.1 The Likelihood

The expression for the likelihood $P(\mathbf{U}|\mathbf{J}, \mathbf{M}, \mathbf{X})$ of correspondence vectors $\mathbf{J}$ given the data $\mathbf{U}$ is very similar to the case without occlusion or clutter: we only need to add the likelihood of spurious measurements. On the assumption that spurious measurements have uniform probability of appearing in the image area A, the image likelihood $P(\mathbf{U}_i|\mathbf{j}_i, \mathbf{m}_i, \mathbf{X})$ can be split up in a spurious and non-spurious part:

$$P(\mathbf{U}_i|\mathbf{j}_i, \mathbf{m}_i, \mathbf{X}) = \left(\frac{1}{A}\right)^{S_i} \prod_{\mathbf{j}_{ik} \neq 0} P(\mathbf{u}_{ik}|\mathbf{j}_{ik}, \mathbf{m}_i, \mathbf{x}_{\mathbf{j}_{ik}}) \tag{7.23}$$

where S_i is defined as the number of spurious measurements in image i. Note that we can treat each image in isolation because conditional independence between images is assumed, given the correspondence vector $\mathbf{J}$.

It is convenient to reason in terms of imperfect bipartite matchings. As in Section 5.4 (page 70), we can view the correspondence problem in each image in terms of weighted matchings of the bipartite graph $G = (U, V, E)$, where the vertices $U = \{u_k | k \in 1..K_i\}$ correspond to the image measurements $\mathbf{u}_{ik}$, and the vertices $V = \{v_j | j \in 1..n\}$ are identified with the features $\mathbf{x}_j$. However, where before we only allowed perfect matchings or assignments, *we now also allow imperfect matchings* where (a) some vertices u_k can be unmatched, indicating that they are spurious, and (b) some vertices v_j can be unmatched, indicating that they are occluded in image i.

The associated bipartite graph is fully connected by the edges $E = U \times V$, and the edge weights are defined as before:

$$w(u_k, v_j) \triangleq -\log P(\mathbf{u}_{ik}|\mathbf{j}_{ik}, \mathbf{m}_i, \mathbf{x}_{\mathbf{j}_{ik}}) \tag{7.24}$$

Substituting this into equation (7.23), we obtain the following simple expression for the image likelihood:

$$P(\mathbf{U}_i|\mathbf{j}_i, \mathbf{m}_i, \mathbf{X}) = \left(\frac{1}{A}\right)^{S_i} e^{-w(\mathbf{j}_i)} \tag{7.25}$$

where the *weight* $w(\mathbf{j}_i)$ of an assignment is now defined as

$$w(\mathbf{j}_i) = \sum_{\mathbf{j}_{ik} \neq 0} w(u_k, \mathbf{j}_i(u_k))$$

7.3.2 The Posterior

Now that an expression for the prior is available, we can combine it with the likelihood (7.25) and derive an expression for the posterior probability $f(\mathbf{J})$ of correspondence vectors $\mathbf{J}$. We do this below for the simple combined visibility-detection model. For this model the posterior $f(\mathbf{J})$ factors over the images:

$$f(\mathbf{J}) = \prod_{i=1}^{m} P(\mathbf{j}_i|\mathbf{U}_i, \mathbf{m}_i, \mathbf{X})$$

where the individual posterior probabilities $P(\mathbf{j}_i|\mathbf{U}_i, \mathbf{m}_i, \mathbf{X})$ for the image correspondence vectors $\mathbf{j}_i$ are proportional to the product of the image likelihood and the correspondence prior:

$$P(\mathbf{j}_i|\mathbf{U}_i, \mathbf{m}_i, \mathbf{X}) \propto P(\mathbf{j}_i|\mathbf{m}_i, \mathbf{X}) P(\mathbf{U}_i|\mathbf{j}_i, \mathbf{m}_i, \mathbf{X}) \tag{7.26}$$

Substituting (7.25) for the likelihood and (7.20) for the prior into (7.26) and simplifying, we obtain the following expression for the posterior, where the image area A is eliminated:

$$\begin{aligned} P(\mathbf{j}_i|\mathbf{U}_i, \mathbf{m}_i, \mathbf{X}) &= \left[\frac{e^{-\lambda}}{(S_i + D_i)!}\lambda^{S_i} q^{D_i}(1-q)^{n-D_i}\right]\left[\left(\frac{1}{A}\right)^{S_i} e^{-w(\mathbf{j}_i)}\right] \\ &= \left(\frac{e^{-\lambda}}{(S_i + D_i)!}\right)\left(\frac{\lambda}{A}\right)^{S_i} q^{D_i}(1-q)^{n-D_i} \exp\left[-w(\mathbf{j}_i)\right] \\ &\propto \gamma^{S_i} q^{D_i}(1-q)^{n-D_i} \exp\left[-w(\mathbf{j}_i)\right] \end{aligned}$$

Recall that $\gamma = \lambda/A$ is the expected number of spurious measurements per unit area, and $w(\mathbf{j}_i)$ is the weight of the imperfect matching defined by the correspondence assignment $\mathbf{j}_i$:

$$w(\mathbf{j}_i) \triangleq \sum_{\mathbf{j}_{ik} \neq 0} w(u_k, \mathbf{j}_i(u_k)) = -\sum_{\mathbf{j}_{ik} \neq 0} \log P(\mathbf{u}_{ik}|\mathbf{j}_{ik}, \mathbf{m}_i, \mathbf{x}_{\mathbf{j}_{ik}})$$

Since $D_i = K_i - S_i$, and K_i is known when we evaluate the posterior, *and* if $q \neq 1$, we can further simplify this by isolating constant factors and dropping them from the equation:

$$\begin{aligned} P(\mathbf{j}_i|\mathbf{U}_i, \mathbf{m}_i, \mathbf{X}) &\propto \gamma^{S_i} q^{K_i - S_i}(1-q)^{n-K_i+S_i} \exp\left[-w(\mathbf{j}_i)\right] && (7.27) \\ &\propto \left(\gamma\frac{1-q}{q}\right)^{S_i} \exp\left[-w(\mathbf{j}_i)\right] && (7.28) \\ &\propto \alpha^{S_i} \exp\left[-w(\mathbf{j}_i)\right] && (7.29) \end{aligned}$$

where α is defined as

$$\alpha \triangleq \gamma\frac{1-q}{q}$$

The factor α increases with increasing occlusion and clutter. Thus, for a high value of α configurations with more spurious and occluded features are more probable.

Boundary Cases

In case all features are know to be visible, i.e. $q = 1$, then the simplification above does not work. However, we get an equally simple expression:

$$P(\mathbf{j}_i|\mathbf{U}_i, \mathbf{m}_i, \mathbf{X}) \propto \gamma^{S_i} \exp\left[-w(\mathbf{j}_i)\right]$$

i.e. this is of the same form as (7.3), but with $\alpha = \gamma$.

Clearly, if there is no occlusion *or* clutter, i.e. $q = 1$ and $\gamma = 0$, we recover the familiar Gibbs distribution from Chapter 5 (equation 5.8 on page 71):

$$P(\mathbf{j}_i|\mathbf{U}_i, \mathbf{m}_i, \mathbf{X}) \propto \exp\left[-w(\mathbf{j}_i)\right]$$

7.4 Sampling Imperfect Matchings

To approximate the E-step in the MCEM algorithm in the presence of occlusion and clutter, we need to sample over the imperfect matchings as defined above. This can be done in almost the same way as for perfect assignments (Section 5.5 on page 72). However, the proposal distribution, which involved simulating a "mini" Markov chain MC with transition probabilities defined by the weights, will be slightly modified to cope with free vertices and a special "spurious vertex", in which case an alternating cycle cannot be obtained.

7.4.1 Occluded Features

If features can be occluded, we need to allow free vertices v. We use the same proposal distribution, but now terminate the run of the Markov chain MC when a free vertex v is reached. In this case we have a simple path p, not a cycle. The path p is used in the same way as before to propose a new assignment $J' = J \oplus p$, i.e., we "flip" the assignments on the path of alternating edges.

For the simple chain flipping proposals, the acceptance ratio is gain equal to 1, as

$$\frac{f(J')}{f(J)} = \frac{e^{-w(J')}}{e^{-w(J)}} = \prod_{u \in p} \frac{q(u, J'(u))}{q(u, J(u))}$$

and

$$\frac{Q(J; J')}{Q(J'; J)} = \prod_{u \in p} \frac{q(u, J(u))}{q(u, J'(u))}$$

so

$$a_{CF} = \frac{f(J')}{f(J)} \frac{Q(J;J')}{Q(J';J)} = 1$$

Similarly as before, if we modify the MC transition probabilities to disallow matched edges, we get a modified acceptance ratio equal to:

$$a_{SMART} = \prod_{u \in p} \frac{1 - q(u, J(u))}{1 - q(u, J'(u))} \tag{7.30}$$

7.4.2 Spurious Measurements

To model spurious measurements we introduce a special null-vertex v_0 that can be matched with several u vertices.

Furthermore, we extend the edge weights w to $\bar{w}$ such that

$$\bar{w}(u,v) \triangleq \begin{cases} -\log \alpha & \text{if } v = v_0 \\ w(u,v) & \text{otherwise} \end{cases} \tag{7.31}$$

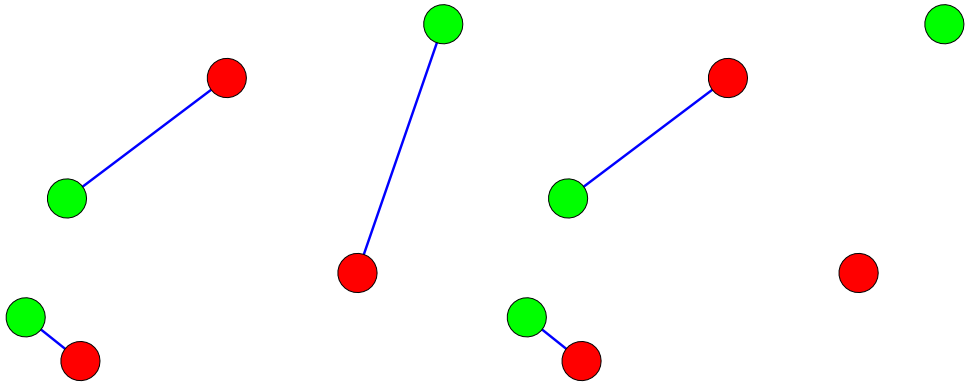
Intuitively, $\bar{w}(u, v_0) = -\log \alpha$ is the penalty for spurious measurements. It is highest (infinite) when $\alpha = 0$, and decreases with increasingly larger values of α. Then, from (7.29) we have

$$P(\mathbf{j}_i | \mathbf{U}_i, \mathbf{m}_i, \mathbf{X}) \propto \exp\left[-\sum_{k=1}^{K_i} \bar{w}(u_k, \mathbf{j}_i(u_k)) \right] \tag{7.32}$$

Note that the original weights now need to be defined more carefully, since they need to be balanced against $-\log \alpha$. For example, for a d-dimensional, isotropic Gaussian measurement error we have (in image i):

$$\begin{aligned} w(u_k, v_j) &= -\log\left[\frac{1}{(2\pi\sigma^2)^{d/2}} \exp\left(-\frac{1}{2\sigma^2} ||\mathbf{u}_{ik} - \mathbf{h}(\mathbf{m}_i, \mathbf{x}_j)||^2 \right) \right] & (7.33) \\ &= \frac{d}{2}\log(2\pi\sigma^2) + \frac{1}{2\sigma^2} ||\mathbf{u}_{ik} - \mathbf{h}(\mathbf{m}_i, \mathbf{x}_j)||^2 & (7.34) \end{aligned}$$

Again the proposal algorithm is the same, with the additional termination criterion when the special null-vertex is reached. The acceptance ratios are exactly as in the previous section, as the null-vertex can be regarded as a special vertex that is always considered “free”.



(a) True Matching (b) MAP Matching

Figure 7.1: Example matching. Red vertices represent predicted feature locations, whereas green vertices represent measurements.

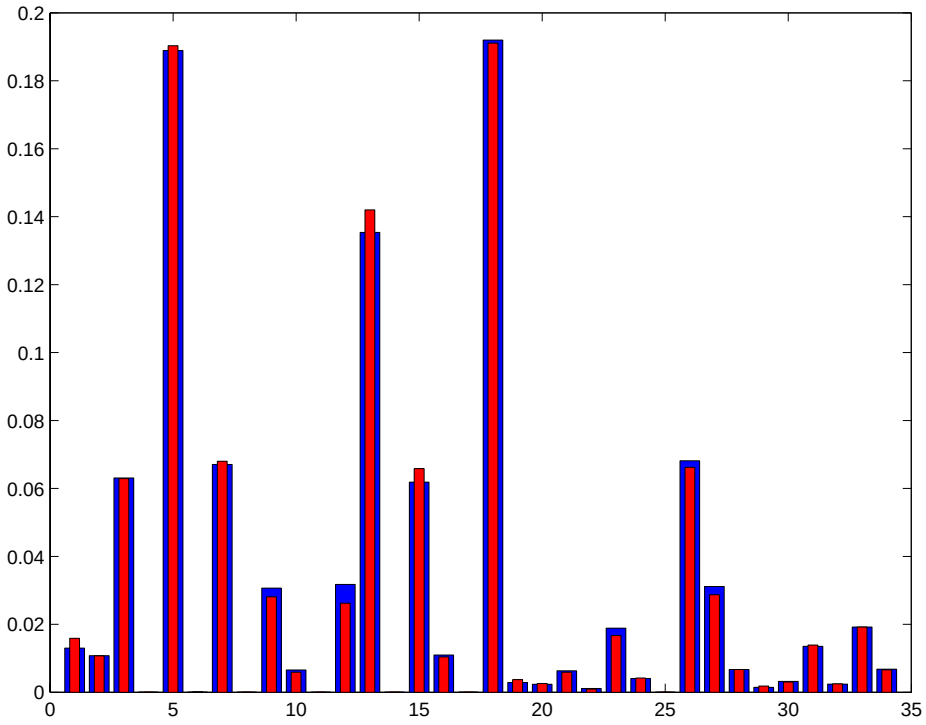


Figure 7.2: Result of sampling to approximate the true distribution (blue) by a sample histogram (red).

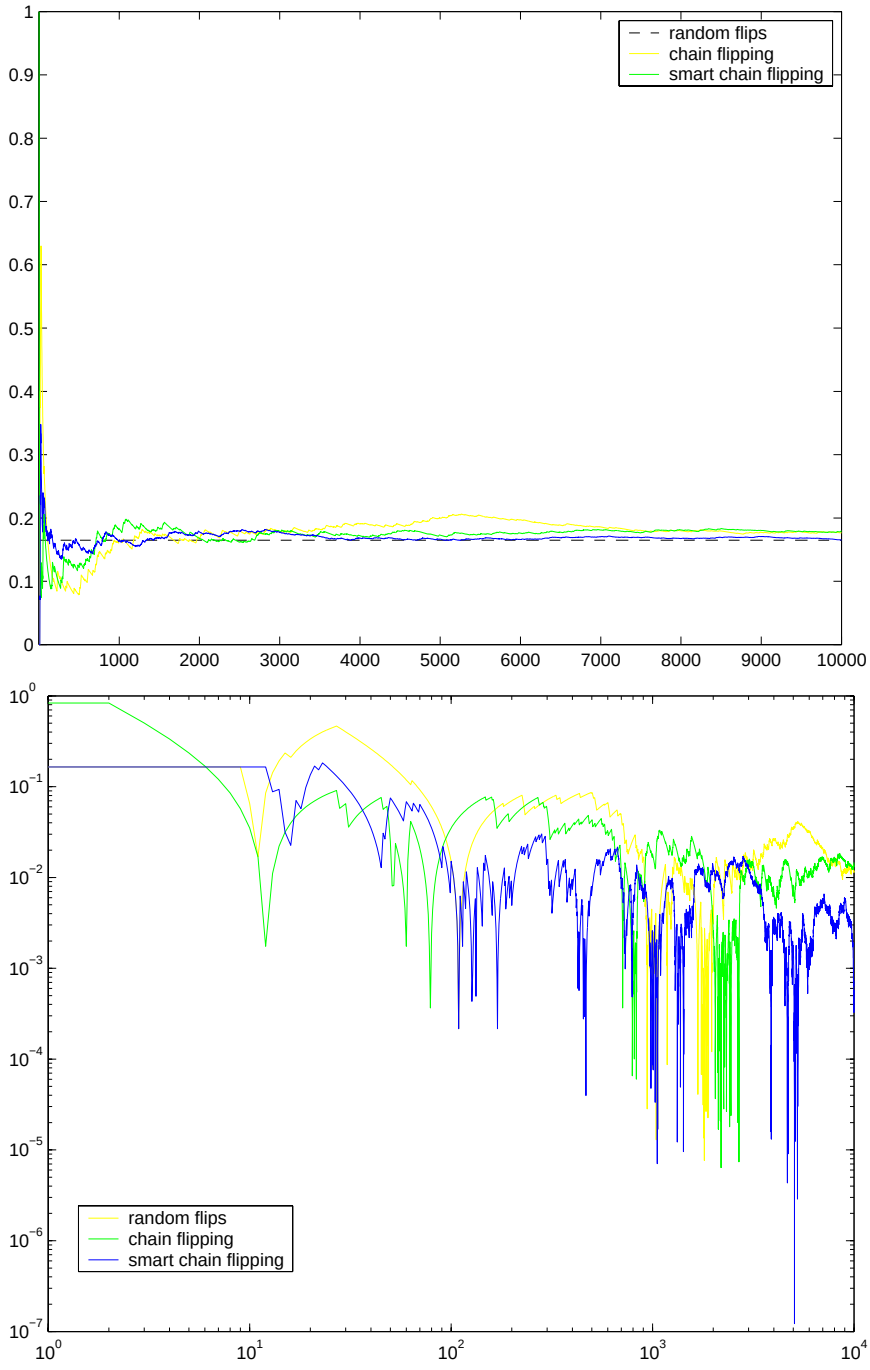


Figure 7.3: Time series of one marginal statistic Statistic and the corresponding log-log error plot for three different proposal strategies.

7.5 Results for Sampling with a Visibility Model

This sampling scheme can be tested experimentally and compared with the ground truth distribution for small values of n. In Figure 7.1 on page 141 an example of a matching is shown for $n = 3$ and $K = 3$. In this case, the true matching is actually an assignment, but the maximum a posteriori (MAP) matching declares one of the measurements to be spurious. After sampling, we can compare a histogram of the samples with the true distribution over all possible correspondences. This is done in Figure 7.2 on page 141, where the correspondences are arranged along the x-axis in arbitrary order. As you can see from this example, singling out one specific correspondence would skew our perception, as there are at least three different correspondences with roughly equal probability. Finally, the performance of the different proposal strategies (flipping, chain flipping and smart chain flipping) is compared in Figure 7.3 on the preceding page, in the same way as in Section 5.5.5 on page 79.

Chapter 8

Results with Occlusion and Clutter

In this chapter I present results for image sets with either clutter or occlusion. However, it was borne out by experimentation that, once clutter and occlusion are modeled, the wealth of new explanations that can be given to the data leads to many more local maxima.

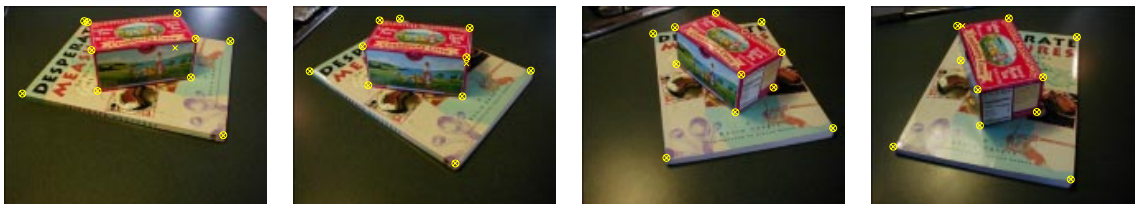
One approach to deal with the more challenging optimization problem resulting from the presence of occlusion and/or clutter is the use of prior knowledge. An advantage of formulating the geometric estimation problem with unknown correspondence as a MAP (maximum a posteriori) estimation problem is that incorporating prior knowledge can be done in a seamless manner. We only need to modify the M-step by adding an appropriate log-prior term to the objective function to be minimized.

Most of the results in this chapter have been obtained using a prior on the camera motion, which will be explained first in Section 8.1. Results are then presented for sequences with occlusion only (Section 8.2), clutter only (Section 8.3), and sequences with both occlusion and clutter (Section 8.4).

A second way of minimizing the impact of occlusion and clutter is by incorporating feature appearance, which will be discussed in Chapter 9.

8.1 The Arc Prior

In order to cope with the more challenging optimization problem in the presence of occlusion and clutter, most of the results presented in this chapter use a prior on the motion. In particular, this is done through an “arc prior”, which codifies the knowledge that (a) the



(a) image 1 (b) image 4 (c) image 7 (d) image 10

Figure 8.1: 4 (out of 10) images of two objects, taken in sequence and at regularly spaced intervals around the object.

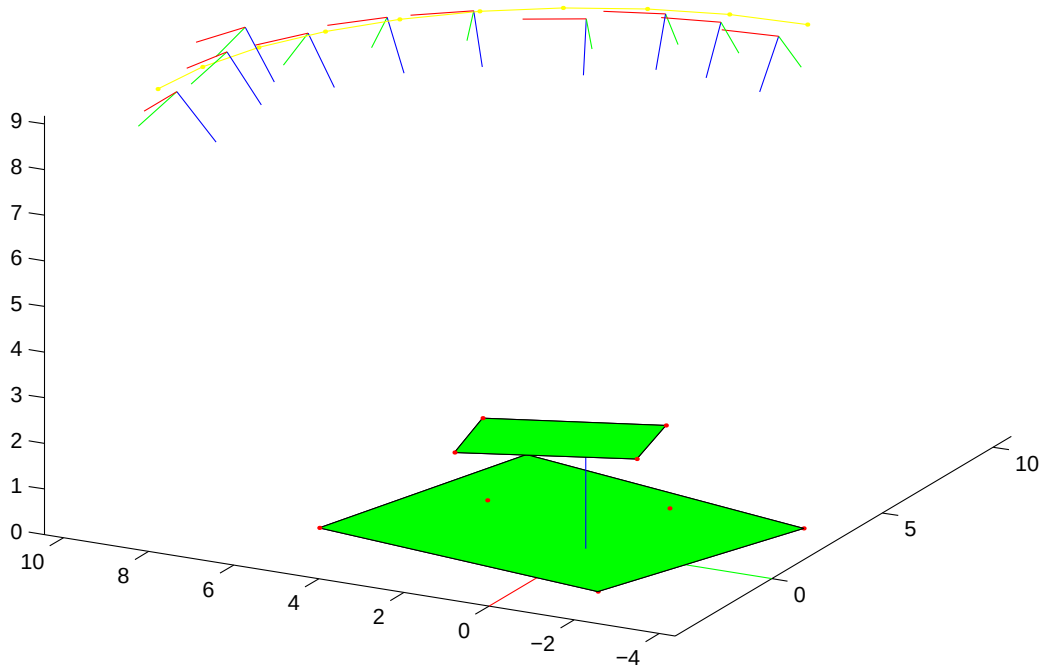


Figure 8.2: The “arc” prior: the idealized trajectory is shown in yellow, along with the MAP estimates for structure and motion for the sequence in Figure 8.1.

images were taken in sequence, and (b) the images were taken at regularly spaced intervals around the object, i.e. the camera was traveling roughly along a circular trajectory. This prior was inspired by a potential commercialization of the technology, which would enable consumers to digitize an object by taking a few snapshots of it. In order to simplify the problem, the snapshots would have to be taken at roughly equal angles and roughly equal distance, e.g. by placing the object on a table and walking around it, taking a snapshot at regular intervals. An example is shown in Figure 8.1.

The "arc" prior is parameterized by two parameters: a height and an arc-angle. Once these are given an ideal trajectory is calculated, and the prior states that the deviation of each camera from its ideal location is small, both in absolute position and in orientation. The ideal orientation is such that the camera faces the origin exactly. The radius of the ideal circular trajectory is fixed, which also fixes the otherwise arbitrary scale of the reconstruction. All this is illustrated in Figure 8.2, which shows the idealized trajectory in yellow, and the MAP estimates for structure and motion. As you can see, the camera frames stay close to their ideal positions and orientations. Note that in taking this image sequence, no emphasis was placed on trying to follow a circular trajectory exactly: the prior only provides a rough sketch.

8.2 Examples with Occlusion

8.2.1 Book

The sequence from Figure 8.1 (see also Figure 8.3) was used to demonstrate the MCEM approach in the presence of occlusion. Again, measurements were extracted by hand. However, some features are now occluded in some of the images. There were no spurious measurements (i.e. no clutter). The actual measurements are shown in Figure 8.4.

The MCEM algorithm was run for 25 iterations, and the marginals (or soft correspondences) are shown for a subset of the iterations in Figure 8.5. The marginals are presented in such a way that the ground truth correspondence yields identity matrices as before. However, in the case of occlusion some rows will be missing, as some features are occluded in some of the images.

Figure 8.5e shows that the ground truth correspondence is recovered. An input image where one of the features was occluded is shown in Figure 8.3.

Figure 8.3: Image 1 of the "book" sequence with the measurements and the MAP estimate superimposed. Note that the position of the occluded corner of the box is predicted but a measurement is not available due to occlusion.

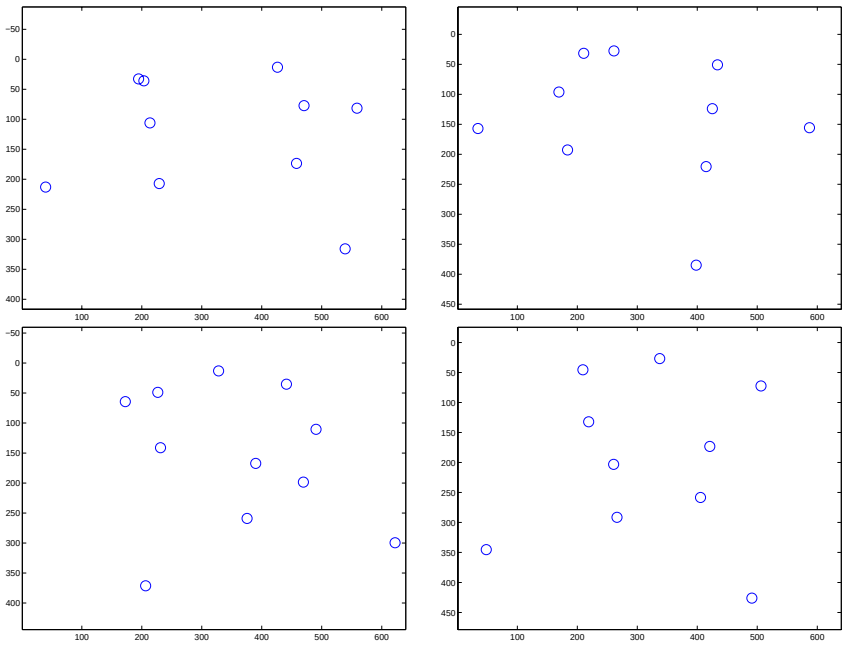


Figure 8.4: Measurements in the 4 (out of 10) input images.

(a) it 1 (b) it 7 (c) it 13 (d) it 19 (e) it 25

Figure 8.5: Marginal probabilities computed in the E-step. Occlusion shows up as breaks in the "perfect" correspondence matrix (an identity matrix). In this example, the first four images are missing a measurement on the last feature, whereas the last image does not have an observation on feature 2.

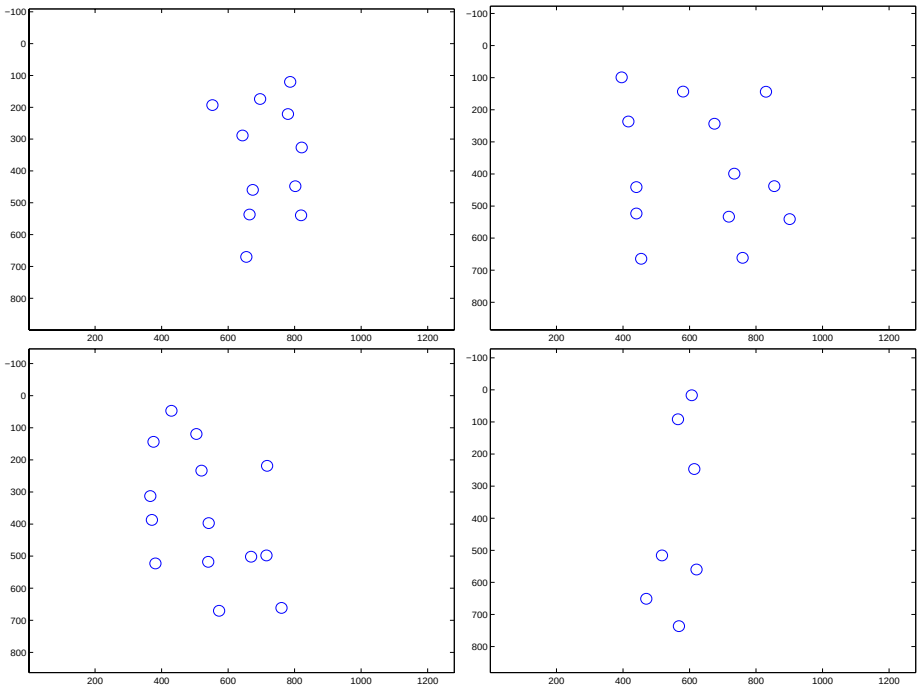


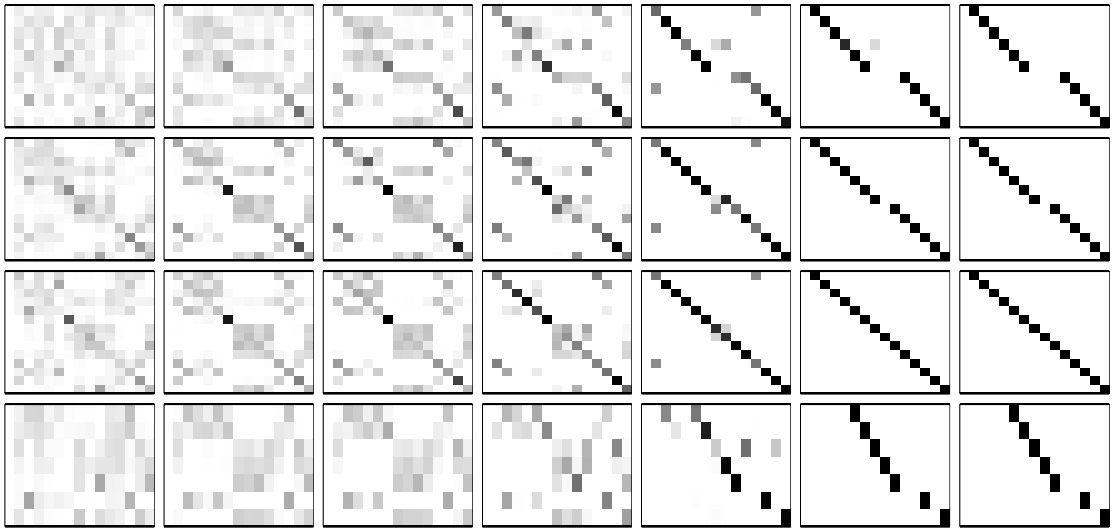
Figure 8.6: Measurements in the 4 “Canon” input images.

8.2.2 Canon

The measurements for a second sequence, also taken under the “arc-prior” assumption, are shown in Figure 8.6. Please note the difficulty of determining the 3D structure of the object based on these measurements alone.

The MCEM approach, however, manages quite nicely, in no small part because of the strong motion prior. The evolution of the marginals probabilities over the course of 25 iterations is shown in Figure 8.7. Note that there is considerably more occlusion than in the “book” sequence from Section 8.2.1. Especially in the last image, almost half of the features visible in the first image are occluded.

Finally, trajectories of the projected structure over time and the original input images are shown in Figures 8.8 and 8.9, respectively.



(a) it 1 (b) it 5 (c) it 9 (d) it 13 (e) it 17 (f) it 21 (g) it 25

Figure 8.7: Marginal probabilities computed in the E-step (“Canon”).

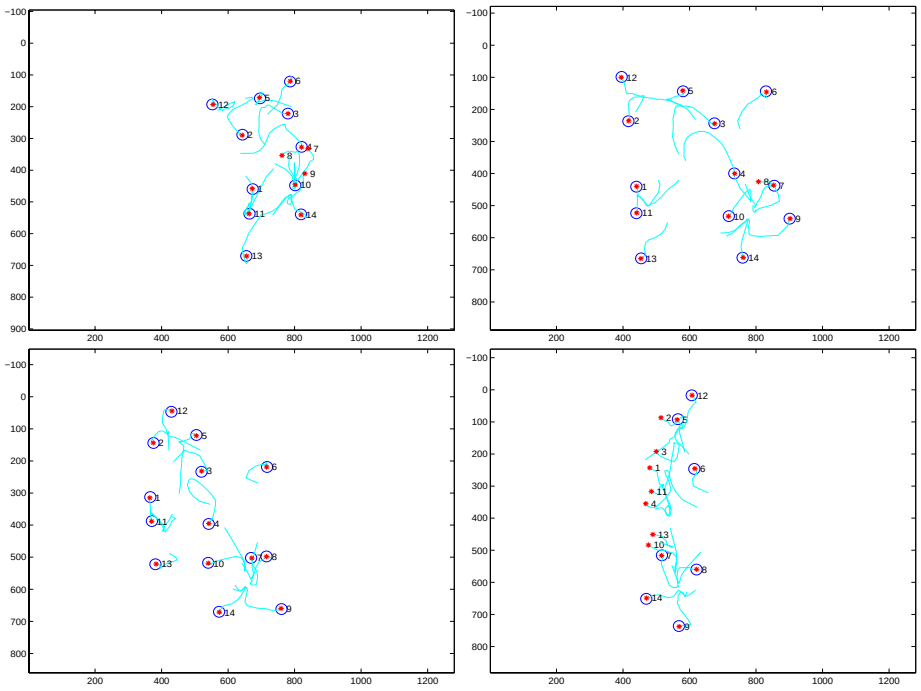


Figure 8.8: Plot of the predicted location for each of the features over time. The last predicted location is marked with an asterisk. Measurements are shown as circles.

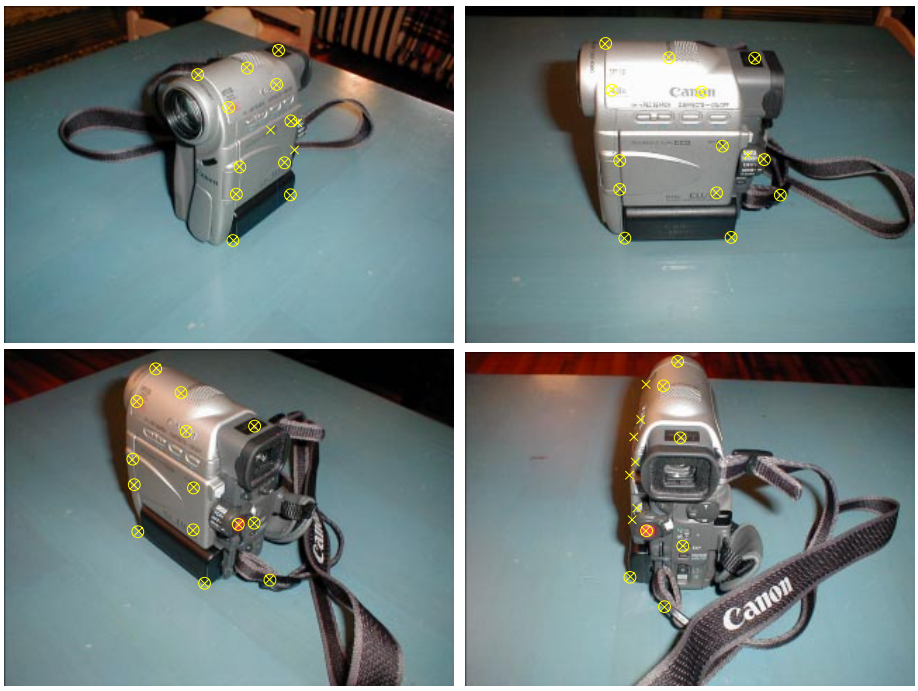


Figure 8.9: Input images for the "Canon" sequence. The last predicted location is marked with an asterisk. Measurements are shown as circles.

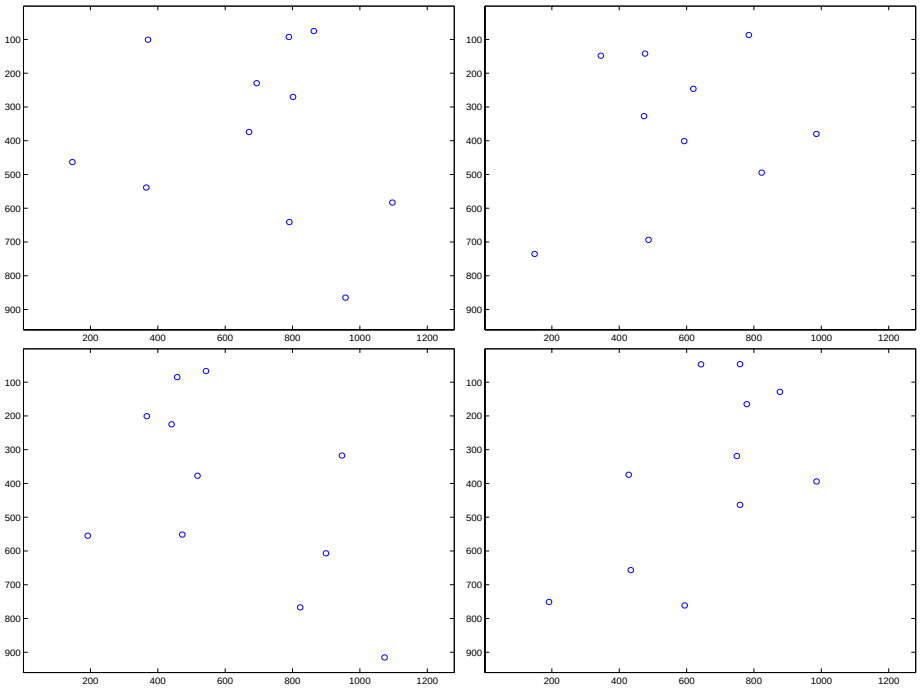
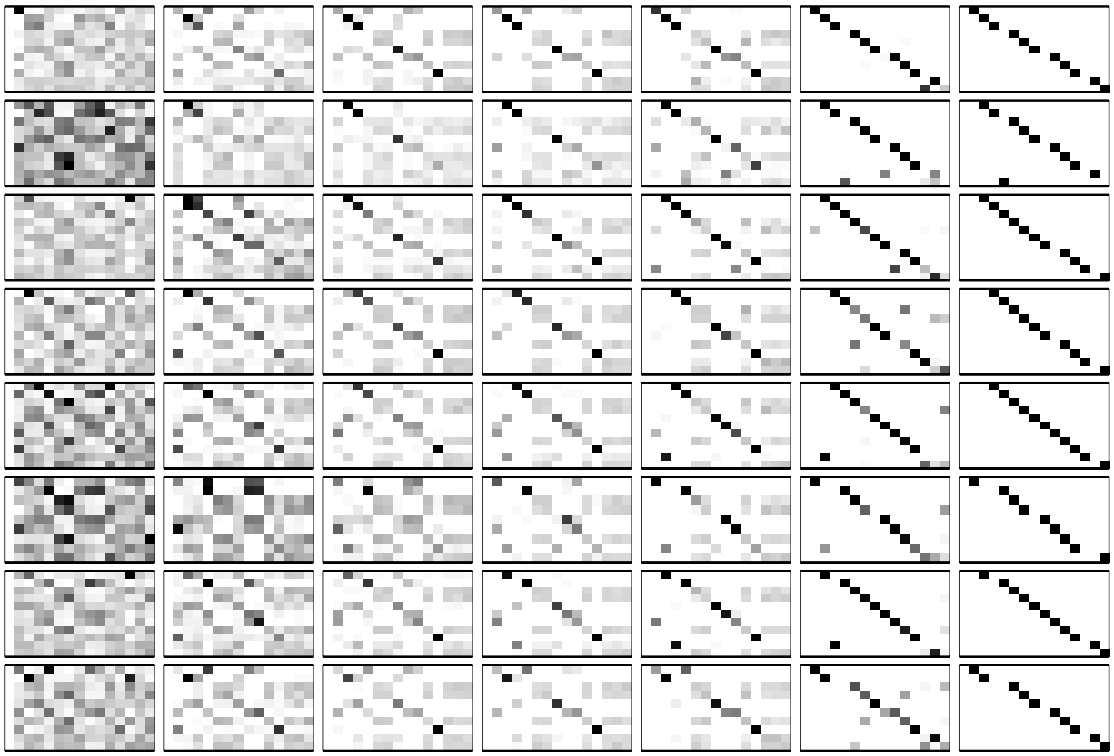


Figure 8.10: Measurements in 4 (out of 8) “horse” input images.

8.2.3 Horse

Measurements for a last sequence with occlusion only are shown in Figure 8.10. This sequence has a lot of occlusion as the 8 images were taken from all around the object. Hence, the prior for the arc-angle was set to 45 degrees. Note that this is only a prior and an initial estimate: the angle is also optimized for in the M-step.

Again EM was run for 25 iterations, and the by now familiar marginal probability plots and predicted structure trajectories are shown in Figures 8.11 and 8.12, respectively. The original input images are shown in Figure 8.13.



(a) it 1 (b) it 5 (c) it 9 (d) it 13 (e) it 17 (f) it 21 (g) it 25

Figure 8.11: Marginal probabilities computed in the E-step. Up to 5 (out of 11) features are occluded in each image.

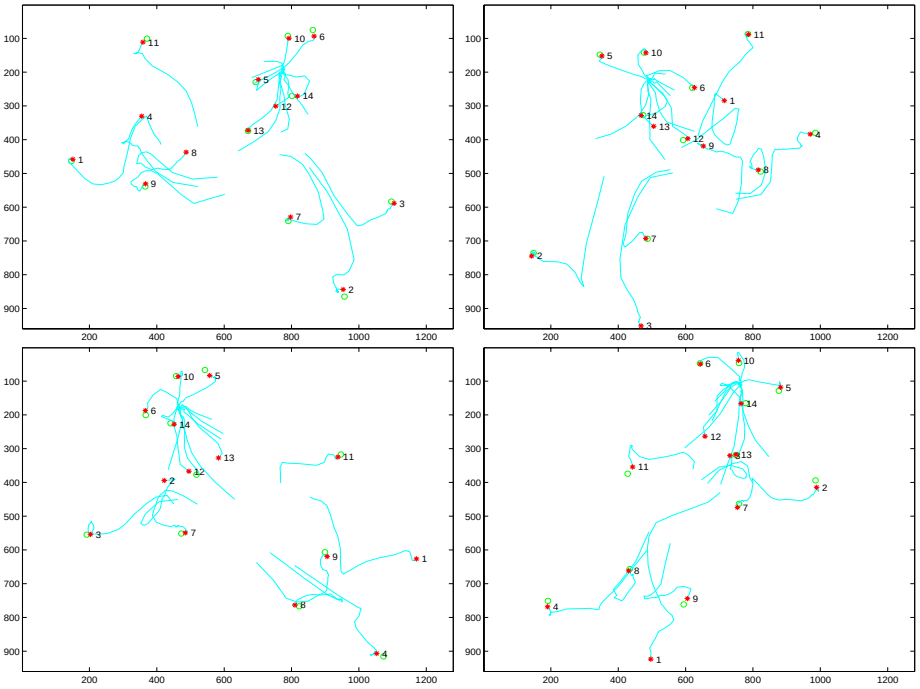


Figure 8.12: Plot of the predicted location for each of the features over time. The last predicted location is marked with an asterisk. Measurements are shown as circles.

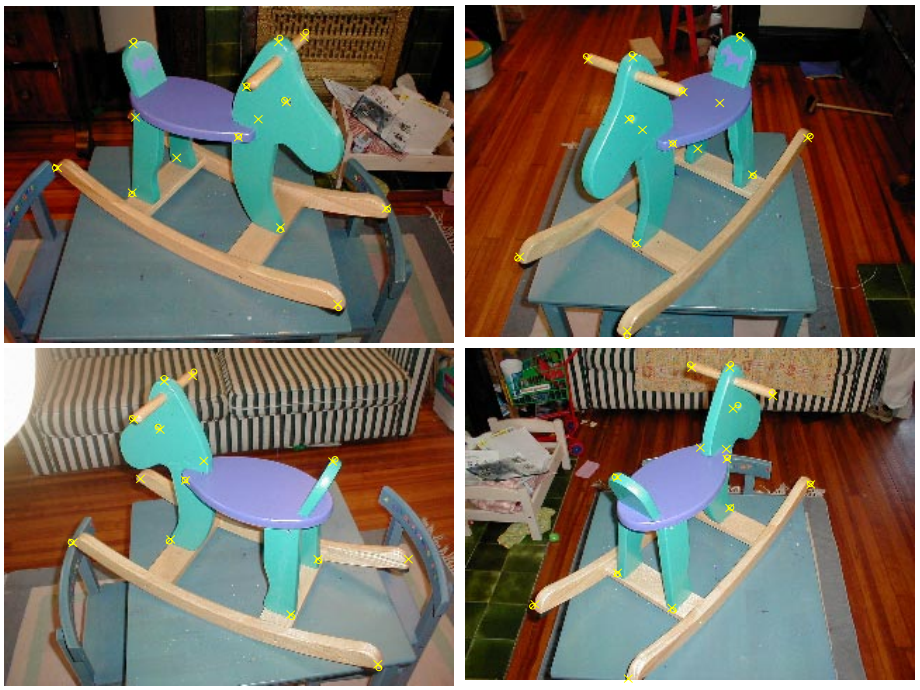
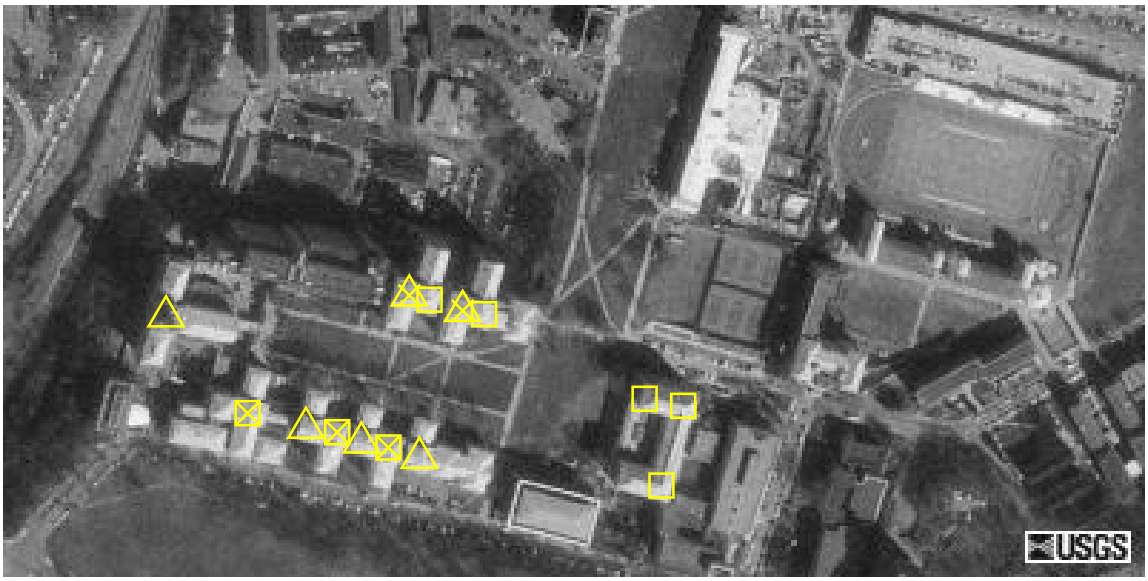


Figure 8.13: Input images for the "horse" sequence. The last predicted location is marked with an asterisk. Measurements are shown as circles.



(a) Measurements

Figure 8.14: Translational pose estimation example from the introduction, with spurious measurements. Measurements corresponding to model features are marked with a cross.

8.3 Examples with Clutter

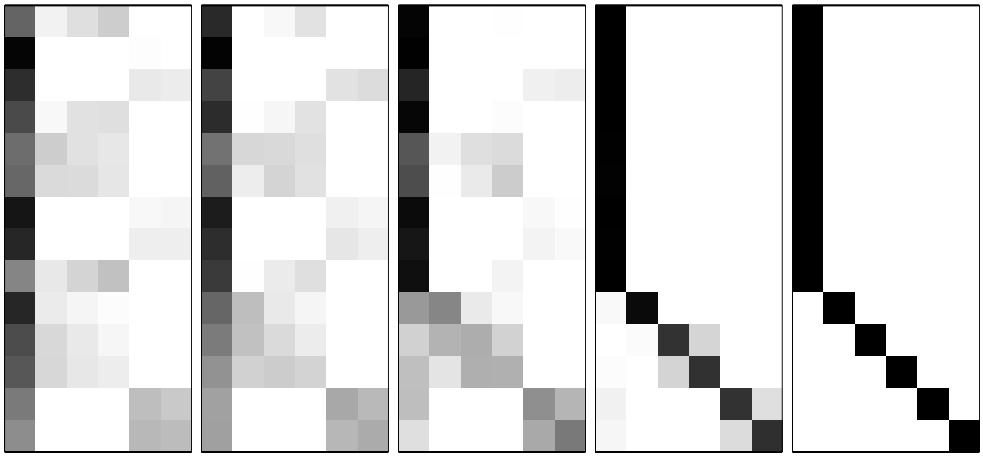
The MCEM approach can not only be used for structure and motion problems, but also for simpler geometric estimation problems, such as pose estimation. An instance of a pose estimation problem in the presence of clutter, but no occlusion, is shown in Figure 8.14 and 8.15. It is the same example as was used in the introduction: the top panel shows an idealized model of the CMU quad, to be located in the aerial image at the bottom. A "corner building" detector was simulated to generate the measurements.

Note that there are two different types of measurements: squares represent a "right-handed" corner, and triangles represent "left-handed" corners. The use of symbolic appearance attributes such as these will be discussed in detail in Chapter 9, but the bottom-line is that the sampler will only allow correspondences that consistently assign measurements of a given type to model features of the same type.

In total, there were 9 spurious measurements (i.e. clutter measurements) versus only 5 actual measurements. The MCEM approach has no trouble recovering the pose, however, and does so in only 5 iterations. The marginal probabilities over the course of the EM

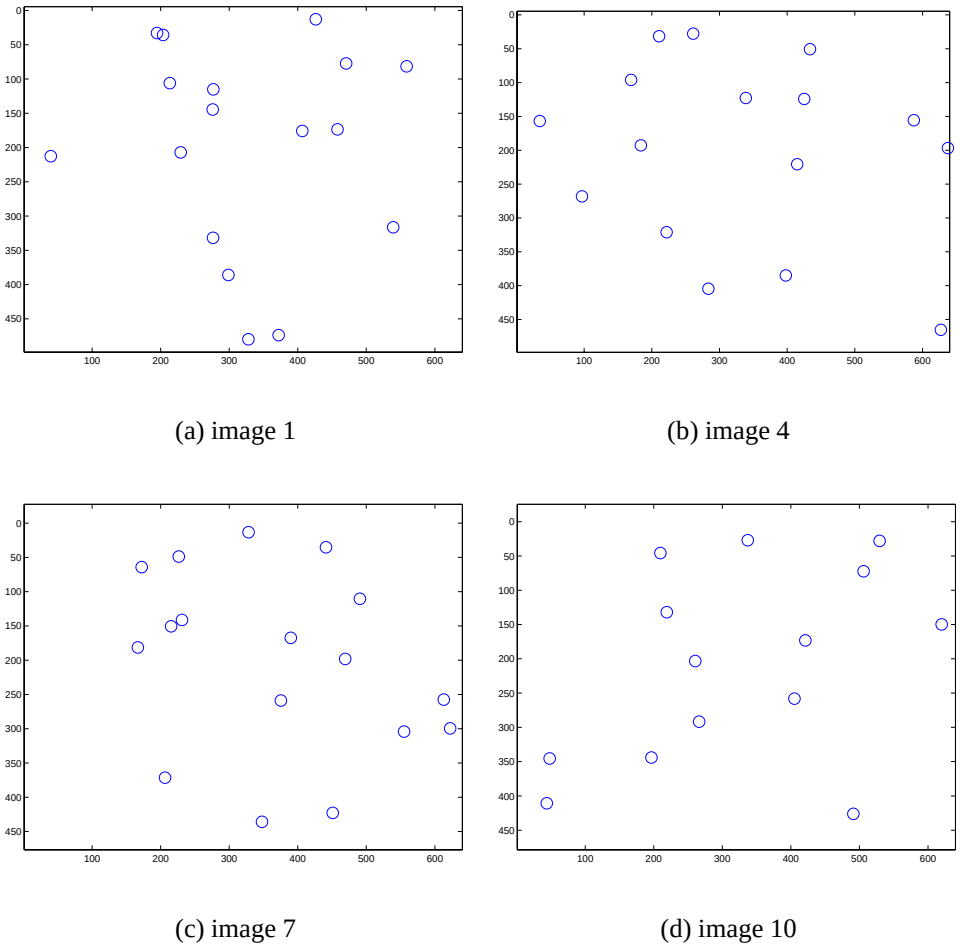
(a) Model

Figure 8.15: The model of the CMU quad whose location is to be estimated in the image of Figure 8.14.



(a) it 1 (b) it 2 (c) it 3 (d) it 4 (e) it 5

Figure 8.16: Marginal probabilities computed in the E-step.



(a) image 1

(b) image 4

(c) image 7

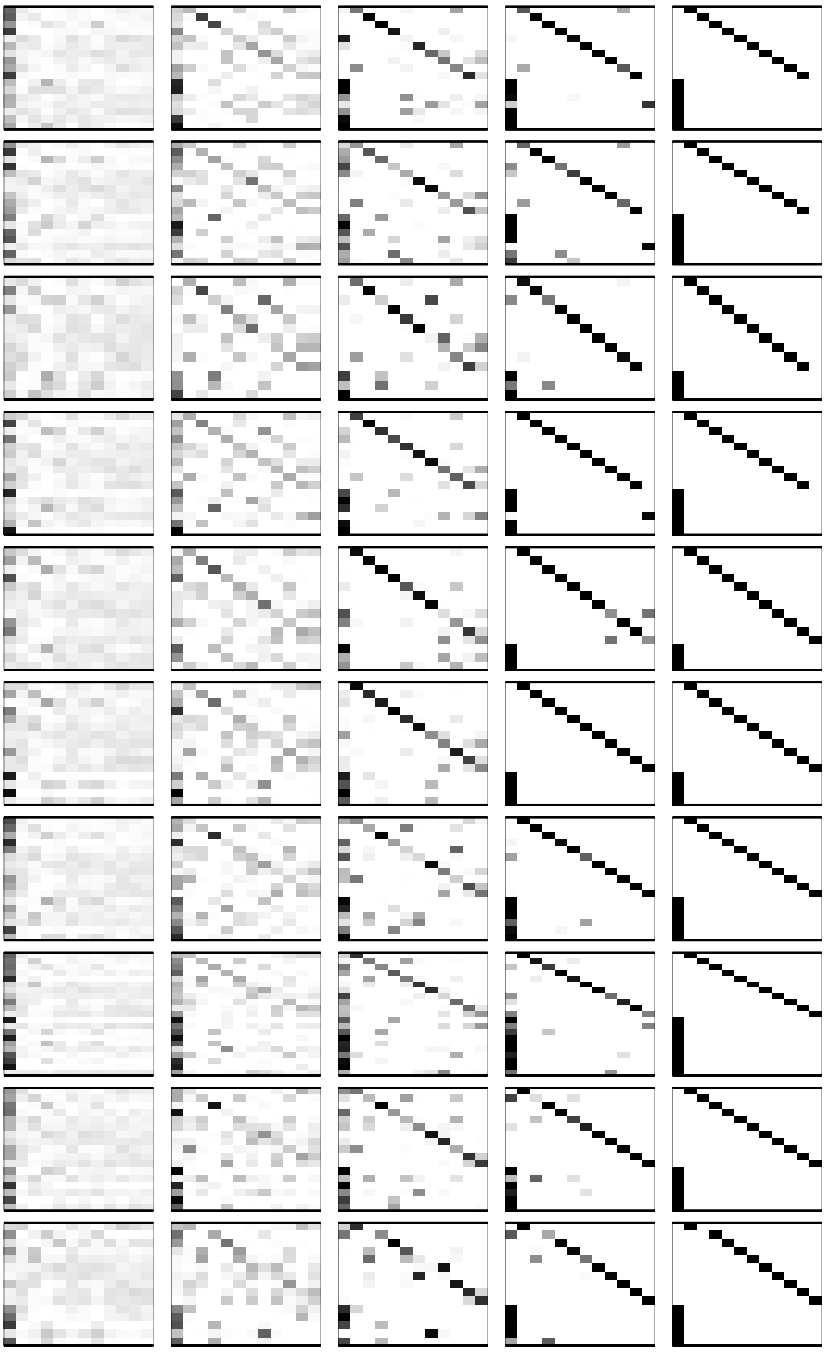
(d) image 10

Figure 8.17: Measurements in the 4 (out of 10) input images.

iterations are shown in Figure 8.16. In this figure, the first column in each matrix is reserved to indicate the probability that a measurement is spurious. The marginals are re-arranged such that the spurious measurements are the first 9 measurements, and the algorithm can be seen to converge to the ground truth assignment.

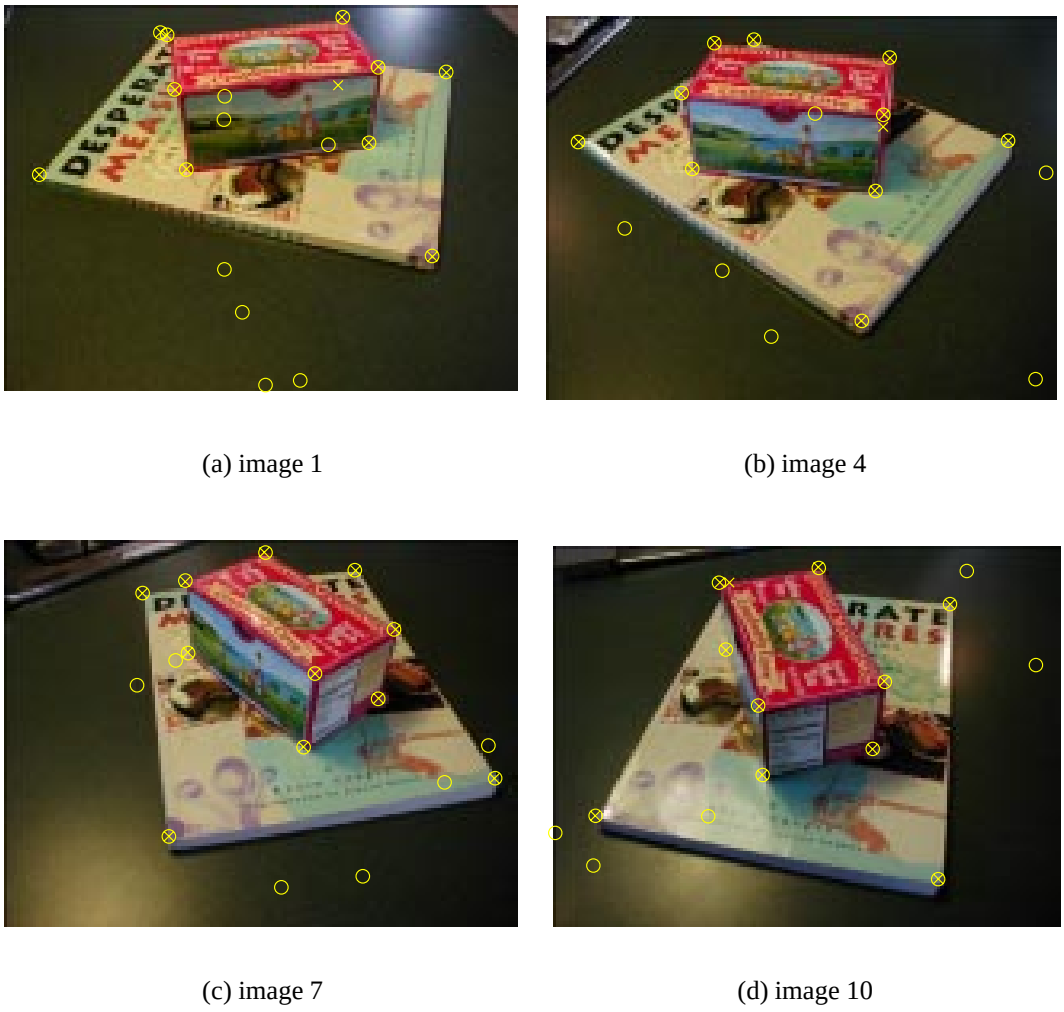
8.4 A SFM Example with Occlusion *and* Clutter

Finally, an example of a structure from motion problem in the presence of both occlusion and significant clutter is shown in Figure 8.17. In this case, the number of features n was equal to 11. Measurements were extracted from the images (shown later) by hand, but some features were occluded in some of the images. To simulate clutter, spurious



(a) it 1 (b) it 7 (c) it 13 (d) it 19 (e) it 25

Figure 8.18: Marginal probabilities computed in the E-step. Note that in this sequence there is relatively little occlusion.



(a) image 1

(b) image 4

(c) image 7

(d) image 10

Figure 8.19: Input images with projected structure estimate and measurements. The last predicted location is marked with an asterisk. Measurements are shown as circles. Note the significant amount of clutter measurements.

measurements were generated randomly, with a uniform probability over the image. The number of spurious features S_i for each image was drawn from a Poisson distribution with mean 5. In the problem instance shown, the values drawn for each of the 10 images were: 7, 7, 3, 6, 3, 4, 6, 10, 6, and 5, respectively. In other words, in some cases there were as many spurious as non-spurious measurements. Given that no appearance information about the measurements is used at all, recovering structure from this data is not straightforward.

Again an "arc" prior was used, with a prior of 10 degrees on the arc-angle. The EM algorithm was run for 25 iterations, and the evolution of the marginal probabilities is shown in Figure 8.18. Note that in this case the marginals were arranged in such a way that the spurious measurements are ordered last in each image. From panel (a), corresponding to the first iteration, we see that the initial distribution over correspondences is rather removed from the actual ground truth correspondence: almost all measurements are mostly estimated as spurious. However, the picture gradually improves, and the ground truth is finally recovered by the last iteration. The original input images, with spurious measurements, are shown in Figure 8.19.

8.5 Discussion

While all of the problem instances shown above converged, the structure recovery is considerably more challenging in the presence of occlusion and clutter. Especially if the amount of clutter is increased, the MCEM algorithm needs to be restarted multiple times or fails to converge at all. In addition, the approach often fails in part or completely without using a motion prior.

This is not all too surprising, given that *no appearance information is used at all*. The following chapter will discuss how appearance information can easily be incorporated within the MCEM framework, and how it alleviates the convergence problem in the presence of clutter and/or occlusion.

Chapter 9

Incorporating Appearance

In this chapter I discuss how appearance information can be incorporated into the geometric estimation process. Since in the presence of occlusion and clutter the number of possible correspondence matchings grows dramatically, the number of local maxima and the cost of sampling over the space of matchings both increase. Adding appearance information can help constrain the sampling over correspondences, and hence make the entire problem more tractable. However, reliable models of appearance measurements are hard to come by, since the appearance of 3D features can change significantly if images are taken from widely separated viewpoints. Even though recent work on appearance-based matching has produced impressive results for a restricted set of image transformations (Schmid and Mohr, 1997; Lowe, 1999; Mikolajczyk and Schmid, 2001), no simple appearance models are available that are invariant under 3D transformations. Hence, we are obligated to either adopt a complicated model of appearance, e.g. oriented surface patches, or accept a more limited range of viewpoints that can be handled.

9.1 An Appearance Measurement Model

Before we can incorporate appearance information, we need to model the process of how *appearance measurements* $\mathbf{A}$ are generated given that the structure is described by the *appearance parameters* $\mathbf{Y}$. Whereas more sophisticated models are possible, below I will assume a simple model wherein appearance is measured independently for each feature.

9.1.1 Appearance Measurements and Parameters

Let us assume that the appearance measurements $\mathbf{A}$ are in the form of a collection of individual *appearance measurements* $\mathbf{a}_{ik}$, one for each associated location measurement $\mathbf{u}_{ik}$, i.e. $\mathbf{A} = \{\{\mathbf{a}_{ik}|k \in 1..K_i\}|i \in 1..m\}$, where the $\mathbf{a}_{ik}$ can be either continuous, discrete, or a mix of both. Below I often refer to appearance measurements using a single index, i.e. $\mathbf{A} = \{\mathbf{a}_k|k \in 1..K\}$, where we define $K \stackrel{\Delta}{=} \sum_i K_i$ as the total number of measurements.

In order to model the appearance measurement process, I introduce hidden appearance parameters $\mathbf{Y}$. In particular, let us introduce for every feature $\mathbf{x}_j$ an appearance variable $\mathbf{y}_j$, which comprises of parameters that describe the appearance of the feature. The appearance parameters for the entire structure are denoted by $\mathbf{Y} \stackrel{\Delta}{=} \{\mathbf{y}_j|j \in 1..n\}$.

A number of useful appearance representations $\mathbf{Y}$ come to mind. For example, if the feature is seen from roughly the same orientation and distance in each image in which it is visible, the appearance parameters $\mathbf{y}_j$ can be a collection of pixels, predicting the pixel values $\mathbf{a}_{ik}$ in a small window around the projected feature $\mathbf{h}(\mathbf{m}_i, \mathbf{x}_{\mathbf{j}_{ik}})$. Optionally, we can incorporate surface orientation, in which case the predicted pixel values would be obtained by first appropriately transforming the surface patch model in the image. Another, less involved approach is to predict grayscale or color invariants that can be measured in the image. Finally, the appearance model can be symbolic, e.g. stating that the feature is "corner-like", or a "T-junction", or any other discrete attribute that can be reliably extracted from the images.

An example of the latter, a symbolic appearance model, is shown in Figure 9.1. Here the appearance is modeled by a binary random variable denoting either square or triangular features, i.e. $\mathbf{y}_j \in \{S, T\}$, for $j \in \{1, 2\}$. The appearance measurements are also binary, with $\mathbf{a}_k \in \{s, t\}$, for $k \in \{1, 2, 3, 4\}$. The lower-case notation makes it explicit that s and t are *measurement* values.

9.1.2 The Appearance Likelihood Model

What is needed is a probabilistic description of the appearance measurement process. In general, this process can be completely described by a conditional probability density function $P(\mathbf{A}|\mathbf{J}, \mathbf{\Theta}, \mathbf{Y}) = P(\mathbf{a}_1, \ldots, \mathbf{a}_K|\mathbf{J}, \mathbf{\Theta}, \mathbf{y}_1, \ldots, \mathbf{y}_n)$. In order to simplify this description, I make the modeling assumption that, *given* the geometry $\mathbf{\Theta}^t$ and the structure appearance parameters $\mathbf{Y} = \{\mathbf{y}_j|j \in 1..n\}$, the measured appearance values $\mathbf{a}_{ik}$ are conditionally

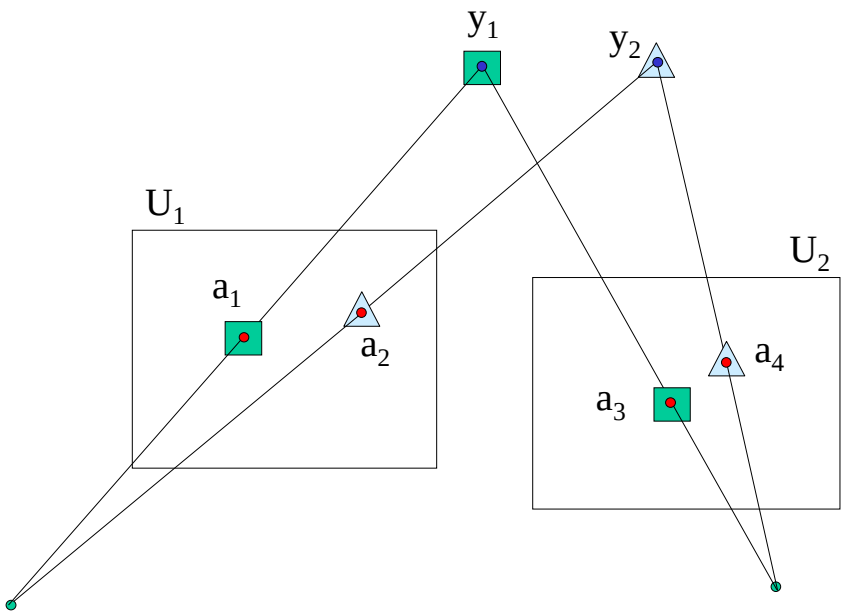


Figure 9.1: Example of appearance parameters $\mathbf{y}_j$ and measurements $\mathbf{a}_k$.

$P(s\|S)$	0.9
$P(s\|T)$	0.2
$P(t\|S)$	0.1
$P(t\|T)$	0.8

Table 9.1: Example of a measurement model for a binary appearance model.

independent of each other, i.e.

$$P(\mathbf{A}|\mathbf{J}, \Theta^t, \mathbf{Y}) = \prod_{i=1}^{m} \prod_{k=1}^{K_i} P(\mathbf{a}_{ik}|\mathbf{j}_{ik}, \mathbf{m}_i^t, \mathbf{X}^t, \mathbf{Y}) \tag{9.1}$$

As an illustration, Table 9.1 provides an example measurement model for the binary appearance example from Figure 9.1. Because of the conditional independence assumption, we only need to provide four numbers to specify the entire joint appearance measurement model. In this case, squares are more reliably measured than triangles. Note that, as required, $P(s|S) + P(t|S) = 1$, and $P(s|T) + P(t|T) = 1$.

9.1.3 Spurious Measurements

We can simplify expression 9.1 by being explicit about which measurements correspond to actual features and which do not, i.e. are *spurious*. If we introduce a special model $P_0(\mathbf{a}_{ik}) = P(\mathbf{a}_{ik}|\mathbf{j}_{ik} = 0)$ to describe the appearance of spurious features, we can decompose the expression above into a spurious and non-spurious part:

$$P(\mathbf{A}|\mathbf{J}, \mathbf{\Theta}^t, \mathbf{Y}) = \left(\prod_{i=1}^{m} \prod_{\mathbf{j}_{ik}=0} P_0(\mathbf{a}_{ik}) \right) \left(\prod_{i=1}^{m} \prod_{\mathbf{j}_{ik}\neq 0} P(\mathbf{a}_{ik}|\mathbf{m}_i^t, \mathbf{x}_{\mathbf{j}_{ik}}^t, \mathbf{y}_{\mathbf{j}_{ik}}) \right) \tag{9.2}$$

Because of this we can now use, in the non-spurious part above, the feature location $\mathbf{x}_{\mathbf{j}_{ik}}^t$ and appearance $\mathbf{y}_{\mathbf{j}_{ik}}$ that correspond to $\mathbf{a}_{ik}$ according to $\mathbf{J}$. For notational simplicity, we define the *spurious appearance likelihood*

$$L_0(S_0) \stackrel{\Delta}{=} P(S_0|\mathbf{J}) = \prod_{\mathbf{a}_k \in S_0} P_0(\mathbf{a}_k) \tag{9.3}$$

where $S_0 = \{\mathbf{a}_k|\mathbf{j}_k = 0\}$ is defined to as the set of spurious measurements. Note that $L_0(\emptyset) = 1$. Using this definition we obtain:

$$P(\mathbf{A}|\mathbf{J}, \mathbf{\Theta}^t, \mathbf{Y}) = L_0(S_0) \prod_{i=1}^{m} \prod_{\mathbf{j}_{ik}\neq 0} P(\mathbf{a}_{ik}|\mathbf{m}_i^t, \mathbf{x}_{\mathbf{j}_{ik}}^t, \mathbf{y}_{\mathbf{j}_{ik}}) \tag{9.4}$$

9.1.4 Partitioning the Measurements into Sets

The correspondence $\mathbf{J}$ induces a set partition on the measurements, which allows us to rewrite expression (9.4) in an insightful way. Indeed, it can be re-arranged as a product of n factors, each one concerned with the appearance of a given feature $\mathbf{x}_j$. To see this, note that, given a specific correspondence vector $\mathbf{J}$, every appearance measurement $\mathbf{a}_{ik}$ is paired with one and only one feature $\mathbf{x}_{\mathbf{j}_{ik}}$ and its corresponding appearance parameters $\mathbf{y}_{\mathbf{j}_{ik}}$. In other words, the correspondence vector $\mathbf{J}$ induces a *set partition* on the measurements $\mathbf{J}$. Define S_j to be the set of measurements that correspond to feature $\mathbf{x}_j$, with $j \in 1..n$. Then we can re-arrange the product over all measurements as n products over the sets S_j:

$$P(\mathbf{A}|\mathbf{J}, \mathbf{\Theta}^t, \mathbf{Y}) = L_0(S_0) \prod_{j=1}^{n} \prod_{\mathbf{a}_k \in S_j} P(\mathbf{a}_k|\mathbf{M}^t, \mathbf{x}_j^t, \mathbf{y}_j) \tag{9.5}$$

Note that the correspondence vector $\mathbf{J}$ disappeared from the equation: it is subsumed by the partitioning of the measurements $\mathbf{U}$ in sets S_j. Also, we need to condition on all motion parameters $\mathbf{M}^t$, as the sets S_j can measurements in several images.

We can illustrate the partitioning over sets S_j with the example of Figure 9.1. Let us represent correspondence vectors by a string of numbers enclosed in square brackets. For example, in the figure, the correct correspondence $\mathbf{J} = [1212]$ is shown. This assignment induces two sets: $S_1 = \{\mathbf{a}_1, \mathbf{a}_3\}$, and $S_2 = \{\mathbf{a}_2, \mathbf{a}_4\}$. In this case, the appearance likelihood of $\mathbf{J}$ given $\mathbf{Y}$ is

$$\begin{aligned} P(\mathbf{A}|\mathbf{J} = [1212], \mathbf{\Theta}^t, \mathbf{Y}) &= P(\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3, \mathbf{a}_4|\mathbf{J} = [1212], \mathbf{y}_1 = S, \mathbf{y}_2 = T) \\ &= \prod_{j=1}^{2} \prod_{\mathbf{a}_k \in S_j} P(\mathbf{a}_k|\mathbf{y}_j) \\ &= [P(\mathbf{a}_1|\mathbf{Y}_1)P(\mathbf{a}_3|\mathbf{Y}_1)] \times [P(\mathbf{a}_2|\mathbf{Y}_2)P(\mathbf{a}_4|\mathbf{Y}_2)] \\ &= [P(s|S)P(s|S)] \times [P(t|T)P(t|T)] \end{aligned}$$

Using the values from Table 9.1, we have

$$P(\mathbf{A}|\mathbf{J} = [1212], \mathbf{y}_1 = S, \mathbf{y}_2 = T) = 0.9^2 0.8^2 = 0.5184$$

9.2 Some Simple Appearance Models

This section discusses some simple appearance models that I have used in order to demonstrate the use of the MCEM approach for structure from motion.

9.2.1 Sophisticated Models

Before considering simpler models, it is of interest to note that quite sophisticated models can be used. In particular, we could use *oriented surface patches* to model a patch of texture around each feature point, which is then appropriately transformed into the images using texture mapping. Oriented surface particles have been used before for geometry modeling (Szeliski and Tonnesen, 1992), and stereo (Fua, 1997). The appearance parameters $\mathbf{y}_j$ would then correspond to the texture on the patch, and can be estimated in parallel with the geometry, as explored in (Dellaert et al., 1998a; Dellaert et al., 1998b). Another approach would be to use a deformable mesh, with textured polygons to model the appearance.

9.2.2 A Simple Discrete Measurement Model

The example of Figure 9.1 used discrete measurements, but is still quite general in that there were no restrictions on the conditional probability table specifying the model. In contrast,

I will refer the *simple discrete measurement model* when the following assumptions are satisfied:

1. The appearance parameters $\mathbf{y}_j$ and measurements $\mathbf{a}_k$ are discrete (symbolic) and defined over the same set of labels $\mathbf{Y} = \mathbf{A} = \{1..|\mathbf{A}|\}$.

2. The measurement model can be characterized with a simple *reliability measure* $p_c \stackrel{\Delta}{=} P(\mathbf{a} = c|\mathbf{y} = c)$, where $c \in \{1..|\mathbf{A}|\}$. The reliability measure p is the probability that a measurement $\mathbf{a}$ assumes the correct value c, assuming we know $\mathbf{y} = c$. In case the measurement does not agree, we assume the probability $P(\mathbf{a} \neq c|\mathbf{y} = c)$ of seeing any other measurement is equal to $q_c \stackrel{\Delta}{=} (1 - p_c)/(|A| - 1)$.

This model is more restrictive than a general discrete model, in that it cannot model if two labels are easily confused. However, note that every label c can have a different reliability p_c, i.e. we can model the fact that some labels are more reliably estimated than others.

9.2.3 The Perfect Measurement Model

In the simple discrete model above, we have $p = 1$ if the appearance measurements are absolutely reliable. Let us call this the *perfect measurement model*. For example, if in a computer vision application there are several easily distinguishable features, this could be an appropriate model.

9.2.4 A Simple Continuous Measurement Model

A simple continuous measurement model assumes we have n_c appearance measurements $\mathbf{a}_k = \{\mathbf{a}_{kc}|c \in 1..n_c\}$ that are simply copies of a corresponding set of appearance parameters $\mathbf{y}_j = \{\mathbf{y}_{jc}|c \in 1..n_c\}$, corrupted by i.i.d. normally distributed noise. Under those assumptions, the conditional probability $P(\mathbf{a}_k|\mathbf{m}_i^t, \mathbf{x}_j^t, \mathbf{y}_j)$ of a measurement vector $\mathbf{a}_k$ is a Gaussian distribution with diagonal covariance matrix $I\sigma^2$, and the appearance parameters $\mathbf{y}_j$ as the mean:

$$P(\mathbf{a}_k|\mathbf{M}^t, \mathbf{x}_j^t, \mathbf{y}_j) = \prod_{c=1}^{n_c} P(\mathbf{a}_{kc}|\mathbf{y}_{jc}) = (2\pi\sigma^2)^{-\frac{n_c}{2}} \exp\left\{-\frac{1}{2\sigma^2}\sum_{c=1}^{n_c}(\mathbf{a}_{kc} - \mathbf{y}_{jc})^2\right\}$$

This simple model is appropriate when the appearance is modeled using predicted pixel values, neglecting the geometric situation and correlations between the measured pixel

values that can be introduced in the image formation process. This model is frequently used in RANSAC based methods and also underlies most stereo work. However, its obvious disadvantage is that it is not able to withstand large rotations or displacements between the images, as in that case the appearance of features can change substantially.

This simple Gaussian model can also be used to predict grayscale or color invariants, such as described in (Schmid and Mohr, 1997) and (Montesinos et al., 1998). These multi-dimensional quantities are calculated to provide invariance with respect to rotation and translation. However, general 3D invariants are not available.

9.3 EM with Appearance

If all we are interested in is the structure and motion Θ, then in principle we need to *integrate out* the hidden appearance parameters $\mathbf{Y}$. In contrast, if we were to simultaneously estimate the appearance as well, the resulting structure and motion estimates would be biased. This is because the resulting estimate of the geometry is associated with one set of appearance parameters only, while there might be other appearance parameters that are almost as plausible. This is completely analogous to the bias we have if a single, "best" set of correspondences is obtained rather than considering a distribution over them.

In this section I show how appearance can be integrated out in the E-step, and how it will influences the posterior distribution over correspondence assignments $\mathbf{J}$. The re-estimation of the structure and motion estimate Θ^{t+1} in the M-step, however, will not be affected.

The biggest disadvantage to taking this approach is that sampling will now no longer decouple over the respective images, i.e. we need to sample over joint correspondence vectors $\mathbf{J}$ instead of sampling image correspondence vectors $\mathbf{j}_i$ separately. How this can be done will be explained in the next section, Section 9.4.

9.3.1 EM with Appearance

If appearance information is available, the E-step needs to be adapted, but the M-step remains the same. This is shown below.

Suppose that, aside from location, additional measurement data $\mathbf{A}$ is available about the *appearance* of the features detected in the images. As in previous chapters, we want to find

the MAP estimate $\mathbf{\Theta}^*$ of structure and motion $\mathbf{\Theta}$, but now given both location information $\mathbf{U}$ and appearance information $\mathbf{A}$. That is

$$\mathbf{\Theta}^* = \underset{\mathbf{\Theta}}{\text{argmax}} \; P(\mathbf{\Theta}|\mathbf{U}, \mathbf{A})$$

Analogous to the previous chapters, the total likelihood is found by integrating over all possible correspondence vectors $\mathbf{J}$:

$$\mathbf{\Theta}^* = \underset{\mathbf{\Theta}}{\text{argmax}} \; \sum_{\mathbf{J}} P(\mathbf{J}, \mathbf{\Theta}|\mathbf{U}, \mathbf{A})$$

Because this is intractable in general, we instead use the EM-algorithm, which iteratively maximizes the sum of the log-prior on $\mathbf{\Theta}$ and the expected log-likelihood $Q^t(\mathbf{\Theta})$, where

$$Q^t(\mathbf{\Theta}) \triangleq \sum_{\mathbf{J}} P(\mathbf{J}|\mathbf{U}, \mathbf{A}, \mathbf{\Theta}^t) \log P(\mathbf{U}, \mathbf{A}, \mathbf{J}|\mathbf{\Theta})$$

This is the analogous to expression 4.3 on page 55, but now incorporating appearance information $\mathbf{A}$.

The **M-step** will be as before. As always, in the M-step, we optimize for structure and motion $\mathbf{\Theta}$:

$$\mathbf{\Theta}^{t+1} = \underset{\mathbf{\Theta}}{\text{argmax}} \; \langle \log P(\mathbf{U}, \mathbf{A}, \mathbf{J}|\mathbf{\Theta}) \rangle + \log P(\mathbf{\Theta}) \tag{9.6}$$

As the appearance information $\mathbf{A}$ does not influence the geometric estimation of structure and motion $\mathbf{\Theta}$, we have $\log P(\mathbf{U}, \mathbf{A}, \mathbf{J}|\mathbf{\Theta}) = \log P(\mathbf{U}, \mathbf{J}|\mathbf{\Theta})$, which is the same as before. Therefore, the M-step does not change.

However, the **E-step** does change. Recall, in the E-step we need to compute (or estimate) the marginal probabilities of the correspondence posterior probability

$$P(\mathbf{J}|\mathbf{U}, \mathbf{A}, \mathbf{\Theta}^t)$$

which now is also conditioned on appearance information. *The appearance yields information on which measurements $\mathbf{u}_{ik}$ are likely to correspond to the same feature $\mathbf{x}_j$, and hence the posterior distribution over correspondences changes.* The E-step is modified in that the posterior $P(\mathbf{J}|\mathbf{U}, \mathbf{A}, \mathbf{\Theta}^t)$ will now have an additional appearance factor in it. To see this, apply the chain rule:

$$P(\mathbf{J}|\mathbf{U}, \mathbf{A}, \mathbf{\Theta}^t) \propto P(\mathbf{U}, \mathbf{A}, \mathbf{J}, \mathbf{\Theta}^t) = P(\mathbf{U}|\mathbf{A}, \mathbf{J}, \mathbf{\Theta}^t) P(\mathbf{A}|\mathbf{J}, \mathbf{\Theta}^t) P(\mathbf{J}|\mathbf{\Theta}^t) P(\mathbf{\Theta}^t)$$

As $\mathbf{\Theta}^t$ is given at the time of the E-step, the prior $P(\mathbf{\Theta}^t)$ is a constant. We also make the following assumption: given the correspondence $\mathbf{J}$ and the structure and motion guess

$\mathbf{\Theta}^t$, the location of features is independent of the appearance $\mathbf{A}$, i.e. $P(\mathbf{U}|\mathbf{A}, \mathbf{J}, \mathbf{\Theta}^t) = P(\mathbf{U}|\mathbf{J}, \mathbf{\Theta}^t)$. Thus, the posterior is the product of a location likelihood $P(\mathbf{U}|\mathbf{J}, \mathbf{\Theta}^t)$, an *appearance likelihood* $P(\mathbf{A}|\mathbf{J}, \mathbf{\Theta}^t)$, and a correspondence prior $P(\mathbf{J}|\mathbf{\Theta}^t)$:

$$P(\mathbf{J}|\mathbf{U}, \mathbf{A}, \mathbf{\Theta}^t) \propto P(\mathbf{A}|\mathbf{J}, \mathbf{\Theta}^t) P(\mathbf{U}|\mathbf{J}, \mathbf{\Theta}^t) P(\mathbf{J}|\mathbf{\Theta}^t)$$

From Chapter 7 (equation 7.22 on page 136) we know that, when using a simple visibility model, the product of the latter two factors is given by

$$P(\mathbf{U}|\mathbf{J}, \mathbf{\Theta}^t) P(\mathbf{J}|\mathbf{\Theta}^t) \propto \prod_i \alpha^{S_i} \exp\left[-w(\mathbf{j}_i)\right]$$

What remains to be done is obtain an expression for the appearance likelihood $P(\mathbf{A}|\mathbf{J}, \mathbf{\Theta}^t)$. Note that this is a different expression from the measurement model (9.5), as the appearance parameters $\mathbf{Y}$ are assumed unknown. In the section below I show that in order to obtain an expression for $P(\mathbf{A}|\mathbf{J}, \mathbf{\Theta}^t)$, we need to *integrate* over the hidden appearance parameters $\mathbf{Y}$.

9.3.2 Integrating over Unknown Appearance

In the E-step we need to integrate over the unknown appearance $\mathbf{Y}$ of the structure. Recall that we are interested in the appearance likelihood $P(\mathbf{A}|\mathbf{J}, \mathbf{\Theta}^t)$. To evaluate it, we need to integrate over all possible values for the appearance parameters $\mathbf{Y}$:

$$P(\mathbf{A}|\mathbf{J}, \mathbf{\Theta}^t) = \int_{\mathbf{Y}} P(\mathbf{A}|\mathbf{J}, \mathbf{\Theta}^t, \mathbf{Y}) P(\mathbf{Y}|\mathbf{\Theta}^t) \tag{9.7}$$

where $P(\mathbf{Y}|\mathbf{\Theta}^t)$ is a prior on appearance.

Note that the structure and motion estimate $\mathbf{\Theta}^t$ is computed in the M-step, and might or might not be necessary in the calculation of the likelihood $P(\mathbf{A}|\mathbf{J}, \mathbf{\Theta}^t, \mathbf{Y})$. In fact, in general a value for $\mathbf{\Theta}^t$ is only needed if the geometric dependence of the appearance in the image is modeled, e.g. for an oriented surface patch. In the case that there is no geometric dependence, we have

$$P(\mathbf{A}|\mathbf{J}, \mathbf{\Theta}^t, \mathbf{Y}) = P(\mathbf{A}|\mathbf{J}, \mathbf{Y})$$

A similar comment holds for the prior $P(\mathbf{Y}|\mathbf{\Theta}^t)$: it is conditioned on our current guess $\mathbf{\Theta}^t$ for the geometry, i.e. if we wanted we could model effects like “features close in space have similar appearance”.

Unlike $\mathbf{\Theta}^t$, the values of the appearance parameters $\mathbf{Y}$, are not computed in the M-step: they are nuisance variables. This is the reason why $\mathbf{Y}$ has to be integrated out in the E-step. This type of reasoning has been applied in a different context, as well, by Pasula in (Pasula et al., 1999). In the next few sections it is shown how, under mild assumptions, this can be done in a tractable manner.

9.3.3 The Appearance Likelihood as a Product of Set Scores

Expression 9.5 tells us what the likelihood of a given correspondence vector $\mathbf{J}$ is, given the appearance measurements $\mathbf{A}$ *and* the structure appearance parameters $\mathbf{Y}$. However, recall that we need to integrate out the appearance parameters $\mathbf{Y}$. Substituting (9.5) into expression 9.7 on the preceding page we obtain:

$$P(\mathbf{A}|\mathbf{J},\mathbf{\Theta}^t) = L_0(S_0)\int_{\mathbf{Y}} P(\mathbf{Y})\prod_{j=1}^{n}\prod_{\mathbf{a}_k\in S_j} P(\mathbf{a}_k|\mathbf{M}^t,\mathbf{x}_j^t,\mathbf{y}_j) \tag{9.8}$$

Let us assume that the appearances $\mathbf{y}_j$ of the features are *a priori* independent of each other and of the geometry $\mathbf{\Theta}^t$, i.e.

$$P(\mathbf{Y}|\mathbf{\Theta}^t) = \prod_{j=1}^{n} P(\mathbf{y}_j) \tag{9.9}$$

In that case, we can perform the integration separately for each feature $\mathbf{x}_j$:

$$P(\mathbf{A}|\mathbf{J},\mathbf{\Theta}^t) = L_0(S_0)\prod_{j=1}^{n}\int_{\mathbf{y}_j} P(\mathbf{y}_j)\prod_{\mathbf{a}_k\in S_j} P(\mathbf{a}_k|\mathbf{M}^t,\mathbf{x}_j^t,\mathbf{y}_j)$$

If we define the *set score* $L(S_j)$ as

$$L(S_j) \triangleq P(S_j|\mathbf{J},\mathbf{M}^t,\mathbf{x}_j^t) = \int_{\mathbf{y}_j} P(\mathbf{y}_j)\prod_{\mathbf{a}_k\in S_j} P(\mathbf{a}_k|\mathbf{M}^t,\mathbf{x}_j^t,\mathbf{y}_j) \tag{9.10}$$

we finally obtain

$$P(\mathbf{A}|\mathbf{J},\mathbf{\Theta}^t) = L_0(S_0)\prod_{j=1}^{n} L(S_j) \tag{9.11}$$

The intuition is this: the appearance likelihood of the correspondence $\mathbf{J}$ is the product of (a) the spurious appearance likelihood $L_0(S_0)$, and (b) n likelihood factors or set scores $L(S_j)$. The set score $L(S_j)$ computes how likely it is that a certain set of measurements S_j are associated with each other, given their appearance. The latter factor is an integral,

as all possible values for $\mathbf{y}_j$ have to be "compared" with the joint appearance of the set S_j. Another way to view the set score $L(S_j)$ is as the joint probability of the appearance measurements $\mathbf{a}_k \in S_j$. The score or appearance likelihood for the entire correspondence vector $\mathbf{J}$ is obtained by multiplying all these set scores (and the spurious score).

This calculation has to be done for every possible correspondence vector, which yields a ranking of correspondence in terms of appearance. Note that the likelihood scores are *not* probabilities, and do not have to sum up to 1. To yield a probability distribution over the $\mathbf{J}$, the appearance likelihood scores would still have to be multiplied with the location likelihood and correspondence prior, and renormalized.

A Simple Example

The calculation of the appearance likelihood using set scores, via (9.11), can again be illustrated with the example from Figure 9.1. For the prior on appearance, let us assume that squares are more common than triangles, e.g. $P(T) = 0.6$ and $P(T) = 0.4$. Then the appearance likelihood of the (shown) correspondence vector $\mathbf{J} = [1212]$ is

$$\begin{aligned} P(\mathbf{A}|[1212]) &= L(S_1)L(S_2) \\ &= \left[\sum_{\mathbf{y}_1} P(\mathbf{y}_1) \prod_{\mathbf{a}_k \in \{\mathbf{a}_1, \mathbf{a}_3\}} P(\mathbf{a}_k|\mathbf{y}_1)\right] \left[\sum_{\mathbf{y}_2} P(\mathbf{y}_2) \prod_{\mathbf{a}_k \in \{\mathbf{a}_2, \mathbf{a}_4\}} P(\mathbf{a}_k|\mathbf{y}_2)\right] \\ &= \left[P(S)P(s|S)^2 + P(T)P(s|T)^2\right] \left[P(S)P(t|S)^2 + P(T)P(t|T)^2\right] \\ &= \left[0.6 \times 0.9^2 + 0.4 \times 0.2^2\right] \left[0.6 \times 0.1^2 + 0.4 \times 0.8^2\right] \\ &= [0.486 + 0.016]\,[0.006 + 0.256] = 0.502 \times 0.262 = 0.132 \end{aligned}$$

It is also instructive to follow the calculation in case spurious features are allowed. In that case, we need to specify the probability of a spurious measurement. Let us assume s and t are equally probable: $P_0(s) = 0.5$, and $P_0(t) = 0.5$. Let us examine the appearance likelihood for a correspondence vector $J = [0212]$ that assigns measurement $\mathbf{a}_1$ to be spurious, but all others correctly. By definition 9.3, we have

$$L_0(S_0) = P_0(\mathbf{a}_1) = 0.5$$

Then, noting that now $S_1 = \{3\}$, the score for $J = [0212]$ can be computed as

$$\begin{aligned} P(\mathbf{A}|[1212]) &= L_0(S_0)L(S_1)L(S_2) \\ &= 0.5 \left[\sum_{\mathbf{y}_1} P(\mathbf{y}_1)P(\mathbf{a}_3|\mathbf{y}_1)\right]\left[\sum_{\mathbf{y}_2} P(\mathbf{y}_2) \prod_{\mathbf{a}_k \in \{\mathbf{a}_2, \mathbf{a}_4\}} P(\mathbf{a}_k|\mathbf{y}_2)\right] \\ &= 0.5 \left[P(S)P(s|S) + P(T)P(s|T)\right]\left[P(S)P(t|S)^2 + P(T)P(t|T)^2\right] \\ &= 0.5 \left[0.6 \times 0.9 + 0.4 \times 0.2\right]\left[0.6 \times 0.1^2 + 0.4 \times 0.8^2\right] \\ &= 0.5 \left[0.54 + 0.08\right]\left[0.006 + 0.256\right] = 0.5 \times 0.62 \times 0.262 = 0.081 \end{aligned}$$

Comparing this to the non-spurious example, we see that the set score $L(S_1; \mathbf{A})$ of the set $S_1 = \{3\}$ has increased. This is to be expected: the joint probability of a smaller set of measurements is expected to be higher than that of a larger set. However, the spurious appearance likelihood makes this assignment less likely than the correct one.

9.3.4 Set Scores for Simple Discrete Appearance Models

The Simple Discrete Measurement Model

Under the assumption that each appearance label $\mathbf{y}$ is equally probable a priori, the set scores are particularly simple to calculate for the simple discrete measurement model from Section 9.2.2 on page 165. Indeed, under the assumption that the prior $P(\mathbf{y})$ is uniform, i.e.

$$P(\mathbf{y}) = 1/n_s$$

where n_s is the number of different symbols, the set score (S_j) can be calculated as

$$\begin{aligned} L(S_j) &= \sum_{c=1}^{n_s} P(\mathbf{y}_j = c) \prod_{\mathbf{a}_k \in S_j} P(\mathbf{a}_k|\mathbf{y}_j = c) \\ &\propto \sum_{c=1}^{n_s} p_c^{N_{jc}} q_c^{|S_j|-N_{jc}} \end{aligned}$$

where $N_{jc} \triangleq |\{\mathbf{a}_k \in S_j | \mathbf{a}_k = c\}|$ is the number of measurements in set S taking on the value c. Note that $0 \leq |S_j| \leq m$, and $0 \leq N_{jc} \leq |S_j|$, and in an implementation the values $k(c, N_{jc}, |S_j|) \triangleq p_c^{N_{jc}} q_c^{|S_j|-N_{jc}}$ can be precomputed.

The Perfect Measurement Model

If the appearance measurements are absolutely reliable, we have a perfect appearance model, and the set scores are simply binary, measuring whether an assignment $\mathbf{J}$ is consistent given the appearance measurements $\mathbf{A}$ or not:

$$L(S_j) = \delta(N_j, |S_j|)$$

Here N_j is defined as the number of majority votes. In other words, the set score is 1 if all measurements agree, and 0 otherwise. This result holds for arbitrary appearance priors $P(\mathbf{y}_j)$.

9.4 Sampling Joint Correspondence Vectors

When incorporating appearance, sampling over correspondences changes substantially in one respect: we can no longer sample image correspondence vectors for each image in isolation. The measurement sets S_j in the calculation of the appearance likelihood (9.11) span multiple images. Any change in the set membership induced by modifying the correspondence vector $\mathbf{J}$ will change the set score $L(S_j)$ and hence the appearance likelihood $P(\mathbf{A}|\mathbf{J}, \mathbf{\Theta}^t)$. This means that, if we were to sample image correspondence vectors $\mathbf{j}_i$ in isolation, the appearance likelihood depends on the correspondence assignments in all other images.

Sampling over joint correspondences assignments $\mathbf{J}$ is challenging, as the proposal distributions from Chapter 5 were designed for the single image case. This leads to poor convergence behavior in the joint image case. While the use of importance sampling can alleviate some of that, the underlying problem remains essentially unsolved. The only exception is the case of the perfect discrete appearance measurement model, or when we know the appearance partition sizes for a discrete model (see below). The problem can be avoided, at a cost, by incorporating appearance estimation in the M-step, as will be discussed in the next section, Section 9.5.

9.4.1 A Modified Proposal Strategy

As discussed in the previous paragraphs, we have to sample over aggregate correspondence vectors $\mathbf{J}$ when incorporating appearance. However, we can use almost exactly the same

proposal strategies as in the previous chapters, i.e. chain flipping and smart chain flipping. Instead of sampling in the images independently, we now sample over the entire correspondence $\mathbf{J}$. To propose a change to $\mathbf{J}$, the following strategy is proposed:

1. Choose an image i at random. In contrast to before, we can no longer sample in images in isolation, but it is perfectly valid to limit the action of the proposal step to individual images.

2. Propose a change to $\mathbf{j}_i$, exactly as in Chapters 5 and 7 (specifically, Section 7.4 on page 139).

3. Calculate the original acceptance ratio a, using equation 7.30 on page 140:

$$a_{SMART} = \prod_{u \in p} \frac{1 - q(u, \mathbf{J}(u))}{1 - q(u, \mathbf{J}'(u))}$$

   where p is the proposed alternating path, and the $q(u, \mathbf{J}(u))$ are the modified transition probabilities in the smart Markov chain MC.

4. Multiply the acceptance ratio a_{SMARt} with the appearance likelihood factor:

$$a = a_{SMART} \times \frac{P(\mathbf{A}|\mathbf{J}', \mathbf{\Theta}^t)}{P(\mathbf{A}|\mathbf{J}, \mathbf{\Theta}^t)} \tag{9.12}$$

   where $\mathbf{J}'$ is the proposal correspondence vector. Recall that the appearance likelihood $P(\mathbf{A}|\mathbf{J}, \mathbf{\Theta}^t)$ is given by a product of set scores (equation 9.11 on page 170):

$$P(\mathbf{A}|\mathbf{J}, \mathbf{\Theta}^t) = L_0(S_0) \prod_{j=1}^{n} L(S_j)$$

Note that the random choice of the image in which to change the assignment has no effect on the acceptance ratio. The only substantial change in the calculation of the acceptance ratio is the calculation of the appearance likelihood factor in (9.12).

9.4.2 Statistics on Appearance

In order to monitor the behavior of the sampler, we can look at the probability of the appearance parameters $\mathbf{Y}$ computed in the E-step. Specifically, it would be interesting to see how the probability over the individual appearance parameters $\mathbf{y}_j$ changes over time.

Given the conditional independence assumptions made above we can do this easily. In particular, we are interested in

$$P(\mathbf{y}_j|\mathbf{U},\mathbf{A},\mathbf{\Theta}^t) = \sum_{\mathbf{J}} P(\mathbf{y}_j|\mathbf{U},\mathbf{A},\mathbf{J},\mathbf{\Theta}^t)P(\mathbf{J}|\mathbf{U},\mathbf{A},\mathbf{\Theta}^t) = \left\langle P(\mathbf{y}_j|\mathbf{U},\mathbf{A},\mathbf{J},\mathbf{\Theta}^t)\right\rangle$$

where the expectation is taken with respect to the correspondence posterior $P(\mathbf{J}|\mathbf{U},\mathbf{A},\mathbf{\Theta}^t)$, i.e. the very same one we are computing in the E-step. The posterior $P(\mathbf{y}_j|\mathbf{U},\mathbf{A},\mathbf{J},\mathbf{\Theta}^t)$ is assumed conditionally independent of the location measurements $\mathbf{U}$ and, applying Bayes law, is proportional to the product of the likelihood and the prior:

$$P(\mathbf{y}_j|\mathbf{U},\mathbf{A},\mathbf{J},\mathbf{\Theta}^t) = P(\mathbf{y}_j|\mathbf{A},\mathbf{J},\mathbf{\Theta}^t) = \frac{P(\mathbf{A}|\mathbf{J},\mathbf{\Theta}^t,\mathbf{y}_j)P(\mathbf{y}_j|\mathbf{J},\mathbf{\Theta}^t)}{P(\mathbf{A}|\mathbf{J},\mathbf{\Theta}^t)}$$

We have already obtained the likelihood factor $P(\mathbf{A}|\mathbf{J},\mathbf{\Theta}^t,\mathbf{y}_j)$ as part of the appearance likelihood (equation 9.5). It can be computed by a product over all appearance measurements in the set S_j that is associated with feature $\mathbf{x}_j$ given the correspondence $\mathbf{J}$:

$$P(\mathbf{A}|\mathbf{J},\mathbf{\Theta}^t,\mathbf{y}_j) = \prod_{\mathbf{a}_k \in S_j} P(\mathbf{a}_k|\mathbf{M}^t,\mathbf{x}_j^t,\mathbf{y}_j)$$

We assumed before that the prior on appearance is independent of geometry, we have $P(\mathbf{y}_j|\mathbf{J},\mathbf{\Theta}^t) = P(\mathbf{y}_j)$. Given this, we finally have:

$$P(\mathbf{y}_j|\mathbf{U},\mathbf{A},\mathbf{\Theta}^t) \propto \left\langle \frac{P(\mathbf{y}_j)\prod_{\mathbf{a}_k \in S_j} P(\mathbf{a}_k|\mathbf{M}^t,\mathbf{x}_j^t,\mathbf{y}_j)}{P(\mathbf{A}|\mathbf{J},\mathbf{\Theta}^t)} \right\rangle$$

Comparing this with the definition 9.10 on page 170 of set scores, we see that the statistics we are after are nothing but the normalized posterior probability terms in the set scores. In the case of a symbolic measurement model, we can represent this as a $|\mathbf{A}| \times n$ table of posterior probabilities, which can be easily interpreted.

9.4.3 The Deadlock Problem

While the proposal strategy suggested above is theoretically valid, it leads to very poor convergence behavior in practice.

In fact, in the case of a perfect appearance model even theoretical guarantees on convergence disappear, as the resulting Markov chain is no longer irreducible, one of the requirements for convergence (Gilks et al., 1996; Robert and Casella, 1999). A Markov chain is

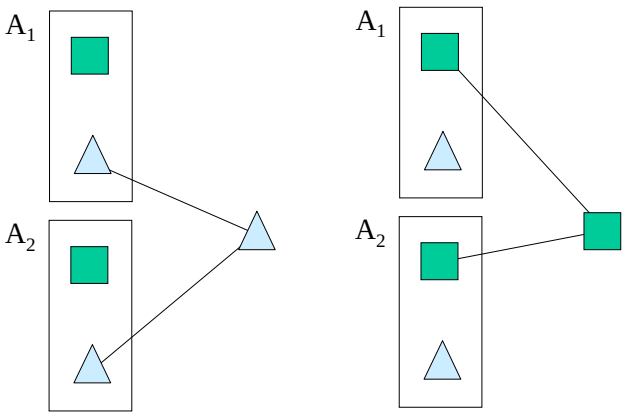


Figure 9.2: Example that illustrates the Markov chain over joint correspondences can be reducible under certain conditions. On the left, the true situation. On the right, a sampler state that can never reach the true state unless occlusion is allowed. See text for further explanation.

irreducible if all states communicate, i.e. from any state of the sampler there is a sequence of (accepted) proposals that can result in any other state. In the case of the perfect discrete appearance model from Section 9.2.3 this is no longer the case. To see this, consider the example in Figure 9.2. In the example, there are two images with associated appearance measurements $\mathbf{A}_1$ and $\mathbf{A}_2$. In the example we assume that occlusion is not allowed, but spurious measurements are. The true situation is the one on the left, where one triangular feature is observed in the two images, and there are two spurious measurements, each with a square appearance. However, if we start the joint correspondence sampler in the state corresponding to the right diagram, where the observed feature is thought to be square, we cannot transition to the true situation by changing the assignment in one image only. The reason is simple: because of the perfect appearance measurement model, any intermediate state would have probability zero, and hence such proposals will not be accepted.

Even though the Markov chain might be irreducible if we use a non-perfect measurement model, convergence will still be very poor. Since we propose a change of assignment in one image only, an incorrect assignment in another image can lock in the incorrect assignment, or make it very improbable that an intermediate state is visited. This is analogous to the problems with the flip proposal problem in the single image case: we need a concerted change not provided for by the proposal distribution. Even for non-perfect measurements,

this *deadlock problem* is no longer absolute, but the probability of overcoming it is very small. The same reasoning holds for continuous models.

9.4.4 Special Cases

No Occlusion or Clutter

In the case that there is no occlusion or clutter, the deadlock problem disappears. In addition, the sampling process is decomposed into $m \times n_s$ smaller sampling problems, where as before m is the number of images and n_s is the number of symbols. Indeed, after looking at a single image we can tell exactly (for a discrete appearance model) how many features there are of any given appearance type. In that case, any valid assignment partitions the correspondence vectors neatly along type boundaries. Since we cannot propose an assignment that changes the appearance (ironically, because of the deadlock problem), we only have to consider the subset of measurements and features that have the same type.

Known Partition Sizes

The same is true in the case that we know exactly how many features there are of each appearance type. Given this additional information, we can restrict ourselves to that part of the space of correspondences that do not change the number of features in each class.

Implementation

Results for the case that we know the partition sizes are given in the next chapter, Chapter 10. Implementing the sampler from Section 9.4.1 is particularly simple in this case: we simply have n_s smaller sampling problems that do not interact.

Indeed, since the partition sizes are known, the features can be partitioned beforehand and designated to be of a particular (discrete) appearance type. The sampler is then run, and on each iteration an image is randomly selected, after which an assignment change is proposed using the usual chain flipping machinery described in Chapter 5. The transition probabilities in the mini Markov chain remain the same, *except* when the proposed transition assigns a measurement of one type to a feature of another type. In that case, the transition probabilities are set to zero, corresponding to infinite edge weights. In other words, those edges that connect measurements with features of a different type are simply deleted from the bipartite graph.

9.4.5 Importance Sampling

If the partition sizes are not known, the modified sampler from Section 9.4.1 has very poor convergence properties, but a technique called *importance sampling* (Tanner, 1996) can to some extent alleviate the problem. The idea is this: instead of multiplying the acceptance ratio with the appearance likelihood ratio

$$\frac{P(\mathbf{A}|\mathbf{J}', \mathbf{\Theta}^t)}{P(\mathbf{A}|\mathbf{J}, \mathbf{\Theta}^t)}$$

as done in equation 9.12, we accord an importance weight equal to $P(\mathbf{A}|\mathbf{J}, \mathbf{\Theta}^t)$ to each accepted sample $\mathbf{J}$. In other words, we sample without regard to appearance, but weight each resulting sample to reflect how likely each sampled correspondence assignment $\mathbf{J}$. In this way, the sampler is not caught in near-trapping states, and the effect of appearance can be integrated nevertheless.

While some results using importance sampling are shown in Chapter 10, the problem of poor convergence has simply been replaced by a different problem, namely that of *high variance*. In particular, the more accurate appearance measurements are (whether they are discrete or continuous), the more extreme the appearance likelihood function $P(\mathbf{A}|\mathbf{J}, \mathbf{\Theta}^t)$ will be as a function of $\mathbf{J}$. If a given correspondence assignment is compatible with the appearance measurements $\mathbf{A}$, the likelihood and the associated importance weight will be very high. Conversely, if there is some inconsistency, the importance weight will be very low. Since there are many more inconsistent assignments than there are consistent ones, the importance weights will be dominated by a small set of very large values. This is a well known problem with importance sampling in general (Tanner, 1996).

More importantly, the problem becomes increasingly worse with the number of images m and the number of features n. Indeed, the state space over correspondences $\mathbf{J}$ grows combinatorially in m and n, but the number of consistent assignments does not. This virtually guarantees that the modified proposal strategy of Section 9.4.1 will fail.

There are two strategies to avoid this problem: (a) we can try to come up with new proposal strategies that are specially built for the multi-image assignment problem, and hopefully avoid the problems associated with proposing changes in isolated images, or (b) we can sidestep the problem and estimate appearance along with structure and motion, in which case the sampling once again decomposes over the different images. In this dissertation I have taken the latter approach, which is explained in detail in the next section, Section 9.5.

9.5 EM for Structure, Motion, and Appearance

9.5.1 Introduction

In order to sidestep the problems with sampling joint correspondence assignments $\mathbf{J}$, we can instead also estimate appearance along with structure and motion. In other words, in the **M-step**, in addition to estimating structure and motion $\mathbf{\Theta}$, we now also optimize for appearance parameters $\mathbf{Y}$:

$$\{\mathbf{\Theta}^{t+1}, \mathbf{Y}^{t+1}\} = \underset{\mathbf{\Theta},\mathbf{Y}}{\operatorname{argmax}} \ \langle \log P(\mathbf{U}, \mathbf{A}, \mathbf{J}|\mathbf{\Theta}, \mathbf{Y})\rangle + \log P(\mathbf{\Theta}) + \log(\mathbf{Y}) \tag{9.13}$$

where the priors $P(\mathbf{\Theta})$ and $P(\mathbf{Y})$ on geometry and appearance, respectively, are assumed independent. One needs to compare this with equation 9.6 on page 168, where (a) only the geometry is treated as an unknown, and (b) the appearance parameters $\mathbf{Y}$ do not appear, as there they are integrated out. In contrast, in (9.13) we treat the appearance parameters $\mathbf{Y}$ as unknown parameters of interest.

In the **E-step** we condition on the current estimate $\mathbf{\Theta}^t$ of structure and motion *and* on the current estimate $\mathbf{Y}^t$ for the appearance parameters $\mathbf{Y}$, to obtain the posterior probability over correspondences $\mathbf{J}$:

$$P(\mathbf{J}|\mathbf{U}, \mathbf{A}, \mathbf{\Theta}^t, \mathbf{Y}^t)$$

The advantage with respect to joint sampling is that this will now decompose over images, i.e. we can sample in each image in isolation. The disadvantage is that we are *not* obtaining an unbiased structure and motion estimate: it will depend on the concurrently found estimate for appearance $\mathbf{Y}$.

9.5.2 The M-Step: Re-estimating Structure, Motion, and Appearance

In the M-step, we re-estimate structure and motion $\mathbf{\Theta}$, and the appearance parameters $\mathbf{Y}$. The expected log-likelihood is given by

$$Q^t(\mathbf{\Theta}) \triangleq \langle \log P(\mathbf{U}, \mathbf{A}, \mathbf{J}|\mathbf{\Theta}, \mathbf{Y})\rangle$$

Applying the chain rule to the likelihood term within the expectation operator, we have

$$\log P(\mathbf{U}, \mathbf{A}, \mathbf{J}|\mathbf{\Theta}, \mathbf{Y}) = \log P(\mathbf{U}|\mathbf{A}, \mathbf{J}, \mathbf{\Theta}, \mathbf{Y}) + \log P(\mathbf{A}|\mathbf{J}, \mathbf{\Theta}, \mathbf{Y}) + \log P(\mathbf{J}|\mathbf{\Theta}, \mathbf{Y}) \tag{9.14}$$

The third term $P(\mathbf{J}|\mathbf{\Theta}, \mathbf{Y})$ above is a conditional prior on correspondence, given geometry $\mathbf{\Theta}$ and appearance $\mathbf{Y}$. This prior on $\mathbf{J}$ is independent of the geometry $\mathbf{\Theta}$ if a simple visibility model is used, as discussed in Section 7.2. Likewise, we will assume here that correspondence is *a priori* independent of the structure appearance $\mathbf{Y}$. Hence, the third term in (9.14) can be dropped from further consideration.

This remainder of the objective function consists of two terms, discussed in turn below:

1. If we assume that the location measurements $\mathbf{U}$ are conditionally independent of the appearance terms $\mathbf{A}$ and $\mathbf{Y}$, given $\mathbf{J}$ and $\mathbf{\Theta}$, we have

$$\log P(\mathbf{U}|\mathbf{A}, \mathbf{J}, \mathbf{\Theta}, \mathbf{Y}) = \log P(\mathbf{U}|\mathbf{J}, \mathbf{\Theta})$$

   This term is well known: it is nothing but the conventional structure and motion objective function. When taking its expectation with respect to the distribution over correspondences $\mathbf{J}$, we have the familiar virtual measurements formulation from Chapter 4 (equation 4.14 on page 59):

$$\langle \log P(\mathbf{U}|\mathbf{J}, \mathbf{\Theta}) \rangle \equiv -\frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{n} (\mathbf{v}_{ij}^t - \mathbf{h}_{ij})^T \mathbf{R}_{ij}^{-1} (\mathbf{v}_{ij}^t - \mathbf{h}_{ij})$$

   with the virtual measurements $\mathbf{v}_{ij}^t$ and covariance matrices $\mathbf{R}_{ij}$ defined as before (equations 4.15 and 4.14, on page 59).

2. The second term has been encountered as well: it is the appearance likelihood, given $\mathbf{Y}$, which can be expressed in terms of measurement sets S_j (equation 9.5 on page 164). In $\log$ terms, and dropping the spurious term (as it does not depend on the unknowns), we have:

$$\log P(\mathbf{A}|\mathbf{J}, \mathbf{\Theta}, \mathbf{Y}) \equiv \sum_{j=1}^{n} \sum_{\mathbf{a}_k \in S_j} \log P(\mathbf{a}_k|\mathbf{m}_i, \mathbf{x}_j, \mathbf{y}_j)$$

   Note that this term involves the geometry-related unknowns $\mathbf{m}_i$ and $\mathbf{x}_j$. This means that, if the appearance measurement model is geometry dependent (e.g. in the case of oriented surface patches), the resulting optimization problem is coupled. However, in all the measurement models considered in this dissertation, appearance measurements are independent of geometry, i.e.

$$\log P(\mathbf{A}|\mathbf{J}, \mathbf{\Theta}, \mathbf{Y}) = \log P(\mathbf{A}|\mathbf{J}, \mathbf{Y}) \equiv \sum_{j=1}^{n} \sum_{\mathbf{a}_k \in S_j} \log P(\mathbf{a}_k|\mathbf{y}_j)$$

If the simple continuous measurement model from Section 9.2.4 is used, this is nothing but a maximum likelihood criterion for a mixture of Gaussians. Maximizing the expected log-likelihood is as simple as estimating the means (and possibly covariances) of the Gaussians by a weighted average, where the weights are exactly the marginal probabilities f_{ijk}^t estimated in the E-step (equation 4.18 in Section 4.4.6).

Since the geometry and appearance related estimation problems neatly decouple, the M-step can be summarized as:

1. Solve for optimal structure and motion $\mathbf{\Theta}^{t+1}$, given virtual measurements $\mathbf{v}_{ij}^t$ and virtual covariance $\mathbf{R}_{ij}$. The appropriate algorithm to use depends on the application.
2. Re-estimate the appearance parameters $\mathbf{y}_j$ for each feature.

Application: The Simple Continuous Model

In the case of the simple model from Section 9.2.4, the appearance is estimated by a weighted average of the appearance measurements $\mathbf{a}_k$. For each component $\mathbf{y}_{jc}$ we have

$$\mathbf{y}_{jc}^{t+1} = \frac{\sum_i \sum_k f_{ijk}^t \mathbf{a}_{kc}}{\sum_i \sum_k f_{ijk}^t}$$

and, if the variances σ_{jc}^2 are unknown:

$$(\sigma_{jc}^2)^{t+1} = \frac{\sum_i \sum_k f_{ijk}^t (\mathbf{a}_{kc} - \mathbf{y}_{jc}^{t+1})^2}{\sum_i \sum_k f_{ijk}^t}$$

9.5.3 The E-Step: Approximating Marginal Correspondence Probabilities given an Appearance Estimate

In the E-step we need to estimate the marginal correspondence probabilities given the current estimates $\mathbf{\Theta}^t$ and $\mathbf{Y}^t$ respectively for structure and motion, and appearance. If we make the same independence assumptions as in the M-step, the conditional posterior over correspondences $\mathbf{J}$ is

$$P(\mathbf{J}|\mathbf{U}, \mathbf{A}, \mathbf{\Theta}^t, \mathbf{Y}^t) \propto P(\mathbf{U}, \mathbf{A}, \mathbf{J}, \mathbf{\Theta}^t, \mathbf{Y}^t) \propto P(\mathbf{J}|\mathbf{\Theta}^t) P(\mathbf{U}|\mathbf{J}, \mathbf{\Theta}^t) P(\mathbf{A}|\mathbf{J}, \mathbf{\Theta}^t, \mathbf{Y}^t)$$

Thus, the posterior can be seen as consisting of two parts: one factor measures the geometric consistency of a correspondence, and the other part measures the appearance consistency. Since we condition on both geometry $\mathbf{\Theta}^t$ and estimated appearance $\mathbf{Y}^t$ from the M-step, this entire expression can be factored over the images. To see this, consider each of the factors in turn:

1. The first two factors, $P(\mathbf{J}|\mathbf{\Theta}^t)P(\mathbf{U}|\mathbf{J},\mathbf{\Theta}^t)$, are together the posterior probability of the correspondence $\mathbf{J}$ given the location measurements $\mathbf{U}$ and an estimate for structure and motion $\mathbf{\Theta}^t$. We know from the previous chapters that, under some mild assumptions, this probability decomposes over the different images:

$$P(\mathbf{U}|\mathbf{J},\mathbf{\Theta}^t)P(\mathbf{J}|\mathbf{\Theta}^t) = \prod_{i=1}^{m} P(\mathbf{U}_i|\mathbf{j}_i,\mathbf{m}_i^t,\mathbf{X}^t)P(\mathbf{J}|\mathbf{m}_i^t,\mathbf{X}^t)$$

   Recall from Chapter 5 that in that case, we can treat the E-step in terms of sampling weighted matchings in bipartite graphs. If we use a simple visibility model, we know from Chapter 7 that (equation 7.32 on page 140):

$$f_i^t(\mathbf{j}_i) \stackrel{\Delta}{=} P(\mathbf{U}_i|\mathbf{j}_i,\mathbf{m}_i^t,\mathbf{X}^t)P(\mathbf{J}|\mathbf{m}_i^t,\mathbf{X}^t) \propto \exp\left[-\sum_{k=1}^{K_i}\bar{w}(u_k,\mathbf{j}_i(u_k))\right] \qquad (9.15)$$

   where the augmented weights are defined by (7.31 on page 140):

$$\bar{w}(u,v) \stackrel{\Delta}{=} \begin{cases} -\log\alpha & \text{if } v = v_0 \\ w(u,v) & \text{otherwise} \end{cases}$$

   Here α depends on the amount of occlusion and clutter, v_0 is a special "spurious" vertex, and the weights $w(u,v)$ measure how far actual measurements u_k are from the projected features v_j, e.g. for 2D isotropic Gaussian noise we have (equation 7.34 on page 140):

$$\begin{aligned} w(u_k,v_j) &\stackrel{\Delta}{=} -\log P(\mathbf{u}_{ik}|\mathbf{m}_i,\mathbf{x}_{\mathbf{j}_{ik}}) \\ &= \log(2\pi\sigma^2) + \frac{1}{2\sigma^2}||\mathbf{u}_{ik} - \mathbf{h}(\mathbf{m}_i,\mathbf{x}_j)||^2 \end{aligned}$$

2. The appearance likelihood $P(\mathbf{A}|\mathbf{J},\mathbf{\Theta}^t,\mathbf{Y}^t)$ also decouples over the images, *given* the appearance parameters $\mathbf{Y}$. Indeed, we already know that the likelihood factors over all images and measurements as follows (slightly rewriting equation 9.2 on page 164:

$$P(\mathbf{A}|\mathbf{J},\mathbf{\Theta}^t,\mathbf{Y}^t) = \prod_{i=1}^{m} P(\mathbf{A}_i|\mathbf{j}_i,\mathbf{m}_i^t,\mathbf{X}^t\,\mathbf{Y}^t) = \prod_{i=1}^{m}\left[\prod_{\mathbf{j}_{ik}=0} P_0(\mathbf{a}_{ik}) \prod_{\mathbf{j}_{ik}\neq 0} P(\mathbf{a}_{ik}|\mathbf{m}_i^t,\mathbf{X}^t\,\mathbf{y}_{\mathbf{j}_{ik}}^t)\right]$$

This is the main advantage of taking appearance $\mathbf{Y}$ into the M-step: we can now sample for each image in isolation. If we assume as above that the appearance measurement is independent of the geometry, we can write the appearance likelihood for a given image correspondence vector $\mathbf{j}_i$ as:

$$P(\mathbf{A}_i|\mathbf{j}_i, \mathbf{m}_i^t, \mathbf{X}^t\,\mathbf{Y}^t) = P(\mathbf{A}_i|\mathbf{j}_i, \mathbf{Y}^t) = \prod_{\mathbf{j}_{ik}=0} P_0(\mathbf{a}_{ik}) \prod_{\mathbf{j}_{ik}\neq 0} P(\mathbf{a}_{ik}|\mathbf{y}_{\mathbf{j}_{ik}}) \tag{9.16}$$

Sampling Appearance-weighted Weighted Matchings

Since both parts decompose over the images, we can once again abstract away from the problem and simply sample over weighted matchings. However, we need to use newly computed weights that take appearance into account. The target distribution is the the image correspondence posterior $P(\mathbf{j}_i|\mathbf{U}_i, \mathbf{A}_i, \mathbf{m}_i^t, \mathbf{X}^t, \mathbf{Y}^t)$. To lessen the burden of notation, let us denote this as

$$f_{\mathbf{Y}i}^t(\mathbf{j}_i) \triangleq P(\mathbf{j}_i|\mathbf{U}_i, \mathbf{A}_i, \mathbf{m}_i^t, \mathbf{X}^t, \mathbf{Y}^t)$$

where the $\mathbf{Y}$ in the subscript indicates we are now also conditioning on an appearance estimate $\mathbf{Y}^t$. Combining equations (9.15) and (9.16) and rewriting the resulting expression we obtain a Gibbs distribution with new weights:

$$\begin{aligned} f_{\mathbf{Y}i}^t(\mathbf{j}_i) &= \exp\left[-\sum_{k=1}^{K_i} \bar{w}(u_k, \mathbf{j}_i(u_k))\right] \prod_{\mathbf{j}_{ik}=0} P_0(\mathbf{a}_{ik}) \prod_{\mathbf{j}_{ik}\neq 0} P(\mathbf{a}_{ik}|\mathbf{y}_{\mathbf{j}_{ik}}) \\ &= \exp\left[\sum_{\mathbf{j}_{ik}=0} (\log\alpha + \log P_0(\mathbf{a}_{ik})) + \sum_{\mathbf{j}_{ik}\neq 0} (\log P(\mathbf{u}_{ik}|\mathbf{m}_i, \mathbf{x}_{\mathbf{j}_{ik}}) + \log P(\mathbf{a}_{ik}|\mathbf{y}_{\mathbf{j}_{ik}}))\right] \\ &= \exp\left[-\sum_{k=1}^{K_i} \widetilde{w}(u_k, \mathbf{j}_i(u_k))\right] \end{aligned}$$

where the new weights are defined as

$$\widetilde{w}(u, v) \triangleq \begin{cases} -\log\alpha - \log P_0(\mathbf{a}_{ik}) & \text{if } v = v_0 \\ -\log P(\mathbf{u}_{ik}|\mathbf{m}_i, \mathbf{x}_{\mathbf{j}_{ik}}) - \log P(\mathbf{a}_{ik}|\mathbf{y}_{\mathbf{j}_{ik}}) & \text{otherwise} \end{cases}$$

Application: The Simple Continuous Model

In the case of the simple model from Section 9.2.4, i.e. independently measured appearance components (e.g. pixels), and with isotropic 2D Gaussian noise for the location measure-

ments, we have:

$$\widetilde{w}(u,v) \triangleq \begin{cases} -\log\alpha - \log P_0(\mathbf{a}_{ik}) & \text{if } v = v_0 \\ \log(2\pi\sigma^2) + \frac{1}{2\sigma^2}||\mathbf{u}_{ik} - \mathbf{h}(\mathbf{m}_i, \mathbf{x}_j)||^2 + \sum_c \frac{1}{2(\sigma_{jc}^2)^t}(\mathbf{a}_{ikc} - \mathbf{y}_{jc}^t)^2 & \text{otherwise} \end{cases}$$

The Spurious Likelihood Model

There is one more issue to resolve before we can implement this in practice: what should the spurious likelihood model $P_0(\mathbf{a})$ be ? We cannot simply drop this term from consideration, as this would unduly favor correspondences with more spurious measurements, as this minimizes the sum of the weights. Instead, the appearance-related penalty associated with spurious measurements should be on the same order as the penalty incurred for non-spurious measurements. For the simple model, this can be accomplished by using the same independent Gaussian model, with mean and covariance derived from the entire collection of measurements:

$$\mathbf{y}_{0c} = \frac{\sum_i \sum_k \mathbf{a}_{kc}}{\sum_i K_i}$$

$$\sigma_{0c}^2 = \frac{\sum_i \sum_k (\mathbf{a}_{kc} - \mathbf{y}_{0c})^2}{\sum_i K_i}$$

Chapter 10

Results for MCEM with Appearance Models

This chapter presents results obtained by incorporating appearance. Results are shown for both the joint correspondence sampling approach and the EM approach with re-estimating appearance.

The first approach is appropriate for discrete appearance models where there is no occlusion or clutter, or in the case that the partition sizes are known. Results for both binary and multi-valued symbols are shown below in Section 10.1. We also show one result, in Section 10.2, for which the partition sizes are not known. To implement this I used importance sampling. However, it is noted that the appearance model can only be used to provide a weak bias, in order to avoid high variance in the importance weights.

The second approach, where appearance is re-estimated in the M-step, is illustrated with pixel templates as the appearance model in Section 10.3.

10.1 Known Partition Sizes

10.1.1 Binary Symbols

In this section I illustrate a sequence with a perfect measurement model, with known partition sizes. The object in the 8 images has features that contain the color red and some that do not. The measurements are shown in Figure 10.1: both their location and symbolic

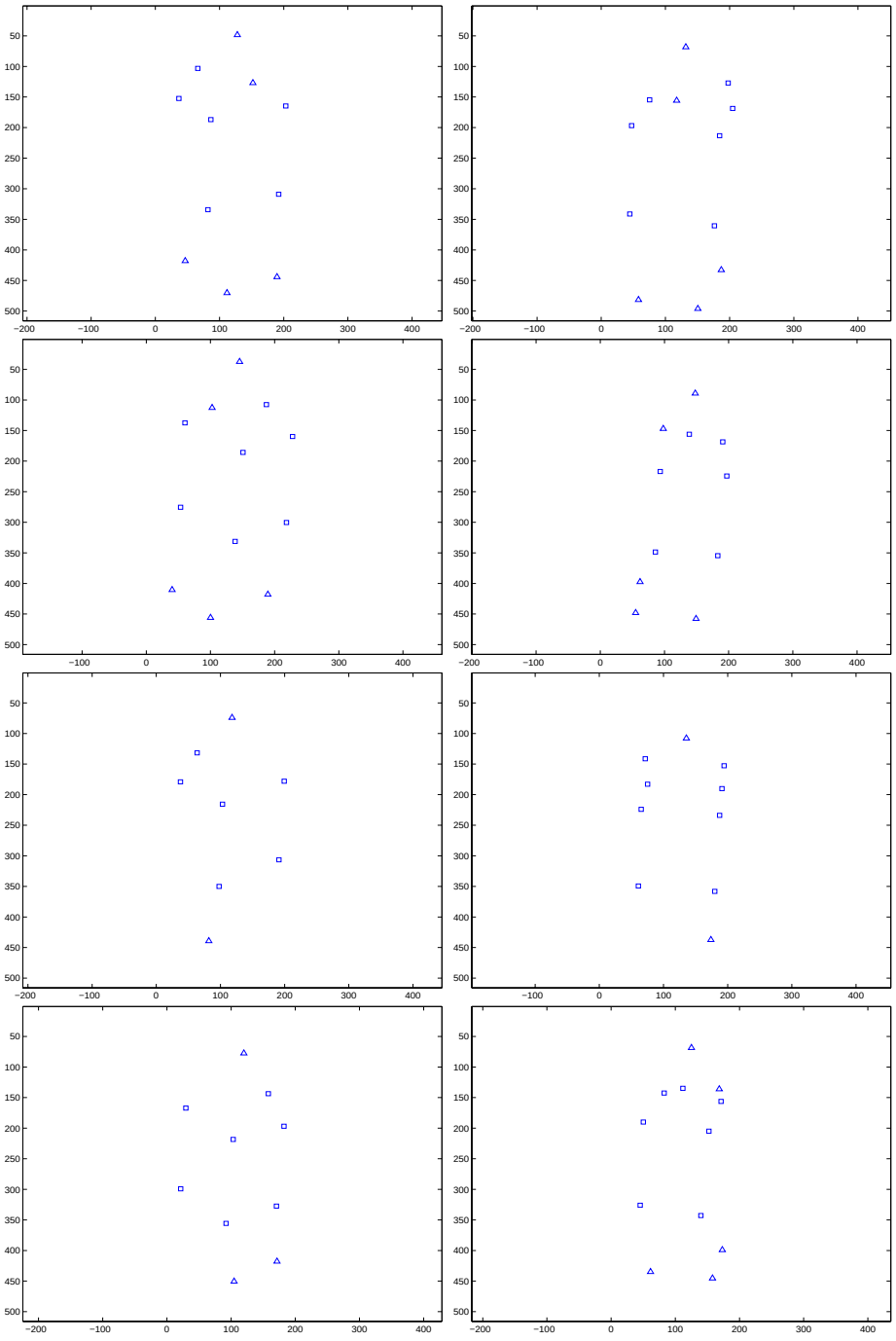
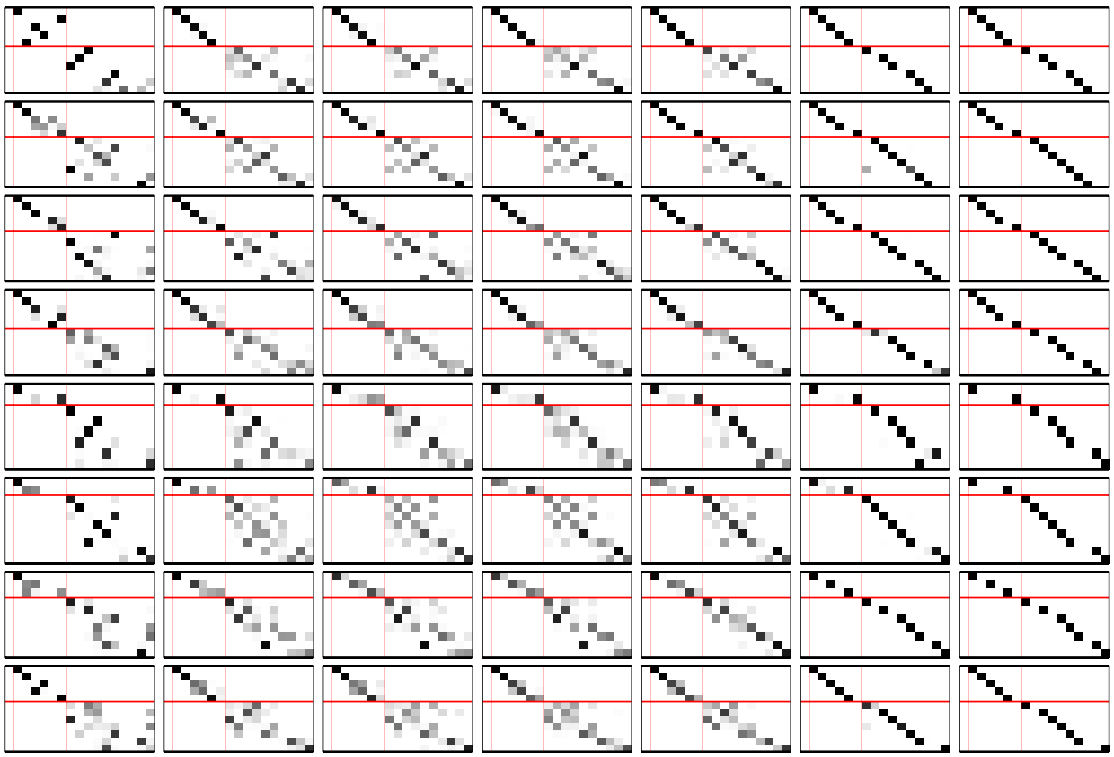


Figure 10.1: Measurements in the 8 input images.



(a) it 1 (b) it 2 (c) it 3 (d) it 5 (e) it 9 (f) it 17 (g) it 25

Figure 10.2: Marginal probabilities computed in the E-step, grouped according to type. Due to the perfect appearance model, there is no interaction between measurements and features of different type. Note that, as before, the first column is reserved to indicate spurious measurements, of which there are none here.

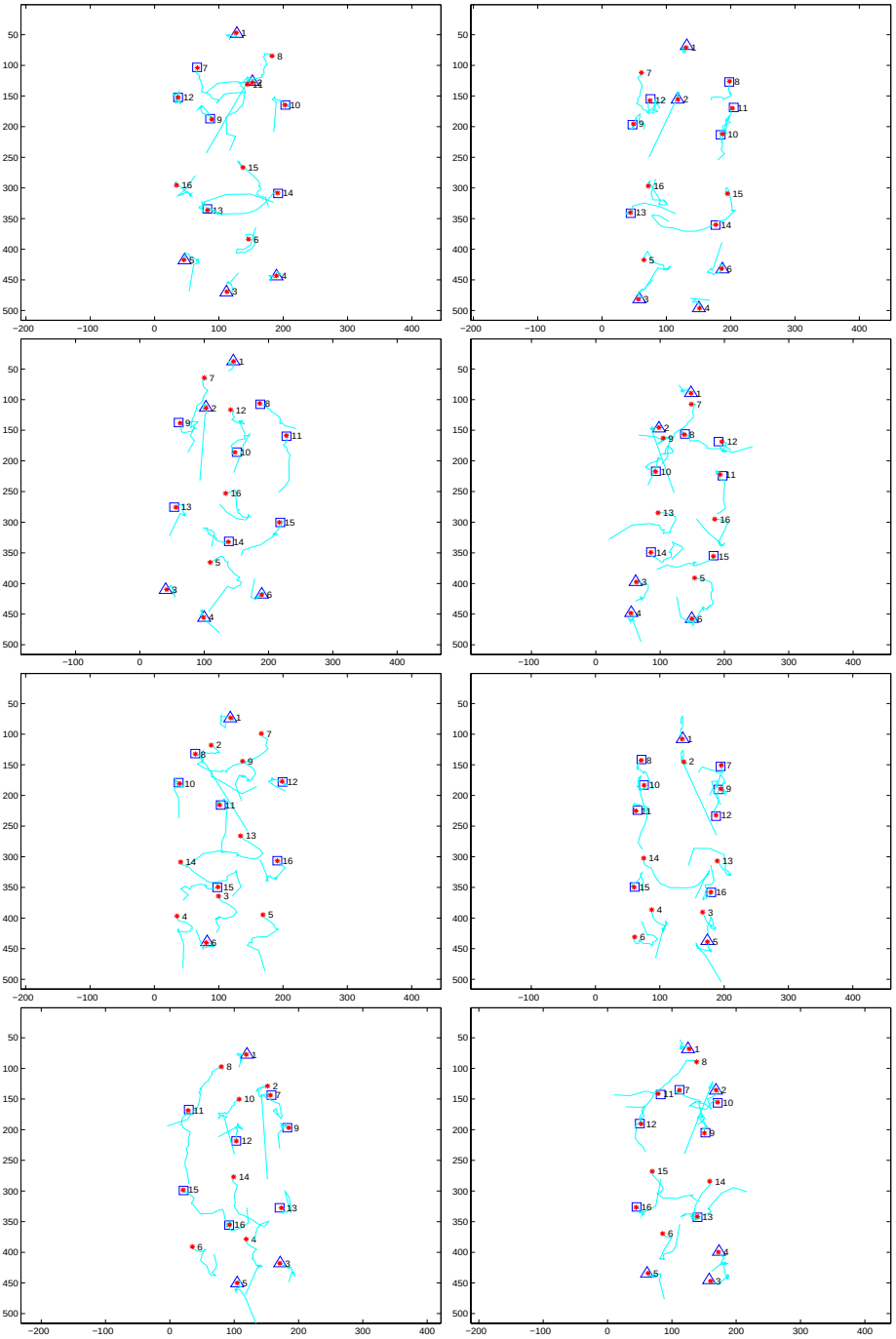


Figure 10.3: Plot of the predicted location for each of the features over time. The last predicted location is marked with an asterisk. Measurements are shown as squares and triangles. Note that in every image some features are occluded, in which case no measurement is shown.

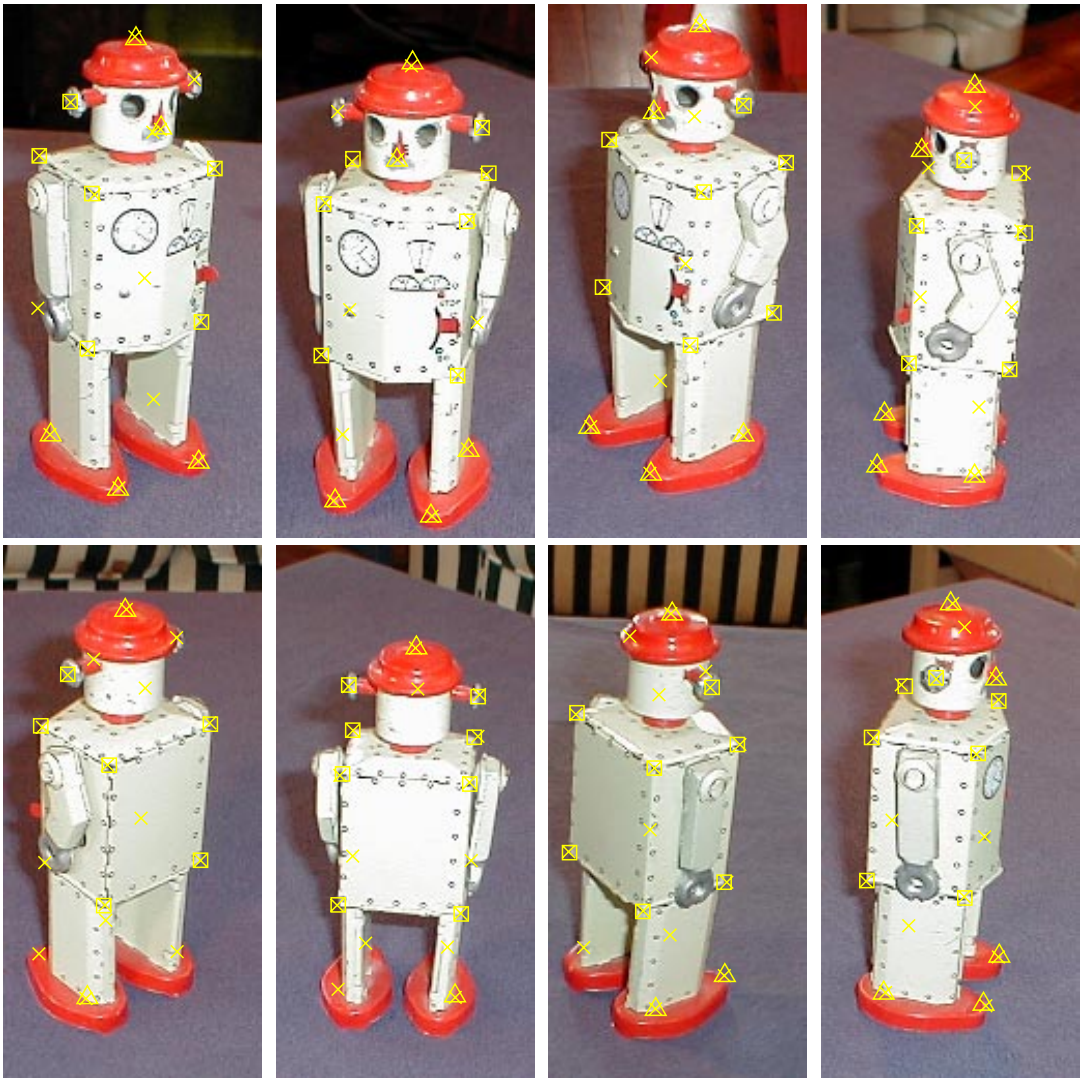


Figure 10.4: Input Images. The last predicted location is marked with an asterisk. Measurements are shown as squares and triangles.

appearance attribute are shown. Triangles and squares respectively represent the measurements with and without the color red.

The effect of incorporating appearance is clearly seen in the E-step. The course of the EM algorithm is illustrated as before, with the soft correspondence in Figure 10.2 and the prediction paths in Figure 10.3. The perfect appearance model makes that there is no probability mass between red measurements and non-red-features, and vice versa. This causes a block-diagonal structure for the marginal probabilities in the early iterations of the algorithm.

10.1.2 Multiple Symbols

To fully appreciate the information added by appearance measurements, consider the wire toy example in Figure 10.5. For this example, location measurements on the beads were obtained manually. If we look closer at one image (Figure 10.6) it is clear that, if the beads are indistinguishable, there is considerable opportunity for confusion, as projections of the beads line up in many images. Furthermore, in every image there are some some beads that are occluded. To illustrate incorporating appearance, the 153 bead measurements n the 8 images were augmented with a symbolic attribute representing the color of the bead. In total, there are 6 red (squares), 5 yellow (triangles), 4 green (circles), 4 orange (upside down triangles), and 4 blue beads (diamonds). This information is given to the algorithm, i.e. we are in the “known partition” case, and the problem is expected to decompose into 5 smaller problems.

Figure 10.7 contrasts the results without (on the left) and with incorporating appearance (on the right). Note that the final correspondence in the former is incorrect, whereas the probability mass in the latter is constrained considerably by appearance. Even so, there is still potential for confusing like measurements. For example, in iteration 9 there is still considerable uncertainty about the yellow and blue beads, mostly so in image 4. Looking back at Figure 10.6 we see the cause: the yellow bead measurements almost overlap, whereas the blue beads projections in the lower right corner are quite close. However, this is quickly resolved by the EM algorithm by using measurements in the other images do disambiguate the situation, and the final correspondence with appearance is the correct one.

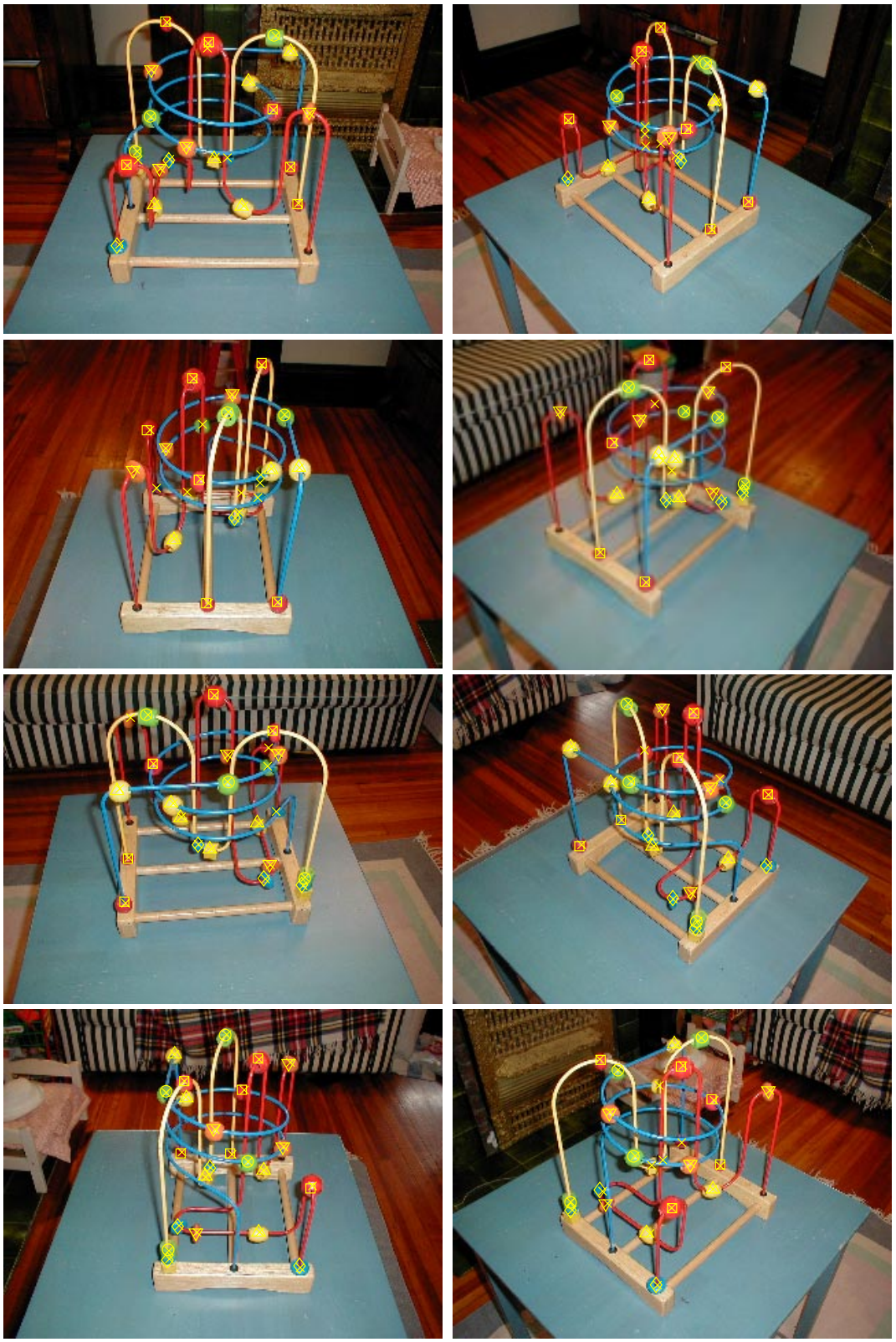


Figure 10.5: Eight input Images for wire toy example. The last predicted location is marked with an asterisk. Measurements are shown as various symbols.

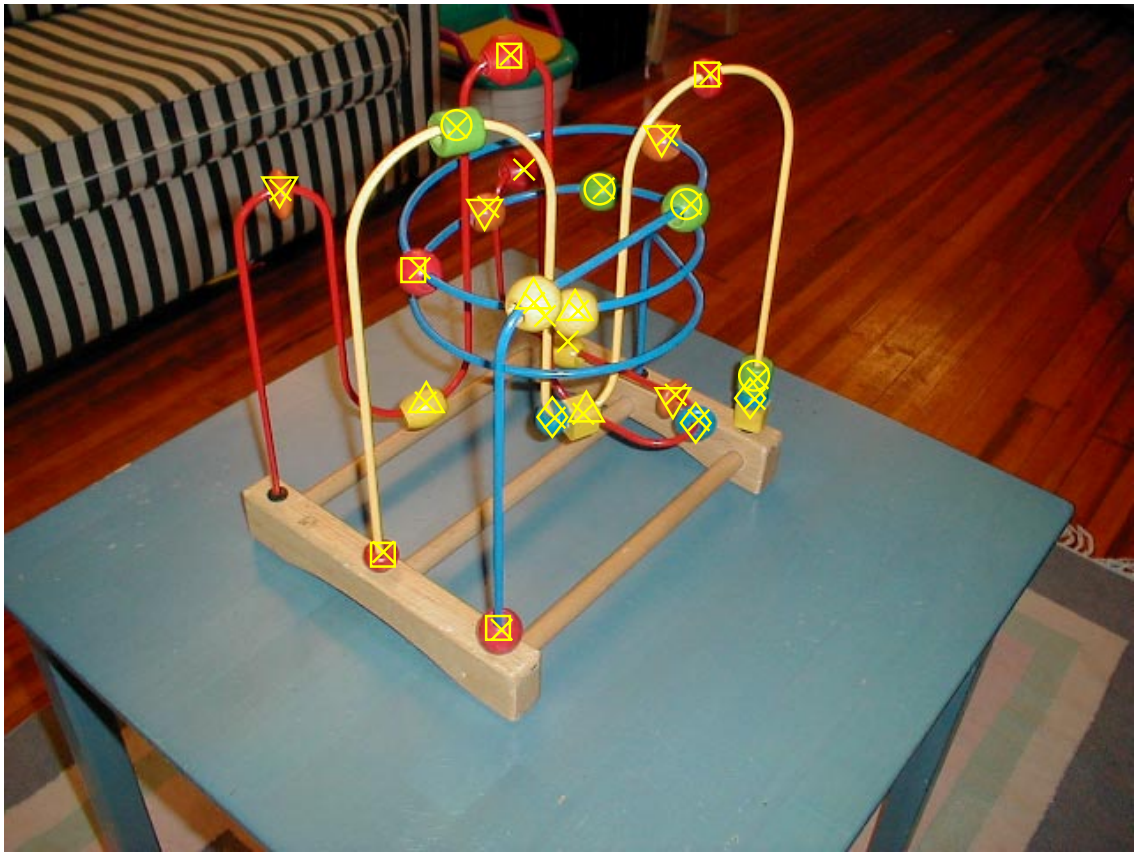
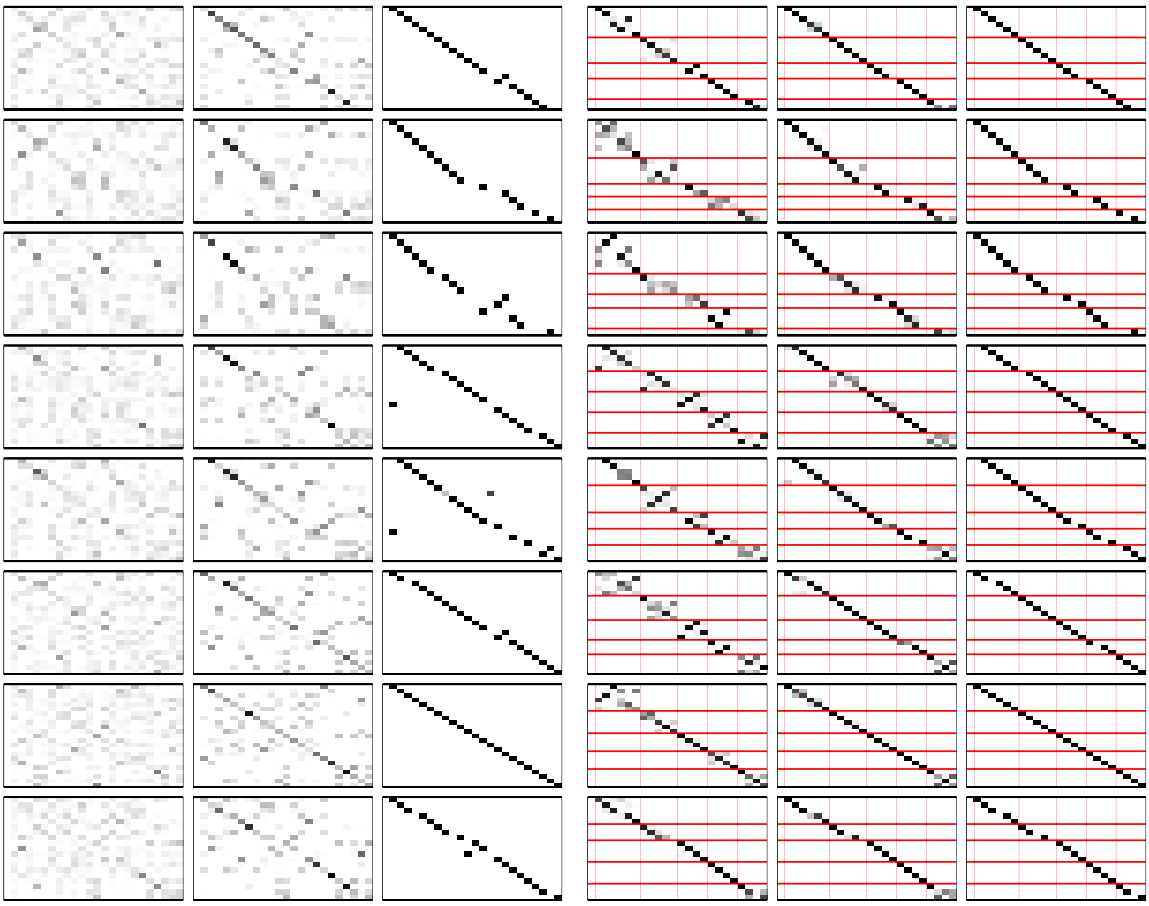
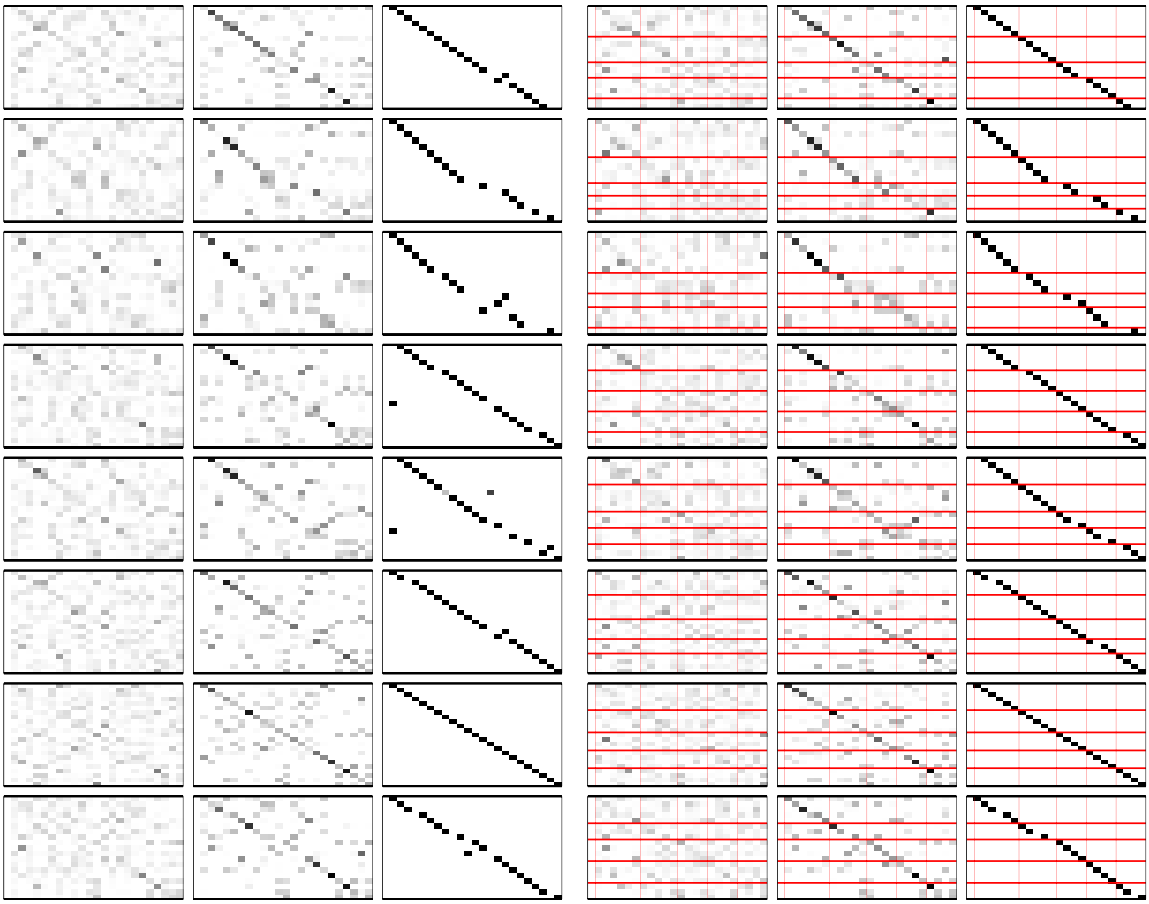


Figure 10.6: Large version of wire toy image 4, clearly showing the symbolic appearance measurements. There is one symbol per color.



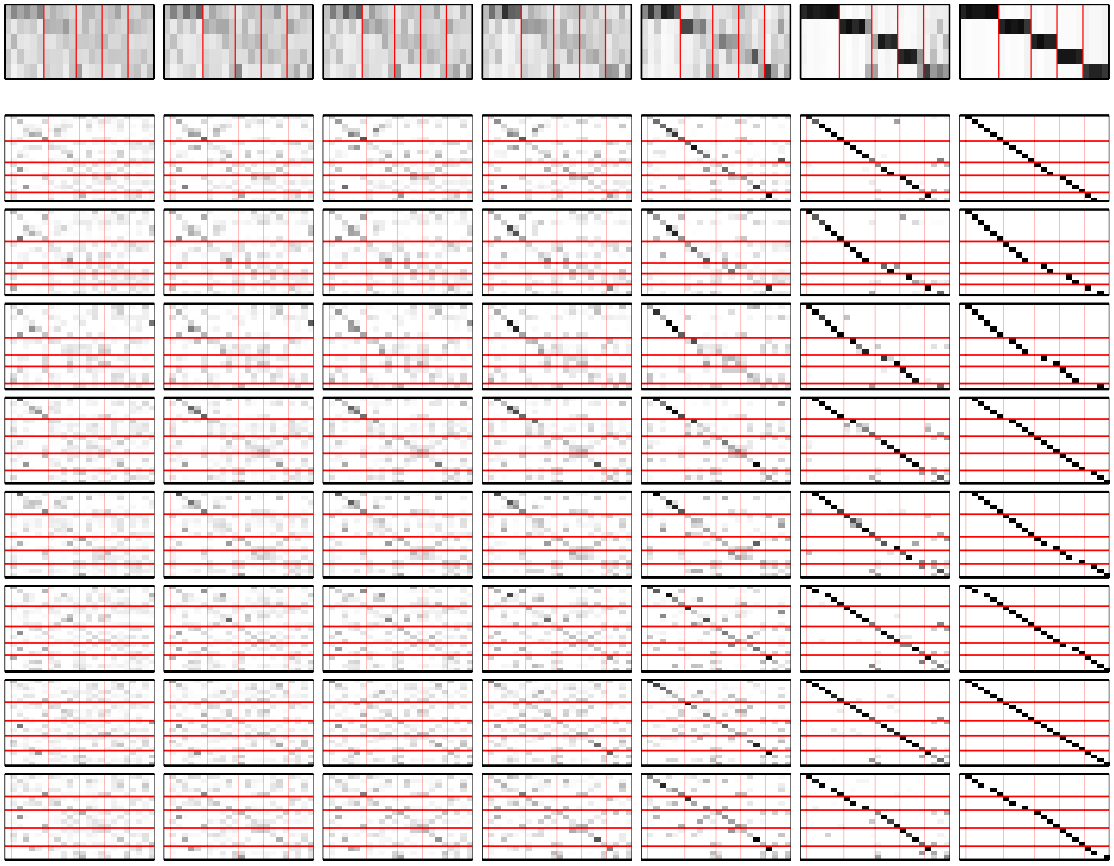
(a) it 1 (b) it 9 (c) it 25 (d) it 1 (e) it 9 (f) it 25

Figure 10.7: Comparing results without (on the left) and with incorporating appearance (on the right). The features and measurements are partitioned according to colors, in the order: red (6), yellow (5), green(4), orange(4), blue(4).



(a) it 1 (b) it 9 (c) it 25 (d) it 1 (e) it 9 (f) it 25

Figure 10.8: Comparing results without (on the left) and with incorporating appearance (on the right), with unknown partitions. The features and measurements are partitioned as before.



(a) it 1 (b) it 2 (c) it 3 (d) it 5 (e) it 9 (f) it 17 (g) it 25

Figure 10.9: Marginal probabilities and appearance statistics. Above each marginals plot, the 5 by 23 grayscale image shows for the posterior probability $P(\mathbf{y}_j = y|\mathbf{U}, \mathbf{A}, \boldsymbol{\Theta}^t)$ for each of the 23 features, where $y \in 1..5$. Thus, each column has 5 entries, one for each color, and the features are grouped the same way as before.

10.2 Unknown Partition Sizes

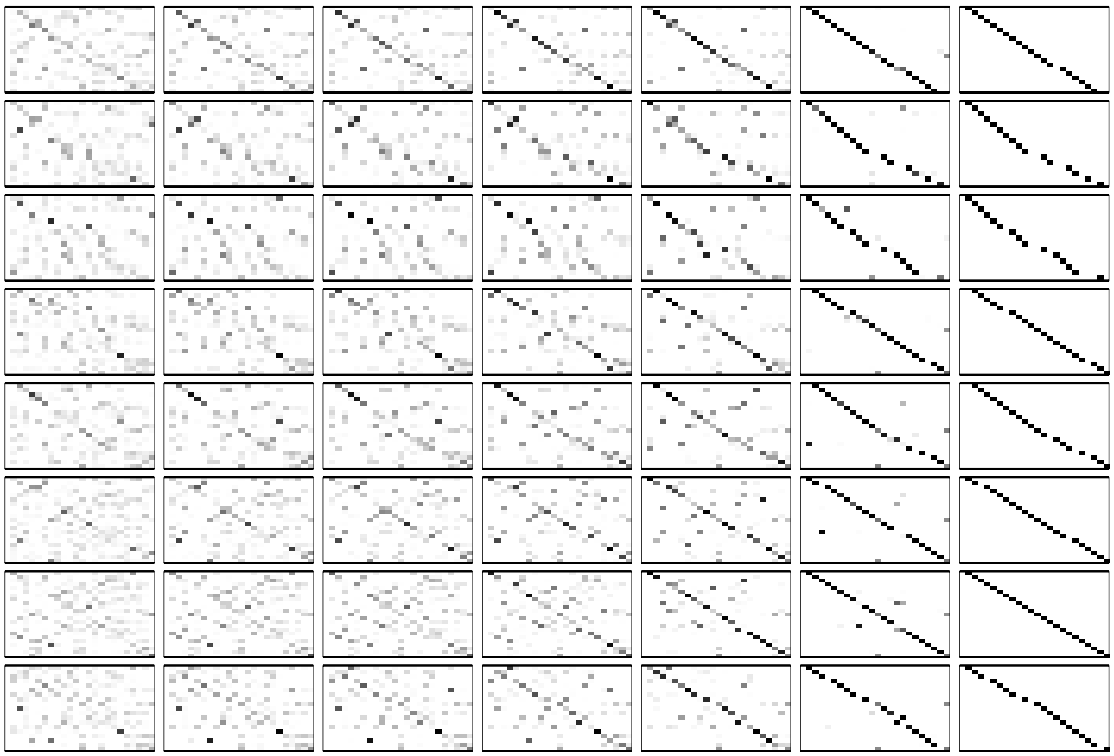
This section shows results on the same "wire toy" image sequence from Section 10.1, but now assuming the partition sizes are unknown. If this more general case, appearance can still be used to bias the sampling. As explained above, this has to be done using importance sampling. In the case of 5 symbolic features, a completely uninformative appearance measurement model would have a reliability $p = 0.2$, i.e. the appearance measurement is modeled as drawn at random from the 5 features. We cannot use the true reliability $p = 1.0$, as this will result in a useless sampler, unless we happen to stumble on a perfectly consistent joint assignment $\mathbf{J}$. Any value for p close to 1.0 will have the same effect: the importance sampler will have high variance, i.e. dominated by a few (or even one) very large importance weights.

With a relatively low appearance bias, the type partitions can be recovered even if unknown. Figure 10.8 shows the marginals obtained with an appearance bias of $p = 0.3$. This means, we *model* the probability of drawing the predicted appearance measurement is 0.3, whereas any other draw has a probability of $(1-0.3)/4 = 0.175$. Compared to the marginals without using appearance (which are again shown the right), there is not a lot of difference, but the appearance bias is enough to avoid the incorrect local minimum that was attained without appearance.

While the appearance parameters $\mathbf{y}_j$ are integrated out in the E-step, it is nevertheless instructive to plot the posterior probabilities $P(\mathbf{y}_j = y|\mathbf{U}, \mathbf{A}, \mathbf{\Theta}^t)$ for each possible color assignment to each of the features. This is done in Figure 10.9, where these posterior probabilities are represented as images, in the same way the marginals are. Note that a near-perfect red-yellow-green-orange-blue partition is recovered in the last iteration.

10.3 EM with a Simple Continuous Model

The approach from Section 9.5, i.e. re-estimating the hidden appearance parameters $\mathbf{Y}$ along with the structure and motion $\mathbf{\Theta}$, is demonstrated for the case of the simple continuous model from Section 9.2.4. In particular, the appearance parameters were taken to be 15×15 templates of predicted pixel values. The image sequence was again the wiretoy image from Figure 10.5, for which this model could be assumed to hold: the beads on the wires do look roughly the same no matter where they are viewed from. The EM algorithm was run for 25 iterations with linear annealing, and the marginal correspondence probabil-



(a) it 1 (b) it 2 (c) it 3 (d) it 5 (e) it 9 (f) it 17 (g) it 25

Figure 10.10: Marginal probabilities with continuous appearance model, where appearance is modeled by image templates.

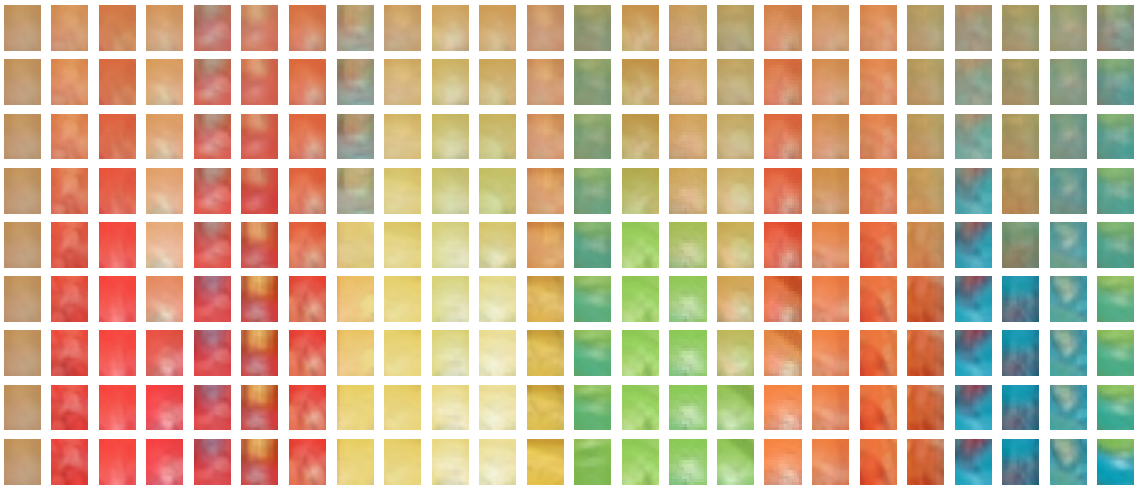


Figure 10.11: Estimated template means, respectively in iteration 1,2,3,5,9,13,17,21, and 25.

ities are shown for a subset of these iterations in Figure 10.10. As can be seen from the last panel, the correct correspondence was recovered. The estimated appearance templates change over time and are shown in Figure 10.11. Since the same ordering was used for the colors as before (red, yellow, greed, orange, and blue), we can see very well that the correct appearance is recovered towards the end of the algorithm.

Chapter 11

Discussion

11.0.1 Summary of Thesis

In this dissertation, I have shown that the Monte Carlo EM algorithm provides a practical way to accurately approximate the optimal solution of multi-view geometric estimation problems with unknown correspondence.

Mathematically, the MCEM approach combines several tools from applied probability and statistics: the expectation-maximization algorithm provides a tractable way to optimally estimate structure from motion with unknown correspondence, provided the marginal probabilities over the space of correspondences can be computed efficiently. The latter is done by approximating the distribution over correspondences at each iteration of EM by a Markov chain Monte Carlo sampler. An efficient sampler, specifically tuned to the correspondence problem, was developed for that purpose. Finally, a deterministic annealing strategy was used to avoid the local maxima problem that can otherwise hamper an EM based approach.

The new proposal strategies I proposed for efficient sampling of assignments bear an interesting relation to research in the field of computational complexity theory. The "chain flipping" proposal is related in terms of mechanism, if not description, to the Broder chain, an MCMC type method to generate (unweighted) assignments at random (Broder, 1986). However, our method is specifically geared towards sampling from *weighted* assignments, and uses the weights to bias proposals towards more likely assignments.

While initially derived under the assumptions of perfect visibility (i.e. all features visible in all images) in Chapters 4, 5, and 6, it was shown in Chapter 7 that the approach is easily extended to handle occlusion and clutter. Results with various degree of occlusion and

simulated clutter were shown in Chapter 8. However, if significant occlusion and clutter is present, any approach based solely on geometry is likely to diverge in many cases, unless strong priors on motion and or structure are imposed. Such a motion prior (the "arc" prior) was introduced in Chapter 8, and many other -application dependent- priors can be imagined. The combined results in Chapters 6 and 8 show that the Bayesian methodology, implemented through the MCEM algorithm, is capable of recovering structure and/or motion from measurement data that present significant challenges, as can be appreciated by looking at the datasets without viewing the original images.

Clearly though, geometry is not the only measurement information that can be derived from images: appearance information can significantly constrain the data-association problem. In Chapter 9 is is shown how appearance information can be incorporated into the geometric estimation process, and experimental results with use of appearance are shown in Chapter 10. It was shown that appearance can be viewed as a nuisance variable, just like correspondence, but that this presents a significant computational challenge. A much simpler approach is to regard appearance as one of the variables to be estimated, and it can be argued that this is indeed the sensible thing to do. It was shown that in that case, the MCEM approach can be straightforwardly extended by incorporating appearance as an unknown in the M-step, and having it constrain the data-association in the E-step.

11.0.2 Future Work

Despite the tools and techniques proposed in this dissertation for the problem of data-association, fully automatic structure from motion without correspondence remains a significant challenge. In particular, I have completely side-stepped the the important issue of feature selection, as all results were obtained on data sets where feature selection was done by hand. This allowed us to concentrate fully on the simultaneous geometric estimation and data-association problem, rather than having to solve the feature selection problem as well. Commonly used feature detectors are far from ideal, and the amount of spurious measurements and missed features makes application of the MCEM algorithm a non-trivial problem, especially in the case when no appearance information is used.

There are at least two possible approaches to push towards fully automatic structure and motion recovery. First, one could push on the feature selection side, i.e. try to extract only features that can be reliably detected across views, and are less prone to spurious measurements than, say, corner detectors. Second, one could concentrate on extracting appearance measurements from the images that are invariant to changes in viewpoint and

can be used to easily identify matches in other images. This is an area of intense research (Schmid and Mohr, 1997; Montesinos et al., 1998)(Tuytelaars and Van Gool, 2000; Tuytelaars and Van Gool, 2001).

The selection of better features is not addressed here, but for the latter, appearance-based approach it was shown in Chapter 9 that such appearance measurements can be easily incorporated in the MCEM approach. However, the problem is not the ability of MCEM to take appearance into account, but rather the fact that reliable appearance models are far from obvious, especially in structure from motion applications. Indeed, invariant appearance descriptors are in general not available under 3D viewing transformations (Schmid and Mohr, 1997). While I have shown results (in Chapter 10) with image sequences in which appearance was relatively stable from view to view, even with large displacements, this is seldomly the case if image sets are taken under more realistic circumstances. For instance, if a detected feature sits on an occlusion boundary (a frequent occurrence), the background can change dramatically depending on from which side it is viewed. Note that this issue is not avoided by the use of other algorithms, e.g. RANSAC based methods, which face exactly the same problem. In fact, the usefulness of RANSAC-based estimation of multiview constraints is severely limited by the implicit assumption of that viewpoint changes will be small, as the initial seeding of correspondences will fail otherwise. This is primarily due to the loss of appearance consistency over large displacements.

One other assumption made in this dissertation is unlikely to be satisfied in practice, namely the assumption that the number of features n is known, *a priori*. This is a valid assumption if there is no occlusion or clutter, as in that case the number of features can be obtained simply by counting the number of measurements in any given image. However, in the presence of occlusion and or clutter, we have a *model selection* problem: what is the number of features that best explains the data ? There are a number of possible solutions that are currently the focus of ongoing work. First, EM can be used in conjunction with a criterion such as the Bayesian Information Criterion (BIC), in order to obtain a MAP estimate for the number of features. In effect, the BIC provides a Bayesian prior on the number of features. A second approach would be to take structure into the E-step, i.e. integrate out the structure of unknown dimension as a nuisance variable, analogous to correspondence. Instead of sampling over correspondences only, we would then also sample over the space of possible structures, a union of spaces of different dimensions (one space for each value of n). We then obtain a MAP estimate for motion $\mathbf{M}$ only, and, if desired, an associated *sample* over the structure $\mathbf{X}$. A third approach is to abandon a point estimate for motion altogether, and simply sample over the joint space of structure and motion, integrating out

correspondence. This would be the purely Bayesian approach.

Finally, it is important to note that the MCEM approach is not limited to point features, or, in general, to feature-based methods. There is no reason why the approach could not be applied, in principle, to the pixel values in the images themselves. In this respect the work of Yuille or Roy (Yuille et al., 1991; Roy and Cox, 1998) can serve as a guide. In those papers, multiview correspondence methods were applied working directly with the individual pixels. However, their approach was limited to stereo (known motion) and the algorithms they used do not guarantee optimality. In addition, they isolate one single, "best" multiview correspondence, the shortcomings of which were one of the motivations behind the EM-based framework presented here. It is of considerable interest to see whether the MCEM approach to data-association can be applied in a computationally efficient manner to structure and motion recovery directly from pixel values, i.e. truly using *all* the information available in the images.

Appendix A

Bundle Adjustment for Point Features

This appendix describes the bundle-adjustment method for point features that was used to generate all results.

A.1 Bundle Adjustment

Recall from Section 2.3 that to find the maximum likelihood (ML) solution $\mathbf{\Theta}^*$ for structure and motion we need to minimize the following objective function (equation 2.3 on page 28):

$$\log L(\mathbf{\Theta}; \mathbf{U}, \mathbf{J}) \propto -\sum_{i=1}^{m} \sum_{k=1}^{K_i} \|\mathbf{u}_{ik} - \mathbf{h}(\mathbf{m}_i, \mathbf{x}_{\mathbf{j}_{ik}})\|^2 \tag{A.1}$$

This can be written using vector notation by collecting all the measurements in a column vector $\mathbf{U}$, and introducing a vector-function $\overline{\mathbf{h}}(\mathbf{\Theta}, \mathbf{J})$ that predicts the measurements given structure and motion $\mathbf{\Theta}$ and a correspondence vector $\mathbf{J}$. If we assume that there are K 2D image measurements, $\overline{\mathbf{h}}$ is $2K$-dimensional. We can then write:

$$-\log L(\mathbf{\Theta}; \mathbf{U}, \mathbf{J}) = \|\mathbf{U} - \overline{\mathbf{h}}(\mathbf{\Theta}, \mathbf{J})\|^2 \tag{A.2}$$

To find the ML solution, we need to minimize (A.2). In order to do this, we must in general use a non-linear minimization method, as $\overline{\mathbf{h}}(.)$ involves an image projection. One such

method is *Gauss-Newton* non-linear minimization, which starting from an initial guess $\mathbf{\Theta}^0$ for structure and motion, and iterates over

$$\mathbf{\Theta}^{t+1} = \mathbf{\Theta}^t + \left(\mathbf{H}_t^T \mathbf{H}_t\right)^{-1} \mathbf{H}_t^T (\mathbf{U} - \overline{\mathbf{h}}(\mathbf{\Theta}^t))$$

in practice implemented by solving the system of *normal equations*:

$$\mathbf{H}_t^T \mathbf{H}_t (\mathbf{\Theta}^{t+1} - \mathbf{\Theta}^t) = \mathbf{H}_t^T (\mathbf{U} - \overline{\mathbf{h}}(\mathbf{\Theta}^t)) \tag{A.3}$$

In this expression the matrix $\mathbf{H}_t$ is defined as the Jacobian of $\overline{\mathbf{h}}(.)$ evaluated at $\mathbf{\Theta}^t$

$$\mathbf{H}_t = \left.\frac{\partial \overline{\mathbf{h}}(\mathbf{\Theta})}{\partial \mathbf{\Theta}}\right|_{\mathbf{\Theta}^t}$$

$\mathbf{H}_t$ has dimension $2K \times N$, where N is the number of unknowns, i.e. the dimension of $\mathbf{\Theta}$. For example, for 6 degree of freedom cameras and 3D points we have $N = 6m + 3n$, with m the number of camera views and n the number of points, and $\mathbf{H}_t$ has dimension $2K \times (6m + 3n)$.

Since $\overline{\mathbf{h}}(.)$ can be very non-linear, straight Gauss-Newton iterations are usually replaced by Levenberg-Marquardt iterations. This method automatically switches to gradient descent when Gauss-Newton diverges, by making the Hessian diagonally dominant. In particular, the diagonal elements $\mathbf{Q}_{kk}$ of $\mathbf{Q} \stackrel{\Delta}{=} \mathbf{H}^T\mathbf{H}$ are replaced by $\mathbf{Q}_{kk}(1+\lambda)$, where λ is a parameter that is automatically adjusted during the course of the algorithm.

A.2 Sparse Solver

The Hessian $\mathbf{H}_t^T\mathbf{H}_t$ is a block-matrix consisting of $(m+n)^2$ sub-matrices, which is quite large if many points and/or camera positions are being considered. Inverting the Hessian is the main cost in iterating (A.3). One way to avoid this inversion is to alternate between structure $\mathbf{X}$ and motion $\mathbf{M}$, keeping one constant while solving for the other. However, this can lead to slow convergence, as structure and motion are not being considered simultaneously.

Hartley (Hartley, 1994) showed that, by making use of the special block structure of $\mathbf{H}$, the exact solution of (A.3) can be found efficiently. Below I present a slightly different treatment which is easier to implement, as we rely on sparse matrix multiplication to perform the bookkeeping for us.

To start with, $\mathbf{H}$ is first written as composed of a $K \times m$ block-matrix $\mathbf{F}$ and a $K \times n$ block-matrix $\mathbf{G}$

$$\mathbf{H} = \begin{bmatrix} \mathbf{F} & \mathbf{G} \end{bmatrix} = \begin{bmatrix} \frac{\partial \overline{\mathbf{h}}(\mathbf{M},\mathbf{X})}{\partial \mathbf{M}} & \frac{\partial \overline{\mathbf{h}}(\mathbf{M},\mathbf{X})}{\partial \mathbf{X}} \end{bmatrix}$$

where $\mathbf{F}$ and $\mathbf{G}$ are the Jacobians of $\overline{\mathbf{h}}(.)$ with respect to motion $\mathbf{M}$ and structure $\mathbf{X}$, respectively. Both $\mathbf{F}$ and $\mathbf{G}$ are sparse matrices, as an image measurement $\mathbf{u}_{ij}$ is only affected by a change in camera pose $\mathbf{m}_i$ and a change in feature position $\mathbf{x}_j$. All sub-matrices corresponding to other combinations will contain only zeros. The Hessian $\mathbf{H}^T\mathbf{H}$ then becomes

$$\mathbf{H}^T\mathbf{H} = \begin{bmatrix} \mathbf{F}^T\mathbf{F} & \mathbf{F}^T\mathbf{G} \\ \mathbf{G}^T\mathbf{F} & \mathbf{G}^T\mathbf{G} \end{bmatrix} \triangleq \begin{bmatrix} \mathbf{U} & \mathbf{W} \\ \mathbf{W}^T & \mathbf{V} \end{bmatrix} \tag{A.4}$$

The sub-matrices $\mathbf{U}$, $\mathbf{V}$, and $\mathbf{W}$ are easily and efficiently computed using sparse matrix multiplication. As a result of the special structure of $\mathbf{F}$ and $\mathbf{G}$, both $\mathbf{U}$ and $\mathbf{V}$ are block-diagonal matrices, whereas $\mathbf{W}$ is in general not sparse. The system (A.3) can now be written as

$$\begin{bmatrix} \mathbf{U} & \mathbf{W} \\ \mathbf{W}^T & \mathbf{V} \end{bmatrix} \delta\mathbf{\Theta} = \begin{bmatrix} \mathbf{F}^T \\ \mathbf{G}^T \end{bmatrix} \mathbf{e} \tag{A.5}$$

where $\delta\mathbf{\Theta} \triangleq (\mathbf{\Theta}^{t+1} - \mathbf{\Theta}^t)$, $\mathbf{e} \triangleq (\mathbf{U} - \overline{\mathbf{h}}(\mathbf{\Theta}^t))$. By performing one step of Gaussian elimination, we transform the Hessian into a lower-triangular block matrix:

$$\begin{bmatrix} \mathbf{U} - \mathbf{W}\mathbf{V}^{-1}\mathbf{W}^T & \mathbf{0} \\ \mathbf{W}^T & \mathbf{V} \end{bmatrix} \delta\mathbf{\Theta} = \begin{bmatrix} \mathbf{F}^T - \mathbf{W}\mathbf{V}^{-1}\mathbf{G}^T \\ \mathbf{G}^T \end{bmatrix} \mathbf{e} \tag{A.6}$$

We now find the *motion update* $\delta\mathbf{M}$ by solving only the top half of (A.6):

$$(\mathbf{U} - \mathbf{W}\mathbf{V}^{-1}\mathbf{W}^T)\delta\mathbf{M} = (\mathbf{F}^T - \mathbf{W}\mathbf{V}^{-1}\mathbf{G}^T)\mathbf{e} \tag{A.7}$$

Once $\delta\mathbf{M}$ is found, it is substituted in the bottom half, which yields an expression for the *structure update* $\delta\mathbf{X}$:

$$\delta\mathbf{X} = \mathbf{V}^{-1}(\mathbf{G}^T\mathbf{e} - \mathbf{W}^T\delta\mathbf{M}) \tag{A.8}$$

Both update equations (A.7) and (A.8) involve $\mathbf{V}^{-1}$, which can be computed in $O(n)$ time (with n the number structure elements) due to the block-diagonal structure of $\mathbf{V}$.

A.3 Point Features

Below we derive expressions for $\mathbf{F}$ and $\mathbf{G}$ in the case that camera rotation is parameterized by incremental rotation angles ω_x, ω_y and ω_z, with respect to a base rotation $\mathbf{R}_i^{base}$. In the following we treat the case where each feature is seen in each image, but this is easily generalized to extended sequences.

Parameterization

To be precise, the cameras are parameterized as

$$\mathbf{m}_i = \left(\mathbf{t}_i, \mathbf{R}_i^{base}, \omega_i = \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} \right), i \in \{1..m\}$$

where ω_i is an angular velocity vector that specifies an incremental rotation with respect to $\mathbf{R}_i^{base}$, as will be explained below. We assume the camera focal length f, the aspect ratio a, and the principal point (u_0, v_0) to be known, and skew s to be zero, although these assumptions will be relaxed later.

The structure is parameterized as

$$\mathbf{x}_j = \begin{bmatrix} x \\ y \\ z \end{bmatrix}, j \in \{1..m\}$$

Note that we dropped the i and j subscripts of the scalars to avoid notation clutter. With the parameterization above, the 2×1 vector-valued measurement function $\mathbf{h}(\mathbf{m}_i, \mathbf{x}_j)(.)$ that predicts the measurement $\mathbf{u}_{ij} = (u_{ij}, v_{ij})$ is written as

$$\mathbf{h}(\mathbf{m}_i, \mathbf{x}_j) = \frac{1}{z'} \begin{bmatrix} fx' + u_0 z' \\ afy' + v_0 z' \end{bmatrix} = \begin{bmatrix} u_0 + \frac{fx'}{z'} \\ v_0 + \frac{afy'}{z'} \end{bmatrix} \tag{A.9}$$

where we define $\mathbf{x}_j^i$ as the coordinates of the point $\mathbf{x}_j$ expressed in the camera coordinate frame i:

$$\mathbf{x}_j^i \stackrel{\Delta}{=} \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \mathbf{R}_i(\mathbf{x}_j - \mathbf{t}_i) \tag{A.10}$$

In the expression above, the rotation matrix $\mathbf{R}_i$ is the product of the incremental rotation matrix $\mathbf{\Delta R}(\omega_i)$ and the base rotation $\mathbf{R}_i^{base}$:

$$\mathbf{R}_i = \mathbf{\Delta R}(\omega_i)\mathbf{R}_i^{base}$$

The incremental rotation matrix $\mathbf{\Delta R}(\omega_i)$ is given by Rodriguez's formula (Faugeras, 1993):

$$\mathbf{\Delta R}(\omega_i) = I + \frac{\sin\theta}{\theta} J(\omega_i) + \frac{1 - \cos\theta}{\theta^2} J(\omega_i)^2 \tag{A.11}$$

where $\theta = \|\omega_i\|$, and $J(\omega)$ is the skew symmetric operator $J : \mathbb{R}^3 \rightarrow \mathcal{SO}(3)$:

$$J(\omega) = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix}$$

The base rotation is updated after each iteration in the optimization process, using (A.11), so that at the working point $\mathbf{\Delta R}(\omega_i)$ is always equal to the identity matrix, and around this point the incremental rotation angles are small. This avoids the singularities normally associated with an Euler angle parameterization (Hartley, 1994; Shum and Szeliski, 2000).

Change with respect to an arbitrary parameter

The partial derivative of (A.9) with respect to an arbitrary parameter q (other than the camera intrinsics) can be found by applying the chain rule (Lowe, 1991):

$$\frac{\partial \mathbf{h}(\mathbf{m}_i, \mathbf{x}_j)}{\partial q} = \frac{f}{(z')^2} \begin{bmatrix} z' & 0 & -x' \\ 0 & az' & -ay' \end{bmatrix} \frac{\partial}{\partial q} \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = A(\mathbf{x}_j^i) \frac{\partial \mathbf{x}_j^i}{\partial q} \tag{A.12}$$

where we define $A(\mathbf{x}_j^i)$ to be the 2×3 matrix that is shared by all the derivatives. Below we will specialize this for the $6m$ unknown camera parameters and the $3n$ structure parameters, respectively.

Change in Camera Parameters

The camera parameters we are optimizing for are the translation $\mathbf{t}_i$ and the incremental rotation ω_i, yielding 6 unknowns per camera. The matrix $\mathbf{F}$, the partial derivative of $\overline{\mathbf{h}}$ with respect to the camera parameters, is a block matrix with $K \times m$ sub-blocks $\mathbf{F}_{ki}$, each of size 2×6, where K is the number of measurements (e.g. $K = mn$ if there is no occlusion or clutter). However, there are only K non-zero blocks, as measurement $\mathbf{u}_{ij}$ is only affected by a change in the camera parameters $\mathbf{m}_i$. Using (A.12) we have the following expression for each of these K nonzero blocks:

$$\mathbf{F}_{(ij)i} = \left. \frac{\partial \mathbf{h}(\mathbf{m}_i, \mathbf{x}_j)}{\partial (\mathbf{t}_i, \omega_i)} \right|_{\mathbf{m}_i, \mathbf{x}_j} = \left. A(\mathbf{x}_j^i) \frac{\partial \mathbf{x}_j^i}{\partial (\mathbf{t}_i, \omega_i)} \right|_{\mathbf{m}_i, \mathbf{x}_j}$$

The partial derivative of $\mathbf{x}_j^i$ with respect to translation $\mathbf{t}_i$ is simply

$$\frac{\partial \mathbf{x}_j^i}{\partial \mathbf{t}_i} = \frac{\partial \left(\mathbf{R}_i(\mathbf{x}_j - \mathbf{t}_i)\right)}{\partial \mathbf{t}_i} = -\mathbf{R}_i = -\mathbf{R}_i^{base} \tag{A.13}$$

and is independent of the feature point $\mathbf{x}_j$. The last equality follows because at the linearization point $\omega_x = \omega_y = \omega_z = 0$, and $\mathbf{R}_i = \mathbf{R}_i^{base}$.

An incremental rotation is slightly more complicated. Note that, for small ω_i, $\mathbf{\Delta R}(\omega_i)$ can be approximated by

$$\mathbf{\Delta R}(\omega_i) \approx I + J(\omega_i)$$

From this, we see that the effect of the incremental rotation on $\mathbf{x}_j^i$ can be written in terms of a cross product:

$$\mathbf{\Delta R}(\omega_i)\mathbf{x}_j^i \approx \mathbf{x}_j^i + \omega_i \times \mathbf{x}_j^i$$

as $J(\omega)\mathbf{p} = \omega \times \mathbf{p}$ for any arbitrary ω and $\mathbf{p}$. Taking the partial derivative with respect to ω_i yields

$$\frac{\partial \mathbf{\Delta R}(\omega_i)\mathbf{x}_j^i}{\partial \omega_i} = \frac{\partial(\omega_i \times \mathbf{x}_j^i)}{\partial \omega_i} = -J(\mathbf{x}_j^i)$$

The final expression for the 2×6 matrix $\mathbf{F}_{(ij)i}$ is

$$\mathbf{F}_{(ij)i} = -A(\mathbf{x}_j^i) \begin{bmatrix} \mathbf{R}_i^{base} & J(\mathbf{x}_j^i) \end{bmatrix} \tag{A.14}$$

Change in Structure Parameters

We have three unknown structure parameters x, y, and z for each point $\mathbf{x}_j$. The matrix $\mathbf{G}$, the partial derivative of $\overline{\mathbf{h}}$ with respect to the structure parameters, is a block matrix with $K \times n$ sub-blocks $\mathbf{G}_{kj}$, each of size 2×3. However, there are only K non-zero blocks, as measurement $\mathbf{u}_{ij}$ is only affected by a change in the feature point $\mathbf{x}_j$. Again each non-zero block is found using (A.12):

$$\mathbf{G}_{(ij)j} = \left.\frac{\partial \mathbf{h}(\mathbf{m}_i, \mathbf{x}_j)}{\partial \mathbf{x}_j}\right|_{\mathbf{m}_i, \mathbf{x}_j} = \left. A(\mathbf{x}_j^i)\frac{\partial \mathbf{x}_j^i}{\partial \mathbf{x}_j}\right|_{\mathbf{m}_i, \mathbf{x}_j}$$

The 3×3 partial derivative of $\mathbf{x}_j^i$ with respect to the feature point $\mathbf{x}_j$ is

$$\frac{\partial \mathbf{x}_j^i}{\partial \mathbf{x}_j} = \frac{\partial \mathbf{R}_i(\mathbf{x}_j - \mathbf{t}_i)}{\partial \mathbf{x}_j} = \mathbf{R}_i = \mathbf{R}_i^{base} \tag{A.15}$$

The final expression for the 2×3 matrix $\mathbf{G}_{(ij)j}$ is then

$$\mathbf{G}_{(ij)j} = A(\mathbf{x}_j^i)\mathbf{R}_i^{base} \tag{A.16}$$

Note that $\mathbf{G}_{(ij)j}$ is equal (up to a sign) to the first 3 columns of $\mathbf{F}_{(ij)i}$. In an implementation this can be used to speed up the calculation of $\mathbf{F}$ and $\mathbf{G}$.

Varying Intrinsic Parameters

If the intrinsic parameters are allowed to vary between cameras, they can simply be added to the unknowns for each camera. The corresponding partial derivatives then need to be appended to the non-zero blocks of $\mathbf{F}$. In the case of varying focal length f, aspect ratio a, and principal point (u_0, v_0) the corresponding partial derivatives are

$$\frac{\partial \mathbf{h}(\mathbf{m}_i, \mathbf{x}_j)}{\partial [f\ a\ u_0\ v_0]^T} = \frac{1}{z'} \begin{bmatrix} x' & 0 & z' & 0 \\ ay' & fy' & 0 & z' \end{bmatrix}$$

Non-zero Skew

The case for non-zero skew can easily be accommodated, but leads to slightly more complicated expressions. In the case the skew can vary between cameras, this is is handled in a similar fashion as varying focal length etc.

A.4 Orthographic Case

In the orthographic case, the expressions of $\mathbf{F}$ and $\mathbf{G}$ are bilinear in the parameters (Morris and Kanade, 1998). We have:

$$\mathbf{h}(\mathbf{m}_i, \mathbf{x}_j) = \begin{bmatrix} \hat{\mathbf{i}}_i^T \mathbf{x}_j \\ \hat{\mathbf{j}}_i^T \mathbf{x}_j \end{bmatrix}$$

and

$$\mathbf{F}_{(ij)i} = \left. \frac{\partial \mathbf{h}(\mathbf{m}_i, \mathbf{x}_j)}{\partial (\hat{\mathbf{i}}_i, \hat{\mathbf{j}}_i)} \right|_{\mathbf{m}_i, \mathbf{x}_j} = \begin{bmatrix} \mathbf{x}_j^T & 0 \\ 0 & \mathbf{x}_j^T \end{bmatrix}$$

$$\mathbf{G}_{(ij)j} = \left. \frac{\partial \mathbf{h}(\mathbf{m}_i, \mathbf{x}_j)}{\partial \mathbf{x}_j} \right|_{\mathbf{m}_i, \mathbf{x}_j} = \begin{bmatrix} \hat{\mathbf{i}}_i^T \\ \hat{\mathbf{j}}_i^T \end{bmatrix}$$

A.5 Imposing Inner Constraints

Because of the position and scale ambiguity inherent in the SFM problem, the Hessian $\mathbf{H}^T\mathbf{H}$ from equation (A.4) will be rank-deficient. In the 3D case, we need to impose 7 constraints to remove this rank deficiency. The *inner constraints* are a set of constraints on the structure update $\Delta\mathbf{X}$ that are optimal in the sense that they minimize the trace of the resulting covariance matrix of $\mathbf{\Theta}^*$ (Cooper and Robson, 1996). They are (from (Cooper and Robson, 1996), p. 41-42) the three positional constraints,

$$\sum \delta x_j = \sum \delta y_j = \sum \delta z_j = 0$$

three rotational constraints,

$$\sum [z_j \delta y_j - y_j \delta z_j] = \sum [-z_j \delta x_j + x_j \delta z_j] = \sum [y_j \delta x_j - x_j \delta y_j] = 0$$

and one scale constraint

$$\sum [x_j \delta x_j + y_j \delta y_j + z_j \delta z_j] = 0$$

where x_j, y_j, and z_j are the coordinates of the 3D point $\mathbf{x}_j$, and $j \in \{1..n\}$. It is convenient to write these constraints in the form

$$\mathbf{C}\delta\mathbf{X} = 0 \tag{A.17}$$

The normal equations (A.5) can now be augmented with the constraints (A.17) as follows:

$$\begin{bmatrix} \mathbf{U} & \mathbf{W} & \mathbf{0} \\ \mathbf{W}^T & \mathbf{V} & \mathbf{C}^T \\ \mathbf{0} & \mathbf{C} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \delta\mathbf{M} \\ \delta\mathbf{X} \\ \lambda \end{bmatrix} = \begin{bmatrix} \mathbf{F}^T \\ \mathbf{G}^T \\ \mathbf{0} \end{bmatrix} \mathbf{e}$$

and after a similar elimination process as the one that lead to (A.6), we get

$$\begin{bmatrix} \mathbf{U} - \mathbf{W}\mathbf{V}^{-1}\mathbf{W}^T & -\mathbf{W}\mathbf{V}^{-1}\mathbf{C}^T \\ -\mathbf{C}\mathbf{V}^{-1}\mathbf{W}^T & -\mathbf{C}\mathbf{V}^{-1}\mathbf{C}^T \end{bmatrix} \begin{bmatrix} \delta\mathbf{M} \\ \lambda \end{bmatrix} = \begin{bmatrix} \mathbf{F}^T - \mathbf{W}\mathbf{V}^{-1}\mathbf{G} \\ -\mathbf{C}\mathbf{V}^{-1}\mathbf{G} \end{bmatrix} \mathbf{e} \tag{A.18}$$

and

$$\delta\mathbf{X} = \mathbf{V}^{-1}(\mathbf{G}\mathbf{e} - \mathbf{W}^T\delta\mathbf{M} - \mathbf{C}^T\lambda) \tag{A.19}$$

A.6 Automatic Differentiation

The sparse solver techniques above are very efficient for the classical 3D point feature setup. However, the analytic derivatives are complex to derive in many other cases, and the process of deriving them is error-prone and time-consuming. In order to lower the entry-barrier to introducing interesting priors or camera models, I have implemented an automatic differentiation toolbox in MATLAB.

Automatic differentiation (Griewank, 1989), abbreviated AD, is neither symbolic differentiation nor numerical differentiation by a finite-difference approximation. Instead, to quote Griewank, "AD simply implements the chain rule in a suitable fashion". When the value of a derivative is needed, AD computes the values of the arguments and their derivatives (recursively), evaluates the function at the arguments, and executes the correct multiplications and additions to implement the chain rule. In contrast to symbolic differentiation, an analytic expression for the derivative is never computed or needed. And, unlike numerical differentiation, the value of the derivative is exact and free of numerical instabilities. In addition, if implemented in a certain way, the computation cost is never more than 5 times the cost of evaluating the function value itself (typically more like 1.5 times the cost).

I have implemented AD by creating a small functional language within MATLAB, where common vector valued functions are adjoined with functions implementing their derivatives. The objective function for optimization problems can be composed from these primitive functions using let statements and various vector operators. When a derivative needs to be evaluated at a given value, the chain rule is applied recursively by a top-level interpreter that produces the numerical value of the derivative and the function value at the same time. The toolbox handles vector-valued functions and large Jacobians (for thousands of variables), and makes use of sparse matrix techniques to attain efficiency.

Appendix B

EM as Lower Bound Maximization

The expectation-maximization (EM) algorithm can be explained in many different ways (Dempster et al., 1977; McLauchlan and Murray, 1995; Tanner, 1996), one of the most insightful being in terms of lower bound maximization (Neal and Hinton, 1998; Minka, 1998). The goal is to maximize the posterior probability of the parameters $\mathbf{\Theta}$ given the data $\mathbf{U}$, or, equivalently, maximize the logarithm of the joint distribution:

$$\mathbf{\Theta}^* = \underset{\mathbf{\Theta}}{\text{argmax}} \ \log P(\mathbf{U}, \mathbf{\Theta}) = \underset{\mathbf{\Theta}}{\text{argmax}} \ \log \sum_{\mathbf{J} \in \mathcal{J}^n} P(\mathbf{U}, \mathbf{J}, \mathbf{\Theta}) \tag{B.1}$$

Here the variable $\mathbf{J}$ represents nuisance variables that cannot be easily integrated out, making an analytic approach to maximizing (B.1) intractable.

The idea behind EM is to start with a guess $\mathbf{\Theta}^t$ for the parameters $\mathbf{\Theta}$, compute an easily computed lower bound $B(\mathbf{\Theta}; \mathbf{\Theta}^t)$ to the function $\log P(\mathbf{\Theta}|\mathbf{U})$, and maximize that bound instead. If iterated, this procedure will converge to a local maximizer $\mathbf{\Theta}^*$ of the objective function, provided the bound improves at each iteration.

To motivate this, note that the key problem with maximizing (B.1) is that it involves the logarithm of a (big) sum, which is difficult to deal with. Fortunately, we can construct a tractable lower bound $B(\mathbf{\Theta}; \mathbf{\Theta}^t)$ that instead contains a sum of logarithms. To derive the bound, first trivially rewrite $\log P(\mathbf{U}, \mathbf{\Theta})$ as

$$\log P(\mathbf{U}, \mathbf{\Theta}) = \log \sum_{\mathbf{J} \in \mathcal{J}^n} P(\mathbf{U}, \mathbf{J}, \mathbf{\Theta}) = \log \sum_{\mathbf{J} \in \mathcal{J}^n} f^t(\mathbf{J}) \frac{P(\mathbf{U}, \mathbf{J}, \mathbf{\Theta})}{f^t(\mathbf{J})}$$

where $f^t(\mathbf{J})$ is an arbitrary probability distribution over the space $\mathcal{J}^n$ of hidden variables

$\mathbf{J}$. By Jensen's inequality, we have

$$B(\mathbf{\Theta};\mathbf{\Theta}^t) \stackrel{\Delta}{=} \sum_{\mathbf{J}\in\mathcal{J}^n} f^t(\mathbf{J}) \log \frac{P(\mathbf{U},\mathbf{J},\mathbf{\Theta})}{f^t(\mathbf{J})} <= \log \sum_{\mathbf{J}\in\mathcal{J}^n} f^t(\mathbf{J}) \frac{P(\mathbf{U},\mathbf{J},\mathbf{\Theta})}{f^t(\mathbf{J})}$$

Note that we have transformed a log of sums into a sum of logs, which was the prime motivation.

B.1 Finding an Optimal Bound

EM goes one step further and tries to find the *best* bound, defined as the bound $B(\mathbf{\Theta};\mathbf{\Theta}^t)$ that touches the objective function $\log P(\mathbf{U},\mathbf{\Theta})$ at the current guess $\mathbf{\Theta}^t$. Intuitively, finding the best bound at each iteration will guarantee that we obtain an improved estimate $\mathbf{\Theta}^{t+1}$ when we locally maximize the bound with respect to $\mathbf{\Theta}$. Since we know $B(\mathbf{\Theta};\mathbf{\Theta}^t)$ to be a lower bound, the optimal bound at $\mathbf{\Theta}^t$ can be found by maximizing

$$B(\mathbf{\Theta}^t;\mathbf{\Theta}^t) = \sum_{\mathbf{J}\in\mathcal{J}^n} f^t(\mathbf{J}) \log \frac{P(\mathbf{U},\mathbf{J},\mathbf{\Theta}^t)}{f^t(\mathbf{J})} \tag{B.2}$$

with respect to the distribution $f^t(\mathbf{J})$. Introducing a Lagrange multiplier λ to enforce the constraint $\sum_{\mathbf{J}\in\mathcal{J}^n} f^t(\mathbf{J}) = 1$, the objective becomes

$$G(f^t) = \lambda \left[1 - \sum_{\mathbf{J}\in\mathcal{J}^n} f^t(\mathbf{J})\right] + \sum_{\mathbf{J}\in\mathcal{J}^n} f^t(\mathbf{J}) \log P(\mathbf{U},\mathbf{J},\mathbf{\Theta}^t) - \sum_{\mathbf{J}\in\mathcal{J}^n} f^t(\mathbf{J}) \log f^t(\mathbf{J})$$

Taking the derivative

$$\frac{\partial G}{\partial f^t(\mathbf{J})} = -\lambda + \log P(\mathbf{U},\mathbf{J},\mathbf{\Theta}^t) - \log f^t(\mathbf{J}) - 1$$

and solving for $f^t(\mathbf{J})$ we obtain

$$f^t(\mathbf{J}) = \frac{P(\mathbf{U},\mathbf{J},\mathbf{\Theta}^t)}{\sum_{\mathbf{J}\in\mathcal{J}^n} P(\mathbf{U},\mathbf{J},\mathbf{\Theta}^t)} = P(\mathbf{J}|\mathbf{U},\mathbf{\Theta}^t)$$

yielding the following bound $B(\mathbf{\Theta};\mathbf{\Theta}^t)$:

$$B(\mathbf{\Theta};\mathbf{\Theta}^t) = \sum_{\mathbf{J}\in\mathcal{J}^n} P(\mathbf{J}|\mathbf{U},\mathbf{\Theta}^t) \log \frac{P(\mathbf{U},\mathbf{J},\mathbf{\Theta})}{P(\mathbf{J}|\mathbf{U},\mathbf{\Theta}^t)} \tag{B.3}$$

By examining the value of the resulting optimal bound at $\mathbf{\Theta}^t$ we see that it indeed touches the objective function:

$$B(\mathbf{\Theta}^t;\mathbf{\Theta}^t) = \sum_{\mathbf{J}\in\mathcal{J}^n} P(\mathbf{J}|\mathbf{U},\mathbf{\Theta}^t) \log \frac{P(\mathbf{U},\mathbf{J},\mathbf{\Theta}^t)}{P(\mathbf{J}|\mathbf{U},\mathbf{\Theta}^t)} = \log P(\mathbf{U},\mathbf{\Theta}^t)$$

B.2 Maximizing The Bound

To maximize $B(\mathbf{\Theta}; \mathbf{\Theta}^t)$ with respect to $\mathbf{\Theta}$, note that we can write (B.3) as

$$\begin{aligned} B(\mathbf{\Theta}; \mathbf{\Theta}^t) &\stackrel{\Delta}{=} \langle \log P(\mathbf{U}, \mathbf{J}, \mathbf{\Theta}) \rangle + \mathcal{H} \\ &= \langle \log P(\mathbf{U}, \mathbf{J} | \mathbf{\Theta}) \rangle + \log P(\mathbf{\Theta}) + \mathcal{H} \\ &= Q^t(\mathbf{\Theta}) + \log P(\mathbf{\Theta}) + \mathcal{H} \end{aligned}$$

where $\langle . \rangle$ denotes the expectation with respect to $f^t(\mathbf{J}) \stackrel{\Delta}{=} P(\mathbf{J}|\mathbf{U}, \mathbf{\Theta}^t)$, and

- $Q^t(\mathbf{\Theta})$ is the expected complete log-likelihood, defined as:

$$Q^t(\mathbf{\Theta}) \stackrel{\Delta}{=} \langle \log P(\mathbf{U}, \mathbf{J} | \mathbf{\Theta}) \rangle$$

- $P(\mathbf{\Theta})$ is the prior on the parameters $\mathbf{\Theta}$
- $\mathcal{H} \stackrel{\Delta}{=} - \langle \log f^t(\mathbf{J}) \rangle$ is the entropy of the distribution $f^t(\mathbf{J}) = P(\mathbf{J}|\mathbf{U}, \mathbf{\Theta}^t)$

Since $\mathcal{H}$ does not depend on $\mathbf{\Theta}$, we can maximize the bound with respect to $\mathbf{\Theta}$ using the first two terms only:

$$\mathbf{\Theta}^{t+1} = \underset{\mathbf{\Theta}}{\text{argmax}} \; B(\mathbf{\Theta}; \mathbf{\Theta}^t) = \underset{\mathbf{\Theta}}{\text{argmax}} \; \left[Q^t(\mathbf{\Theta}) + \log P(\mathbf{\Theta}) \right] \tag{B.4}$$

B.3 The EM Algorithm

At each iteration, the EM algorithm first finds an optimal lower bound $B(\mathbf{\Theta}; \mathbf{\Theta}^t)$ at the current guess $\mathbf{\Theta}^t$ (equation B.2), and then maximizes this bound to obtain an improved estimate $\mathbf{\Theta}^{t+1}$ (equation B.4). Because the bound is expressed as an expectation, the first step is called the "expectation-step" or E-step, whereas the second step is called the "maximization-step" or M-step. The EM algorithm can thus be conveniently summarized as:

- E-step: calculate $f^t(\mathbf{J}) \stackrel{\Delta}{=} P(\mathbf{J}|\mathbf{U}, \mathbf{\Theta}^t)$
- M-step: $\mathbf{\Theta}^{t+1} = \text{argmax}_{\mathbf{\Theta}} \left[Q^t(\mathbf{\Theta}) + \log P(\mathbf{\Theta}) \right]$

It is important to remember that $Q^t(\mathbf{\Theta})$ is calculated in the E-step by evaluating $f^t(\mathbf{J})$ using the *current guess* $\mathbf{\Theta}^t$ (hence the superscript t), whereas in the M-step we are optimizing $Q^t(\mathbf{\Theta})$ with respect to the *free variable* $\mathbf{\Theta}$ to obtain the new estimate $\mathbf{\Theta}^{t+1}$. It can be proven that the EM algorithm converges to a local maximum of $\log P(\mathbf{U}, \mathbf{\Theta})$, and thus equivalently maximizes the log-posterior $\log P(\mathbf{\Theta}|\mathbf{U})$ (Dempster et al., 1977; McLachlan and Krishnan, 1997).

B.4 Relation to the Expected Log-Posterior

Note that we have chosen to define $Q^t(\mathbf{\Theta})$ as the expected log-likelihood as in (Dempster et al., 1977; McLachlan and Krishnan, 1997), i.e.,

$$Q^t(\mathbf{\Theta}) \stackrel{\Delta}{=} \langle \log P(\mathbf{U}, \mathbf{J}|\mathbf{\Theta}) \rangle$$

An alternative route is to compute the expected log-posterior and maximize that in the M-step(Tanner, 1996):

$$\mathbf{\Theta}^{t+1} = \underset{\mathbf{\Theta}}{\text{argmax}} \ \langle \log P(\mathbf{\Theta}|\mathbf{U}, \mathbf{J}) \rangle \tag{B.5}$$

Applying Bayes law, we obtain

$$\langle \log P(\mathbf{\Theta}|\mathbf{U}, \mathbf{J}) \rangle = \langle \log P(\mathbf{U}, \mathbf{J}|\mathbf{\Theta}) + \log P(\mathbf{\Theta}) - \log P(\mathbf{U}, \mathbf{J}) \rangle$$

Here the second term does not depend on $\mathbf{J}$ and can be taken out of the expectation, and the last term does not depend on $\mathbf{\Theta}$. Hence, maximizing (B.5) with respect to $\mathbf{\Theta}$ is equivalent to (B.4):

$$\begin{aligned} \underset{\mathbf{\Theta}}{\text{argmax}} \ \langle \log P(\mathbf{\Theta}|\mathbf{U}, \mathbf{J}) \rangle &= \underset{\mathbf{\Theta}}{\text{argmax}} \ [\langle \log P(\mathbf{U}, \mathbf{J}|\mathbf{\Theta}) \rangle + \log P(\mathbf{\Theta})] \\ &= \underset{\mathbf{\Theta}}{\text{argmax}} \ \left[Q^t(\mathbf{\Theta}) + \log P(\mathbf{\Theta})\right] \end{aligned}$$

Appendix C

Virtual Measurements

In this appendix we prove the following theorem:

Theorem. *Assume that the measurements $\mathbf{u}_{ik}$ are normally distributed around their predicted value $\mathbf{h}(\mathbf{m}_i, \mathbf{x}_j)$, where $\mathbf{m}_i$ are the motion parameters associated with image i and $\mathbf{x}_j$ are the coordinates of feature j, i.e.*

$$P(\mathbf{u}_{ik}|\mathbf{m}_i, \mathbf{x}_j) = \frac{1}{\sqrt{|2\pi\mathbf{R}_{ik}|}} \exp\left[-\frac{1}{2}(\mathbf{u}_{ik} - \mathbf{h}_{ij})^T \mathbf{R}_{ik}^{-1}(\mathbf{u}_{ik} - \mathbf{h}_{ij})\right] \tag{C.1}$$

where we define $\mathbf{h}_{ij} \stackrel{\Delta}{=} \mathbf{h}(\mathbf{m}_i, \mathbf{x}_j)$ for notational convenience; In that case, the expected log-likelihood, given by equation 4.6 on page 56 and repeated here for convenience

$$Q^t(\mathbf{\Theta}) \equiv \sum_{i=1}^{m}\sum_{j=1}^{n}\sum_{k=1}^{K_i} f_{ijk}^t \log P(\mathbf{u}_{ik}|\mathbf{m}_i, \mathbf{x}_j) \tag{C.2}$$

is equivalent to the following virtual measurements formulation (equation 4.14 on page 59):

$$Q^t(\mathbf{\Theta}) \equiv -\frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{n}(\mathbf{v}_{ij}^t - \mathbf{h}_{ij})^T \mathbf{R}_{ij}^{-1}(\mathbf{v}_{ij}^t - \mathbf{h}_{ij}) \tag{C.3}$$

*Here the **virtual measurements** $\mathbf{v}_{ij}^t$ are defined as*

$$\mathbf{v}_{ij}^t \stackrel{\Delta}{=} \mathbf{R}_{ij}\sum_{k=1}^{K_i} f_{ijk}^t \mathbf{R}_{ik}^{-1}\mathbf{u}_{ik} \tag{C.4}$$

*and the **virtual measurement covariances** $\mathbf{R}_{ij}$ are defined by*

$$\mathbf{R}_{ij}^{-1} \stackrel{\Delta}{=} \sum_{k=1}^{K_i} f_{ijk}^t \mathbf{R}_{ik}^{-1} \tag{C.5}$$

Proof. The log-likelihood for a single measurement $\mathbf{u}_{ik}$ can be obtained by taking the logarithm of the Gaussian conditional density (C.1) and dropping the constant:

$$\log P(\mathbf{u}_{ik}|\mathbf{m}_i, \mathbf{x}_j) \equiv -\frac{1}{2}(\mathbf{u}_{ik} - \mathbf{h}_{ij})^T \mathbf{R}_{ik}^{-1}(\mathbf{u}_{ik} - \mathbf{h}_{ij})$$

Substituting this into (C.2) we get the following expression for the expected log-likelihood $Q^t(\mathbf{\Theta})$:

$$Q^t(\mathbf{\Theta}) \equiv -\frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{n}\sum_{k=1}^{K_i} f_{ijk}^t(\mathbf{u}_{ik} - \mathbf{h}_{ij})^T \mathbf{R}_{ik}^{-1}(\mathbf{u}_{ik} - \mathbf{h}_{ij}) \tag{C.6}$$

Expanding the square in (C.6) we obtain

$$-\frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{n}\sum_{k=1}^{K_i} f_{ijk}^t(\mathbf{u}_{ik}^T\mathbf{R}_{ik}^{-1}\mathbf{u}_{ik} - 2\mathbf{h}_{ij}^T\mathbf{R}_{ik}^{-1}\mathbf{u}_{ik} + \mathbf{h}_{ij}^T\mathbf{R}_{ik}^{-1}\mathbf{h}_{ij})$$

Now distribute the sum over measurement indices k, taking constants out of the sums:

$$-\frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{n}\left[\sum_{k=1}^{K_i} f_{ijk}^t\mathbf{u}_{ik}^T\mathbf{R}_{ik}^{-1}\mathbf{u}_{ik} - 2\mathbf{h}_{ij}^T\sum_{k=1}^{K_i} f_{ijk}^t\mathbf{R}_{ik}^{-1}\mathbf{u}_{ik} + \mathbf{h}_{ij}^T\left(\sum_{k=1}^{K_i} f_{ijk}^t\mathbf{R}_{ik}^{-1}\right)\mathbf{h}_{ij}\right] \tag{C.7}$$

The first term in the square brackets can be isolated as a constant C, and we apply the definition of $\mathbf{R}_{ij}$ (definition C.5 on the page before) in the last term:

$$C - \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{n}(-2\mathbf{h}_{ij}^T\sum_{k=1}^{K_i} f_{ijk}^t\mathbf{R}_{ik}^{-1}\mathbf{u}_{ik} + \mathbf{h}_{ij}^T\mathbf{R}_{ij}^{-1}\mathbf{h}_{ij}) \tag{C.8}$$

Now, *define* $\mathbf{v}_{ij}^t$ to satisfy the equation below

$$-2\mathbf{h}_{ij}^T\sum_{k=1}^{K_i} f_{ijk}^t\mathbf{R}_{ik}^{-1}\mathbf{u}_{ik} = -2\mathbf{h}_{ij}^T\mathbf{R}_{ij}^{-1}\mathbf{v}_{ij}^t \tag{C.9}$$

which is obtained by

$$\mathbf{v}_{ij}^t = \mathbf{R}_{ij}\sum_{k=1}^{K_i} f_{ijk}^t\mathbf{R}_{ik}^{-1}\mathbf{u}_{ik}$$

i.e. this is definition C.4. Apply the transformation (C.9) in equation (C.8) to yield

$$C - \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{n}(-2\mathbf{h}_{ij}^T\mathbf{R}_{ij}^{-1}\mathbf{v}_{ij}^t + \mathbf{h}_{ij}^T\mathbf{R}_{ij}^{-1}\mathbf{h}_{ij})$$

Completing the square, we obtain

$$C - \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{n}(-\mathbf{v}_{ij}^t\mathbf{R}_{ij}^{-1}\mathbf{v}_{ij}^t + \mathbf{v}_{ij}^t\mathbf{R}_{ij}^{-1}\mathbf{v}_{ij}^t - 2\mathbf{h}_{ij}^T\mathbf{R}_{ij}^{-1}\mathbf{v}_{ij}^t + \mathbf{h}_{ij}^T\mathbf{R}_{ij}^{-1}\mathbf{h}_{ij})$$

Since the first term, $\frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{n}\mathbf{v}_{ij}^{t}\mathbf{R}_{ij}^{-1}\mathbf{v}_{ij}^{t}$, is independent of $\boldsymbol{\Theta}$ (since it does not include $\mathbf{h}_{ij}$) we can absorb it in the constant. Rewriting the remaining terms as a square yields the desired expression (C.3). □

Bibliography

[1] Aggarwal, J., Davis, L., and Martin, W. (1981). Correspondence processes in dynamic scene analysis. *Proceedings of IEEE*, 69(5):562–572.

[2] Avitzour, D. (1992). A maximum likelihood approach to data association. *IEEE Trans. on Aerospace and Electronic Systems*, 28(2):560–566.

[3] Ayache, N. and Faugeras, O. (1986). Hyper: A new approach for the representation and positioning of two-dimensional objects. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 8(1):44–54.

[4] Azarbayejani, A. and Pentland, A. P. (1995). Recursive estimation of motion, structure, and focal length. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 17(6):562.

[5] Baird, H. (1985). *Model-Based Image Matching Using Location*. MIT Press.

[6] Bar-Shalom, Y. and Fortmann, T. (1988). *Tracking and data association*. Academic Press, New York.

[7] Bar-Shalom, Y., Fortmann, T., and Scheffe, M. (1980). Joint probabilistic data association for multiple targets in clutter. In *Proc. Conf. on Information Sciences and Systems*.

[8] Bar-Shalom, Y. and Li, X. (1993). *Estimation and Tracking: principles, techniques and software*. Artech House, Boston, London.

[9] Bar-Shalom, Y. and Tse, E. (1975). Tracking in a cluttered environment with probabilistic data-association. *Automatica*, 11:451–460.

[10] Barrow, H., Tenenbaum, J., Bolles, R., and Wolf, H. (1977). Parametric correspondence and chamfer matching: Two new techniques for image matching. In *Proc. Fifth Int'l Joint Conf. Artificial Intelligence*, pages 659–663.

[11] Basri, R., Grove, A., and Jacobs, D. (1998). Efficient determination of shape from multiple images containing partial information. *Pattern Recognition*, 31(11):1691–1703.

[12] Beardsley, P., Torr, P., and Zisserman, A. (1996). 3D model acquisition from extended image sequences. In *Eur. Conf. on Computer Vision (ECCV)*, pages II:683–695.

[13] Bedekar, A. and Haralick, R. (1996). Finding correspondence points based on Bayesian triangulation. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 61–66.

[14] Bergman, N. and Doucet, A. (2000). Markov chain Monte Carlo data association for target tracking. In *Int. Conf. Acoust., Speech, and Signal Proc. (ICASSP)*.

[15] Bertsekas, D. (1991). *Linear Network Optimization: Algorithms and Codes*. The MIT press, Cambridge, MA.

[16] Besl, P. and McKay, N. (1992). A method for registration of 3-D shapes. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 14(2).

[17] Binford, T. (1982). Survey of model-based image analysis. *International Journal of Robotics Research*, 1(1):18–64.

[18] Blackman, S. (1986). *Multiple-Target Tracking with Radar Applications*. Artech House, Norwood, MA.

[19] Bolles, R. and Fischler, M. (1981). A RANSAC-based approach to model fitting and its application to finding cylinders in range data. In *Seventh International Joint Conference on Artificial Intelligence, (Vancouver, British Colombia, Canada)*, pages 637–643.

[20] Boykov, Y. and Huttenlocher, D. (1999). A new Bayesian approach to object recognition. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages II:517–523.

[21] Broder, A. Z. (1986). How hard is to marry at random? (On the approximation of the permanent). In *Proceedings of the Eighteenth Annual ACM Symposium on Theory of Computing*, pages 50–58, Berkeley, California.

[22] Broida, T. and Chellappa, R. (1991). Estimating the kinematics and structure of a rigid object from a sequence of monocular images. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 13(6):497–513.

[23] Castellanos, J., Montiel, J., Neira, J., and Tardos, J. (1999). The SPmap: A probabilistic framework for simultaneous localization and map building. *IEEE Trans. on Robotics and Automation*, 15(5):948–953.

[24] Castellanos, J. and Tardos, J. (2000). *Mobile Robot Localization and Map Building: A Multisensor Fusion Approach*. Kluwer Academic Publishers, Boston, MA.

[25] Cheng, Y., Collins, R., Hanson, A., and Riseman, E. (1994). Triangulation without correspondences. In *DARPA Image Understanding Workshop (IUW)*.

[26] Cheng, Y., Wu, V., Collins, R., Hanson, A., and Riseman, E. (1996). Maximum-weight bipartite matching technique and its application in image feature matching. In *Proc. SPIE Visual Comm. and Image Processing, Orlando, FL*.

[27] Chui, H. and Rangarajan, A. (2000). A new algorithm for non-rigid point matching. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*.

[28] Cook, W., Cunningham, W., Pulleyblank, W., and Schrijver, A. (1998). *Combinatorial Optimization*. John Wiley & Sons, New York, NY.

[29] Cooper, M. and Robson, S. (1996). Theory of close range photogrammetry. In Atkinson, K., editor, *Close range photogrammetry and machine vision*, chapter 1, pages 9–51. Whittles Publishing.

[30] Cox, I. (1991). Blanche—an experiment in guidance and navigation of an autonomous robot vehicle. *IEEE Trans. on Robotics and Automation*, 7(2):193–204.

[31] Cox, I. (1993). A review of statistical data association techniques for motion correspondence. *Int. J. of Computer Vision*, 10(1):53–66.

[32] Cox, I. and Hingorani, S. (1994). An efficient implementation and evaluation of Reid's multiple hypothesis tracking algorithm for visual tracking. In *Int. Conf. on Pattern Recognition (ICPR)*, volume 1, pages 437–442, Jerusalem, Israel.

[33] Cox, I. and Hingorani, S. (1996). An efficient implementation of Reid's multiple hypothesis tracking algorithm and its evaluation for the purpose of visual tracking. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 18(2):138–150.

[34] Cox, I., Hingorani, S., Rao, S., and Maggs, B. (1996). A maximum likelihood stereo algorithm. *CVGIP:Image Understanding*, 63(3):542–567.

[35] Cox, I. and Leonard, J. (1994). Modeling a dynamic environment using a Bayesian multiple hypothesis approach. *Artificial Intelligence*, 66(2):311–344.

[36] Cox, I. and Miller, M. (1995). On finding ranked assignments with application to multi-target tracking and motion correspondence. *IEEE Trans. on Aerospace and Electronic Systems*, 31(1):486.

[37] Cross, A. and Hancock, E. (1998). Graph matching with a dual-step EM algorithm. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 20(11):1236–1253.

[38] Danchick, R. and Newnam, G. (1993). A fast method for finding the exact n-best hypotheses for multitarget tracking. *IEEE Trans. on Aerospace and Electronic Systems*, 29:555–560.

[39] Deb, S., Yeddanapudi, M., Pattipati, K., and Bar-Shalom, Y. (1997). A generalized S-D assignment algorithm for multisensor-multitarget state estimation. *IEEE Trans. on Aerospace and Electronic Systems*, 33(2):523–538.

[40] Dellaert, F., Seitz, S., Thorpe, C., and Thrun, S. (2001). EM, MCMC, and chain flipping for structure from motion with unknown correspondence. *Machine learning*. special issue on Markov chain Monte Carlo methods, to appear 2001.

[41] Dellaert, F., Thorpe, C., and Thrun, S. (1998a). Super-resolved tracking of planar surface patches. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*.

[42] Dellaert, F., Thrun, S., and Thorpe, C. (1998b). Jacobian images of super-resolved texture maps for model-based motion estimation and tracking. In *IEEE Workshop on Applications of Computer Vision (WACV)*.

[43] Dempster, A., Laird, N., and Rubin, D. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B*, 39(1):1–38.

[44] Deriche, R. and Faugeras, O. (1990). Tracking line segments. *Image and Vision Computing*, 8:261–270.

[45] Dissanayake, G., Durrant-Whyte, H., and Bailey, T. (2000). A computationally efficient solution to the simultaneous localisation and map building (slam) problem. Working notes of ICRA'2000 Workshop W4: Mobile Robot Navigation and Mapping.

[46] Doucet, A., Gordon, N., and de Freitas, J., editors (2001). *Sequential Monte Carlo Methods In Practice*. Springer-Verlag, Ney York.

[47] Durbin, R., Szeliski, R., and Yuille, A. (1989). An analysis of the elastic net approach to the travelling salesman problem. *Neural Computation*, 1:348–358.

[48] Durbin, R. and Willshaw, D. (1987). An analog approach to the travelling salesman problem using an elastic net method. *Nature*, (326):689–691.

[49] Durrant-Whyte, H., Majunder, S., Thrun, S., de Battista, M., and Scheding, S. (2001). A Bayesian algorithm for simultaneous localization and map building. submitted for publication.

[50] Faugeras, O. (1993). *Three-dimensional computer vision: A geometric viewpoint*. The MIT press, Cambridge, MA.

[51] Faugeras, O. and Luong, Q. (2001). *The geometry of multiple images*. MIT Press. with contributions from T. Papadopoulo.

[52] Feldmar, J. and Ayache, N. (1996). Rigid, affine and locally affine registration of free-form surfaces. *Int. J. of Computer Vision*, 18:99–119.

[53] Fielding, G. and Kam, M. (1997). Applying the Hungarian method to stereo matching. In *Proc. 1997 IEEE Conference on Decision and Control*, pages 549–558.

[54] Fielding, G. and Kam, M. (2000). Weighted matchings for dense stereo correspondence. *Pattern Recognition*, 9:1511–1524.

[55] Fischler, M. and Bolles, R. (1981). Random sample consensus: a paradigm for model fitting with application to image analysis and automated cartography. *Commun. Assoc. Comp. Mach.*, 24:381–395.

[56] Forsyth, D., Ioffe, S., and Haddon, J. (1999). Bayesian structure from motion. In *Int. Conf. on Computer Vision (ICCV)*, pages 660–665.

[57] Fortmann, T., Bar-Shalom, Y., and Scheffe, M. (1980). Multi-target tracking using joint probabilistic data association. In *Proc. 19th IEEE Conf. on Decision & Control*.

[58] Fortmann, T., Bar-Shalom, Y., and Scheffe, M. (1983). Sonar tracking of multiple targets using joint probabilistic data association. *IEEE Journal of Oceanic Engineering*, 8.

[59] Fua, P. (1997). From multiple stereo views to multiple 3D surfaces. *Int. J. of Computer Vision*, 24(1):19–35.

[60] Gauvrit, H., Le Cadre, J., and Jauffret, C. (1997). A formulation of multitarget tracking as an incomplete data problem. *IEEE Trans. on Aerospace and Electronic Systems*, 33(4):1242–1257.

[61] Gavrila, D. and Davis, L. (1996). Model-based tracking of humans in action: a multi-view approach. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 73–80.

[62] Gelman, A. (1996). Inference and monitoring convergence. In (Gilks et al., 1996), pages 131–140.

[63] Gilks, W., Richardson, S., and Spiegelhalter, D., editors (1996). *Markov chain Monte Carlo in practice*. Chapman and Hall.

[64] Gold, S. and Rangarajan, A. (1996). A graduated assignment algorithm for graph matching. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 18(4):377–388.

[65] Gold, S., Rangarajan, A., Lu, C., Pappu, S., and Mjolsness, E. (1998). New algorithms for 2D and 3D point matching. *Pattern Recognition*, 31(8):1019–1031.

[66] Goldgof, D., Huang, T., and Lee, H. (1989). Motion estimation from points without correspondences from orthographic projections. In *Proc. Workshop on Visual Motion*, pages 352–358.

[67] Goldgof, D., Lee, H., and Huang, T. (1992). Matching and motion estimation of three-dimensional point and line sets using eigenstructure without correspondences. *Pattern Recognition*, 25:271–286.

[68] Griewank, A. (1989). On Automatic Differentiation. In Iri, M. and Tanabe, K., editors, *Mathematical Programming: Recent Developments and Applications*, pages 83–108. Kluwer Academic Publishers.

[69] Grimson, W. (1990). *Object recognition by computer : the role of geometric constraints*. MIT Press.

[70] Grimson, W. and Lozano-Pérez, T. (1987). Localizing Overlapping Parts by Searching the Interpretation Tree. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 9(4):469–482.

[71] Grimson, W. E. L., Lozano-Pérez, T., Wells, W. M., Ettinger, G. J., White, S. J., and Kikinis, R. (1996). An automatic registration method for frameless stereotaxy, image guided surgery, and enhanced reality visualization. *IEEE Trans. on Medical Imaging*, 15:126–141.

[72] Gutmann, J.-S. and Nebel, B. (1997). Navigation mobiler roboter mit laserscans. In *Autonome Mobile Systeme*, Berlin. Springer Verlag.

[73] Hartley, H. (1958). Maximum likelihood estimation from incomplete data. *Biometrics*, 14:174–194.

[74] Hartley, R. (1994). Euclidean reconstruction from uncalibrated views. In *Application of Invariance in Computer Vision*, pages 237–256.

[75] Hartley, R. (1997). Lines and points in three views and the trifocal tensor. *Int. J. of Computer Vision*, 22(2):125– 140.

[76] Hartley, R. and Zisserman, A. (2000). *Multiple View Geometry in Computer Vision*. Cambridge University Press.

[77] Hastings, W. (1970). Monte carlo sampling methods using markov chains and their applications. *Biometrika*, 57:97–109.

[78] Hopfield, J. J. and Tank, D. W. (1985). Neural computation of decisions in optimization problems. *Biological Cybernetics*, 52:147–152.

[79] Hornegger, J. (1997). Statistical modeling of relations for 3-D object recognition. In *Int. Conf. Acoust., Speech, and Signal Proc. (ICASSP)*, volume 4, pages 3173–3176, Munich.

[80] Huttenlocher, D., Klanderman, G., and Rucklidge, W. (1993). Comparing images using the Hausdorff distance. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 15(9):850–863.

[81] Irani, M. and Anandan, P. (1999). About direct methods. In Triggs, B., Zisserman, A., and Szeliski, R., editors, *Vision Algorithms: Theory and Practice*, number 1883 in LNCS, pages 267–277, Corfu, Greece. Springer-Verlag.

[82] Jacobs, D. (1997). Linear fitting with missing data: Applications to structure from motion and to characterizing intensity images. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 206–212.

[83] Jazwinsky, A. (1970). *Stochastic Processes and Filtering Theory*. Academic Press, New York.

[84] Jensfelt, P. and Kristensen, S. (1999). Active global localisation for a mobile robot using multiple hypothesis tracking. In *Int. Joint Conf. on Artificial Intelligence (IJCAI)*, pages 13–22.

[85] Kang, S. B. and Szeliski, R. (1997). 3-D scene data recovery using omnidirectional multibaseline stereo. *Int. J. of Computer Vision*, 25(2):167–183.

[86] Kim, W.-Y. and Kak, A. (1991). 3-D object recognition using bipartite matching embedded in discrete relaxation. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 13(3):224 –251.

[87] Kirubarajan, T., Bar-Shalom, Y., and Pattipati, K. (2001). Multiassignment for tracking a large number of overlapping objects [and application to fibroblast cells]. *IEEE Trans. on Aerospace and Electronic Systems*, 37(1):2–21.

[88] Kosowsky, J. J. and Yuille, A. L. (1994). The invisible hand algorithm - solving the assignment problem with statistical physics. *Neural Networks*, 7(3):477–490.

[89] Kozen, D. C. (1991). *The design and analysis of algorithms*. Springer-Verlag.

[90] Lee, C. and Huang, T. (1988). Finding point correspondences and determining motion of a rigid object from two weak perspective views. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 398–403.

[91] Lee, C. and Joshi, A. (1993). Correspondence problem in image sequence analysis. *Pattern Recognition*, 26(1):47–61.

[92] Leonard, J. and Durrant-Whyte, H. (1991a). Mobile robot localization by tracking geometric beacons. *IEEE Trans. on Robotics and Automation*, 7(3):376–382.

[93] Leonard, J. and Durrant-Whyte, H. (1991b). Simultaneous map building and localization for an autonomous mobile robot. In *IEEE Int. Workshop on Intelligent Robots and Systems*, pages 1442–1447.

[94] Leonard, J. and Feder, H. (1999). A computationally efficient method for large-scale concurrent mapping and localization. In Hollerbach, J. and Koditschek, D., editors, *Proceedings of the Ninth International Symposium on Robotics Research*, Salt Lake City, Utah.

[95] Li, S. (1992). Matching: Invariant to translations, rotations, and scale changes. *Pattern Recognition*, 25:583–594.

[96] Li, X. and Bar-Shalom, Y. (1996). Tracking in clutter with nearest neighbor filters: analysis and performance. *IEEE Trans. on Aerospace and Electronic Systems*, 32(3):995–1010.

[97] Liu, Y., Emery, R., Chakrabarti, D., Burgard, W., and Thrun, S. (2001). Using em to learn 3D models with mobile robots.

[98] Longuet-Higgins, H. (1981). A computer algorithm for reconstructing a scene from two projections. *Nature*, 293:133–135.

[99] Lowe, D. (1991). Fitting parameterized three-dimensional models to images. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 13(5):441–450.

[100] Lowe, D. (1999). Object recognition from local scale-invariant features. In *Int. Conf. on Computer Vision (ICCV)*, pages 1150–1157.

[101] Lu, C. and Mjolsness, E. (1994). Two-dimensional object localization by coarse-to-fine correlation matching. In Cowan, J., Tesauro, G., and Alspector, J., editors, *Advances in Neural Information Processing Systems (NIPS)*, volume 6, pages 985–992. Morgan Kaufmann Publishers, Inc.

[102] Lu, F. and Milios, E. (1997). Globally consistent range scan alignment for environment mapping. Technical report, Department of Computer Science, York University.

[103] Luong, Q.-T. and Faugeras, O. (1996). The Fundamental matrix: theory, algorithms, and stability analysis. *Int. J. of Computer Vision*, 17(1):43–76.

[104] MacCormick, J. and Blake, A. (1998). Spatial dependence in the observation of visual contours. In *ECCV*, pages 765–781.

[105] Maybeck, P. (1979). *Stochastic Models, Estimation and Control*, volume 1. Academic Press, New York.

[106] Maybeck, P. (1982). *Stochastic Models, Estimation and Control*, volume 2. Academic Press, New York.

[107] McLachlan, G. and Krishnan, T. (1997). *The EM algorithm and extensions*. Wiley series in probability and statistics. John Wiley & Sons.

[108] McLauchlan, P. and Murray, D. (1995). A unifying framework for structure and motion recovery from image sequences. In *Int. Conf. on Computer Vision (ICCV)*, pages 314–320.

[109] Metropolis, N., Rosenbluth, A., Rosenbluth, M., Teller, A., and Teller, E. (1953). Equations of state calculations by fast computing machine. *Journal of Chemical Physics*, 21:1087–1091.

[110] Mikolajczyk, K. and Schmid, C. (2001). Indexing based on scale invariant interest points. In *Int. Conf. on Computer Vision (ICCV)*.

[111] Minka, T. (1998). Expectation-Maximization as lower bound maximization. Tutorial published on the web at http://www-white.media.mit.edu/ tpminka/papers/em.html.

[112] Molnar, K. and Modestino, J. (1998). Application of the EM algorithm for the multitarget/multisensor tracking problem. *IEEE Trans. on Signal Processing*, 46(1).

[113] Montesinos, P., Gouet, V., and Deriche, R. (1998). Differential invariants for color images.

[114] Morefield, C. (1977). Application of 0-1 integer programming to multitarget tracking problems. *IEEE Trans. on Automation and Control*, 22(3):302–312.

[115] Morris, D. and Kanade, T. (1998). A unified factorization algorithm for points, line segments and planes with uncertainty models. In *Int. Conf. on Computer Vision (ICCV)*, pages 696–702.

[116] Morris, D., Kanatani, K., and Kanade, T. (1999). Uncertainty modeling for optimal structure from motion. In *ICCV Workshop on Vision Algorithms: Theory and Practice.*

[117] Neal, R. (1993). Probabilistic inference using Markov chain Monte Carlo methods. Technical Report CRG-TR-93-1, Dept. of Computer Science, University of Toronto.

[118] Neal, R. and Hinton, G. (1998). A view of the EM algorithm that justifies incremental, sparse, and other variants. In Jordan, M., editor, *Learning in Graphical Models*. Kluwer Academic Press.

[119] Okutomi, M. and Kanade, T. (1993). A multiple-baseline stereo algorithm. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 15(4):353–363.

[120] Olson, C. F. (2000). Maximum-likelihood template matching. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 52–57.

[121] Papadimitriou, C. and Steiglitz, K. (1982). *Combinatorial Optimization: Algorithms and Complexity*. Prentice-Hall.

[122] Pasula, H., Russell, S., Ostland, M., and Ritov, Y. (1999). Tracking many objects with many sensors. In *Int. Joint Conf. on Artificial Intelligence (IJCAI)*, Stockholm.

[123] PattiPati, K., Deb, S., and Bar-Shalom, Y. (1990). Passive multisensor data association using a new relaxation algorithm. In *Multitarget-Multisensor Tracking: Advanced Applications*, pages 219–246. Artech House.

[124] Pattipati, K., Deb, S., Bar-Shalom, Y., and Washburn, R.B., J. (1992). A new relaxation algorithm and passive sensor data association. *IEEE Transactions on Automatic Control*, 37(2):198 –213.

[125] Paulus, D., Hornegger, J., and Niemann, H. (1997). A framework for statistical 3-d object recognition. *Pattern Recognition Letters*, 18:1153–1157.

[126] Pilu, M. (1997). A direct method for stereo correspondence based on singular value decomposion. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 261–266.

[127] Poelman, C. and Kanade, T. (1997). A paraperspective factorization method for shape and motion recovery. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 19(3):206–218.

[128] Popoli, R. and Blackman, S. S. (1999). *Design and Analysis of Modern Tracking Systems*. Artech House Radar Library.

[129] Price, K. (1986). Hierarchical matching using relaxation. *Computer Vision Graphics Image Proc.*, 34:66–75.

[130] Pritchett, P. and Zisserman, A. (1998a). Matching and reconstruction from widely separated views. In *SMILE 98 European Workshop on 3D Structure from Multiple Images of Large-Scale Environments, Freiburg, Germany*.

[131] Pritchett, P. and Zisserman, A. (1998b). Wide baseline stereo matching. In *Int. Conf. on Computer Vision (ICCV)*, pages 754–760.

[132] Pulford, G. W. and Logothetis, A. (1997). An expectation-maximisation tracker for multiple observations of a single target in clutter. In *Proc. 35th Conference on Decision and Control, Kobe, Dec. 1996*, pages 4997–5003.

[133] Pulford, G. W. and Scala, B. F. L. (1996). Manoeuvring target tracking using the expectation-maximisation algorithm. In *Proc. 4th Int Conf. on Control, Automation, Robotics and Vision, Singapore*, pages 2340–2344.

[134] Rago, C., Willett, P., and Streit, R. (1995). A comparison of the jpdaf and pmht tracking algorithms. In *International Conference on Acoustics, Speech, and Signal Processing*, volume 5, pages 3571 –3574.

[135] Ranade, S. and Rosenfeld, A. (1980). Point pattern matching by relaxation. *Pattern Recognition*, 12:269–275.

[136] Rangarajan, A. and Mjolsness, E. (1994). A lagrangian relaxation network for graph matching. In *Proc. Int. Conf. Neural Networks*, volume 7, pages 4629–4634. Inst. Electrical & Electronics Engineers.

[137] Rangarajan, A., Mjolsness, E., Pappu, S., and Davachi, L. (1997). A robust point matching algorithm for autoradiograph alignment. *Medical Image Analysis*, 4(1):379–398.

[138] Rasmussen, C. and Hager, G. (1998). Joint probabilistic techniques for tracking objects using multiple vision clues. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 191–196.

[139] Rasmussen, C. and Hager, G. (2001). Probabilistic data association methods for tracking complex visual objects. *PAMI*, 23(6):560–576.

[140] Reid, D. (1979). An algorithm for tracking multiple targets. *IEEE Trans. on Automation and Control*, AC-24(6):84–90.

[141] Reuter, J. (2000). Mobile robot localization using pdab. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*.

[142] Robert, C. and Casella, G. (1999). *Monte Carlo Statistical Methods*. Springer.

[143] Rosenfeld, A. and Pfalz, J. L. (1966). Sequential operations in digital picture processing. *Journal of the Association for Computing Machinery*, 13:471–494.

[144] Roumerliotis, S. and Bekey, G. (2000). Bayesian estimation and Kalman filtering: a unified framework for mobile robot localization. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*, pages 2985–2992.

[145] Roy, S. and Cox, I. (1998). A maximum-flow formulation of the n-camera stereo correspondence problem. In *Int. Conf. on Computer Vision (ICCV)*, pages 492–499.

[146] Schmid, C. and Mohr, R. (1997). Local grayvalue invariants for image retrieval. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 19(5):530–535.

[147] Schulz, D., Burgard, W., Fox, D., and Cremers., A. B. (2001). Tracking multiple moving targets with a mobile robot using particle filters and statistical data association. In *IEEE Int. Conf. on Robotics and Automation (ICRA)*.

[148] Scott, G. and Longuet-Higgins, H. (1991). An algorithm for associating the features of two images. *Proceedings of Royal Society of London*, B-244:21–26.

[149] Seitz, S. and Dyer, C. (1995). Complete structure from four point correspondences. In *Int. Conf. on Computer Vision (ICCV)*, pages 330–337.

[150] Sengupta, D. and Iltis, R. (1988). Computationally efficient tracking of multiple targets by probabilistic data association using neural networks. In *Int. Conf. Acoust., Speech, and Signal Proc. (ICASSP)*, volume 4, pages 2152–2155.

[151] Sengupta, D. and Iltis, R. (1989). Neural solution to the multitarget tracking data association problem. *IEEE Trans. on Aerospace and Electronic Systems*, 25(1):96–108.

[152] Shapiro, L. and Brady, J. (1992). Feature-based correspondence: An eigenvector approach. *Image and Vision Computing*, 10(5):283–288.

[153] Shapiro, L. and Haralick, R. (1981). Structural descriptions and inexact matching. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 3(5):504–519.

[154] Shashua, A. and Werman, M. (1995). Trilinearity of three perspective views and its associated tensor. In *Int. Conf. on Computer Vision (ICCV)*, pages 920–925.

[155] Shum, H.-Y. and Szeliski, R. (2000). Construction of panoramic mosaics with global and local alignment. *Int. J. of Computer Vision*, 36(2):101–130.

[156] Singer, R., Sea, R., and Housewright, K. (1974). Derivation and evaluation of improved tracking filters for use in dense multitarget environments. *IEEE Trans. Information Theory*, 20:423–432.

[157] Sinkhorn, R. (1964). A relationship between arbitrary positive matrices and doubly stochastic matrices. *Annals of Mathematica Statistics*, 35(2):876–879.

[158] Sittler, R. (1964). An optimal data association problem in surveillance theory. *IEEE Transactions on Military Electronics*, MIL-8:125–139.

[159] Smith, P. and Buechler, G. (1975). A branching algorithm for discriminating and tracking multiple objects. *IEEE Trans. Automatic Control*, 20:101–104.

[160] Spetsakis, M. and Aloimonos, Y. (1991). A multi-frame approach to visual motion perception. *Int. J. of Computer Vision*, 6(3):245–255.

[161] Streit, R. and Luginbuhl, T. (1994). Maximum likelihood method for probabilistic multi-hypothesis tracking. In *Proc. SPIE, Vol. 2335*, pages 394–405.

[162] Szeliski, R. (1989). *Bayesian modeling of uncertainty in low-level vision*. Kluwer Academic Publishers.

[163] Szeliski, R. and Kang, S. (1993). Recovering 3D shape and motion from image streams using non-linear least squares. Technical Report CRL 93/3, DEC Cambridge Research Lab.

[164] Szeliski, R. and Tonnesen, D. (1992). Surface modeling with oriented particle systems. *Computer Graphics (SIGGRAPH'92)*, 26(2):185–194.

[165] Tanner, M. (1996). *Tools for Statistical Inference*. Springer Verlag, New York. Third Edition.

[166] Taylor, C. and Kriegman, D. (1995). Structure and motion from line segments in multiple images. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 17(11):1021–1032.

[167] Thrun, S., Fox, D., and Burgard, W. (1998). A probabilistic approach to concurrent mapping and localization for mobile robots. *Machine Learning*, 31:29–53. also appeared in Autonomous Robots 5, 253–271.

[168] Tomasi, C. and Kanade, T. (1992). Shape and motion from image streams under orthography: a factorization method. *Int. J. of Computer Vision*, 9(2):137–154.

[169] Ton, J. and Jain, A. (1989). Registering landsat images using point pattern matching. *IEEE Transactions on Geoscience and Remote Sensing*, 27(5):642–651.

[170] Torr, P., Fitzgibbon, A., and Zisserman, A. (1998). Maintaining multiple motion model hypotheses over many views to recover matching and structure. In *Int. Conf. on Computer Vision (ICCV)*, pages 485–491.

[171] Torr, P. and Murray, D. (1993). Outlier detection and motion segmentation. In *Proceedings SPIE Sensor Fusion Conference*, pages 432–443.

[172] Torr, P. and Murray, D. (1997). The development and comparison of robust methods for estimating the fundamental matrix. *Int. J. of Computer Vision*, 24(3):271–300.

[173] Torr, P. and Zisserman, A. (1998). Robust computation and parametrization of multiple view relations. In *Int. Conf. on Computer Vision (ICCV)*, pages 485–491.

[174] Torr, P. H. S. and Zisserman, A. (1999). Feature based methods for structure and motion estimation. In Triggs, B., Zisserman, A., and Szeliski, R., editors, *Vision Algorithms: Theory and Practice*, number 1883 in LNCS, pages 278–294, Corfu, Greece. Springer-Verlag.

[175] Triggs, B. (1996). Factorization methods for projective structure and motion. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 845–851.

[176] Triggs, B., McLauchlan, P., Hartley, R., and Fitzgibbon, A. (1999). Bundle adjustment – a modern synthesis. In *Vision Algorithms 99*, Corfu, Greece.

[177] Tsai, R. and Huang, T. (1984). Uniqueness and estimation of three-dimensional motion parameters of rigid objects with curved surfaces. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 6(1):13–27.

[178] Tuytelaars, T. and Van Gool, L. (2000). Matching widely separated views based on affinely invariant neighbourhoods. In *British Machine Vision Conference (BMVC)*, pages 412–422.

[179] Tuytelaars, T. and Van Gool, L. (2001). Matching widely separated views based on affinely invariant neighbourhoods. submitted to Int. Journal on Computer Vision.

[180] Ullman, S. (1979). *The interpretation of visual motion*. MIT Press, Cambridge, MA.

[181] Umeyama, S. (1988). An eigendecomposition approach to weighted graph matching problems. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 10(5):695 –703.

[182] Umeyama, S. (1991). Least-squares estimation of transformation parameters between two point patterns. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 13(4):376 –380.

[183] Umeyama, S. (1993). Parameterized point pattern matching and its application to recognition of object families. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 15(2):136 –144.

[184] Wang, C., H., H. S., Yada, S., and Rosenfeld, A. (1983). Some experiments in relaxation image matching using corner features. *Pattern Recognition*, 16(2):167–182.

[185] Wells, W. (1997). Statistical approaches to feature-based object recognition. *Int. J. of Computer Vision*, 21(1/2):63–98.

[186] Weng, J., Ahuja, N., and Huang, T. (1993). Optimal motion and structure estimation. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 15:864–884.

[187] Willett, P., Ruan, Y., and Streit, R. (1999). Making the probabilistic multi-hypothesis tracker the tracker of choice. In *Proceedings 1999 IEEE Aerospace Conference*, volume 4, pages 387 –399.

[188] Yuille, A., Geiger, D., and Bulthoff, H. (1991). Stereo, mean field theory and psychophysics. *Network: Computation in Neural Systems*, 2:423–442.

[189] Zhang, J. and Katsaggelos, A. (1996). Image recovery using the EM algorithm. In *DSP Handbook*. CRC Press/IEEE Press.

[190] Zhou, B. and Bose, N. (1993). Multitarget tracking in clutter: fast algorithms for data association. *IEEE Trans. on Aerospace and Electronic Systems*, 29(2):352–363.

[191] Zhou, B. and Bose, N. (1995). An efficient algorithm for data association in multitarget tracking. *IEEE Trans. on Aerospace and Electronic Systems*, 31(1):458 –468.

Index