Interactive Search of Adipocytes in Large Collections of Digital Cellular Images

**Adam Goode[†], Mei Chen[‡], Anil Tarachandani[•], Lily Mummert[‡],
Rahul Sukthankar[‡], Casey Helfrich[‡], Alice Stefanni[•], Limor Fix[‡],
Jeffrey Saltzman[•], M. Satyanarayanan[†]**

December 2006
CMU-CS-06-177

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

[†]Carnegie Mellon University, [‡]Intel Research Pittsburgh, [•]Merck & Co., Inc.

Abstract

In the field of lipid research, the measurement of adipocyte size is an important but difficult problem. We describe an imaging-based solution that combines precise investigator control with semi-automated quantitation. By using unfixed live cells, we avoid many complications that arise in trying to isolate individual adipocytes. Instead, we image a small drop of live adipocyte suspension under a microscope, and then quantitate the image using an open-source software tool called FatFind. Since we have developed FatFind on the open-source Diamond distributed search platform, it inherits the scaling, parallelism and remote access attributes of Diamond. This paper reports on the design, implementation, and evaluation of FatFind.

This research was partly supported by the National Science Foundation (NSF) under grant number CNS-0614679. Any opinions, findings, conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the NSF, Carnegie Mellon University, Intel, or Merck.

Keywords: Diamond, OpenDiamond, interactive search, non-indexed search, image processing, measurement tool, FatFind, computer vision, LTI-Lib, ellipse detection, circle detection, lipid

1. Introduction

Adipocytes, or fat cells, serve as reservoirs of energy in the form of lipids in humans, and are tightly regulated with respect to their size and number. Adipose tissue mass, which is comprised mainly of adipocytes, is dependent on the volume as well as the number of adipocytes. Regulation of adipose mass involves endocrine, paracrine and autocrine systems and hypothalamic centers that control appetite, metabolic rate and activity levels. Significant alteration in body mass involves alterations in both adipocyte volume and number.

In obese individuals, an excessive amount of adipose tissue has been linked with the development of type 2 diabetes, premature atherosclerosis, and cardiovascular disease. Adipocyte-derived factors are significantly increased in obesity and represent good predictors of the development of type 2 diabetes [1][2][3]. Moreover, the increase in fat mass has been strongly correlated to the size of the adipocytes, especially in females [4][5][6][7][8]. In the field of lipid research, the measurement of adipocyte size has served as a good marker for a change in fat mass.

In order for researchers to make discoveries of statistical significance, it is important to have tools that enable them to explore large data repositories in an efficient and flexible manner. Specifically, techniques are needed that can quickly locate adipocytes and accurately compute their sizes, as well as mechanisms that allow researchers to study adipocytes of similar sizes across different data samples. Although there has been related work on automated detection and measurement of adipocytes [9] [10][11][12][13], they all focused on processing one data sample at a time, and none of them has provided an infrastructure for efficient investigation of a large data collection.

In this paper, we present FatFind, an interactive system that allows the user to search and quantitate adipocytes of different sizes in a large repository of cell microscopy images. The particular techniques for locating and measuring adipocytes are presented in Section 3.4. The infrastructure that enables this capability is Diamond, a distributed storage system that enables efficient interactive exploration of complex, non-indexed data. Diamond is described in Section 4.

Our approach represents a new method for quantitating adipocytes suspended in a drop under the microscope. Previous efforts to computerize the process have typically involved the use of expensive, proprietary imaging software (such as the Carl Zeiss KS 400) which functions a black box, rather than letting the user control the analysis process. Further, previous approaches have not addressed the need to screen hundreds of thousands of data samples simultaneously, nor do they allow interactive user control [9][10][14]. In contrast, the FatFind application and the Diamond platform on which it runs are open source software that specifically address these needs.

2. Data Collection

The two steps involved in the measurement of adipocyte cells are tissue preparation and cell measurement, described below.

2.1. Tissue preparation

Two types of methods exist for tissue preparation. The first uses sectioning methods to slice adipose tissue mass, fix it and then uses measurement tools to quantitate [10][15][16]. However, since a single section through the three dimensional tissue samples only one layer, the size of adipocytes in that layer may not be representative of the true adipocyte distribution. One way to address this issue would be to aggregate measurements over a large number of cells. The large count would normalize for the variation in adipocyte size at different layers. However, such a method may be unable to quantitate subtle differences in size as well as to resolve multiple size distributions in the population.

The second type of method involves isolation of adipocytes from the tissue and individual measurement of the cells. Standard methods have been developed using collagenase to separate adipocytes from adipose tissues [17][13]. After isolation, some methods have utilized osmium tetroxide as a fixative since adipocytes are delicate and lyse (burst) easily [16][18]. However, osmium tetroxide is extremely toxic and has been reported to cause cell swelling or cell clumping which may lead to anomalous counts [12]. Therefore, unfixed live cell methods are preferred and used in this work.

2.2. Cell measurement

The second step in measurement is the use of instrumentation to quantitate adipocyte size. This is complicated by certain characteristics of adipocytes -- they float, and they are extremely susceptible to lysis (bursting). Traditional methods of cellular size and volume analysis employ the Coulter counter, which measures the change in resistance of a 300 micron pore as the cells flow past, causing a momentary decrease in the pore volume [14]. However, with adipocytes, this method is susceptible to clogs, inaccurate measurement and significant lysis of the larger cells. Therefore, in this work, we prepared adipocytes from collagenase treatment of fat mass and then extracted adipocytes using previously mentioned methods [19]. A small drop of the live adipocyte suspension was then placed on a slide with a circular ridge of silicone grease. The cells typically floated to the top of the drop and could be observed. They were then photographed using a Nikon Diaphot microscope using a high resolution 14-megapixel Kodak DCS Pro14n digital camera. These images were then stored on servers for analysis.

3. FatFind

To assist the researcher with the task of adipocyte measurement, we have developed an application named FatFind. FatFind runs on the Diamond distributed search platform, which is described in Section 4. FatFind consists of a domain-specific front-end application that runs on the user's machine, and domain-specific search code, which runs on servers. The front-end application allows the user to specify adipocytes of interest, while the search code locates and quantitates adipocytes in large collections of digital cell images.

3.1. User Interface design concept

We intend FatFind to be a practical tool for adipocyte researchers. Keeping this audience in mind imposes the following design goals.

- *Interactivity.* Unlike the existing non-interactive systems [9][10][14], we intend to provide a tool that will engage the user and invite confidence in the results. Additionally, we would like to enable experimentation and exploration of the data collection.
- *Domain specificity.* FatFind is designed specifically for adipocyte research.
- *Flexibility.* Adipocyte classification can be subjective. For example, some cells may be out of focus, or lysed. A human uses domain knowledge to determine whether a particular structure is indeed an adipocyte. When viewing cells through the microscope, adjusting the focus knob provides additional information to construct a three-dimensional mental model.

Therefore, FatFind must give researchers the freedom to specify and adjust the way in which a search is conducted.

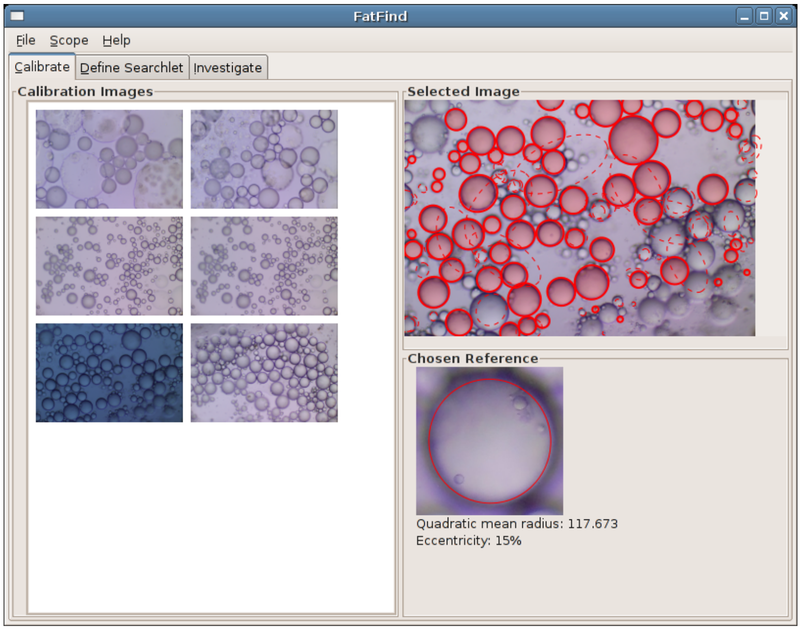
With these ideas in mind, and with an iterative process, we designed the following workflow for FatFind.

3.2. Workflow

The standard FatFind workflow consists of three steps, *Calibrate*, *Define Search*, and *Investigate*. These stages map to the tabs in the user interface.

3.2.1. Calibrate

In this step, shown in Fig. 1, the researcher starts with images from a small local collection. These images help to define a baseline for studying the adipocyte image collection. Upon selecting a calibration image from them, located in the left part of the window, FatFind runs an ellipse extraction algorithm to locate the adipocytes in the image. It displays the results in the upper-right part of the window. All of the detected ellipses are indicated as shaded circles. Ellipses that are considered too eccentric (and thus of low confidence) are shown with a dashed line. Clicking on an outlined ellipse displays information about its measurement, shown in the lower-right of the screen. Currently, FatFind displays quadratic mean radius and eccentricity. Individual adipocytes can be examined in this fashion. Once a final selection is made, the user moves on to the next tab, *Define Search.*



Quadratic mean radius: 117.673
Eccentricity: 15%

Fig. 1. Calibration phase.

3.2.2. Define search

After the reference adipocyte is chosen, the researcher begins defining the search as shown in Fig. 2. FatFind allows a researcher to specify four parameters: maximum radius, minimum radius, maximum eccentricity, and minimum sharpness. When a search is invoked, only images containing adipocytes matching all four parameters are returned.

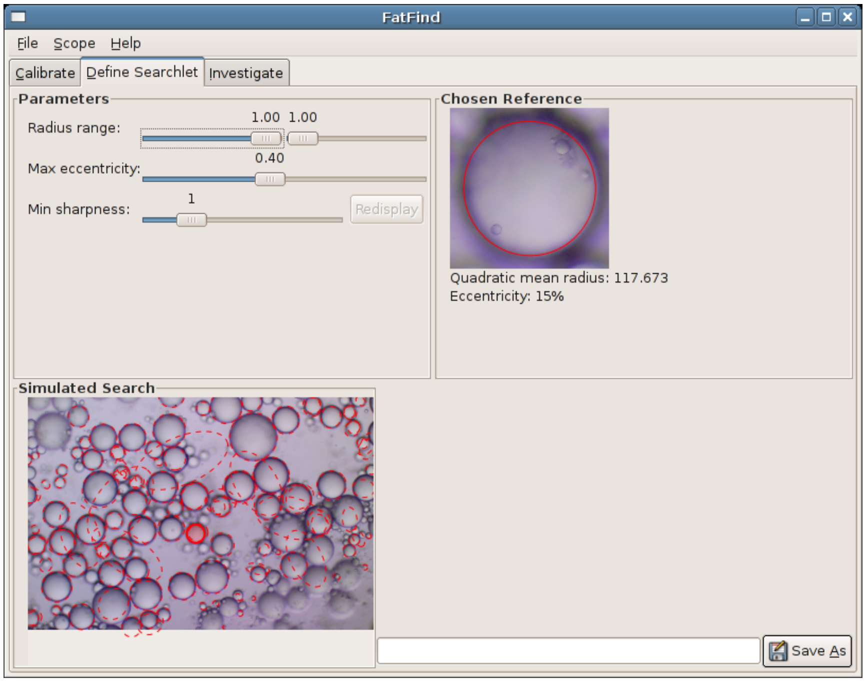
The design of FatFind avoids the use of absolute units. When defining a search, we represent radii as values relative to the radius of the chosen reference adipocyte. The sliders in the user interface work in this way.

To help in parameter selection, FatFind provides a small search preview (shown in the lower left of the window in Fig. 2), which illustrates matching adipocytes located in the calibration image. As the sliders are moved (Fig. 3 and Fig. 4), the search preview is updated in real time, providing immediate feedback about which adipocytes match the parameters specified. Users found this feature to be intuitive and helpful.

Once parameters are determined, the researcher saves the current search under a given name. Once one or more searches are defined, adipocyte investigation can begin.

3.3. Investigate

Now that one or more searches have been defined, the researcher can interactively search for matching adipocytes in the image repository. One can also make adjustments to the adipocyte extraction results, and compute statistics about them.



Radius range: 1.00 1.00

Max eccentricity: 0.40

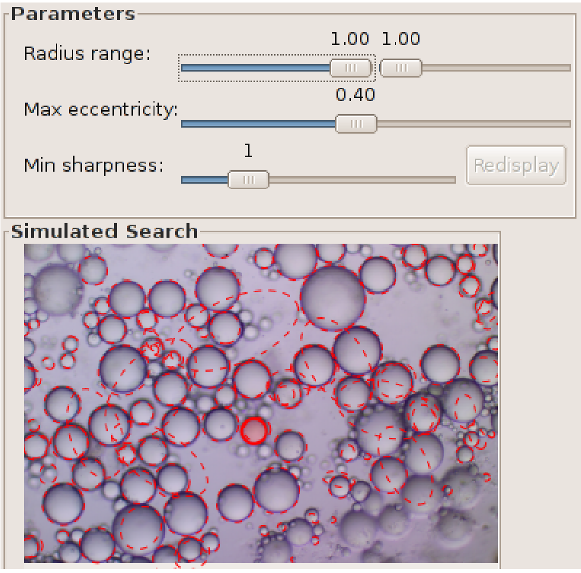
Min sharpness: 1

Quadratic mean radius: 117.673

Eccentricity: 15%

Fig. 2. Search definition phase.

Once a search is initiated, images with matching adipocytes will be displayed as they are processed. If the result appears incorrect, the researcher can stop the search, return to the previous step to adjust search parameters, and begin a new search. This interactivity and flexibility enables researchers to explore the data collection and experiment with it.

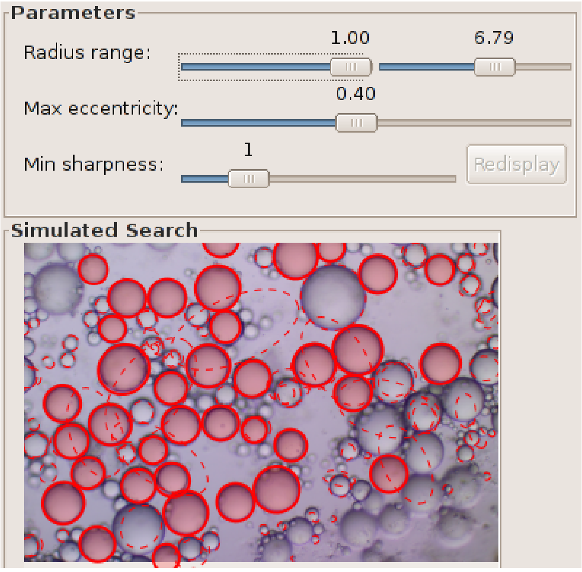


Radius range: 1.00 1.00

Max eccentricity: 0.40

Min sharpness: 1

Fig. 3. Preview before radius range adjustment.



Radius range: 1.00 6.79

Max eccentricity: 0.40

Min sharpness: 1

Fig. 4. Preview after radius range adjustment.

As results begin to appear, any thumbnail image can be selected and examined in more detail at the bottom of the window, as shown in Fig. 5. Within an image, all results are shown from the ellipse extraction algorithm, even those results that fall outside of the search specification. Ellipses falling within the search specification are filled, whereas ellipses outside the search specification are unfilled and drawn with a dashed line, matching the simulated search in the previous steps.

Keeping in mind the subjectivity of adipocyte classification, FatFind allows search results to be modified by the user in three ways.

1. Cells can be interactively defined using the mouse.
2. Cells can be deleted.
3. Cells can be toggled between the dashed and filled states, as shown in Fig. 6, to include or exclude them from the search results.

These modifications effectively allow the investigator to override the preliminary classification by FatFind. This interactivity gives researchers final control over the results of the study.

3.3.1. Derive statistics

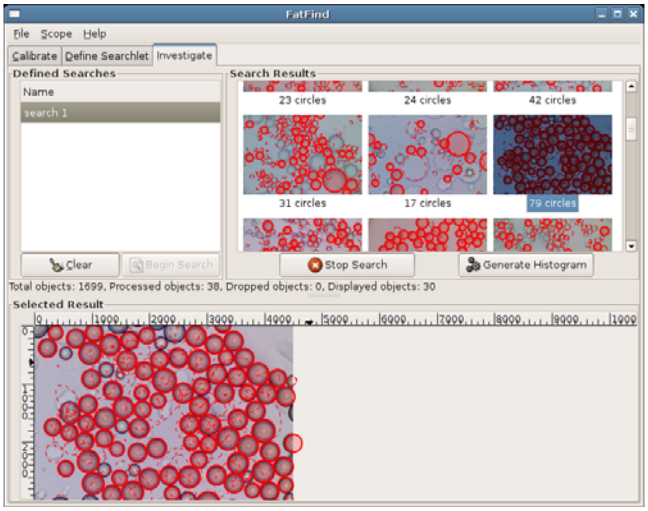
Once adipocytes have been found in the given image, FatFind can generate statistics, such as the histogram of detected adipocyte sizes. These quantitative measurements enable researchers to visualize changes in adipocyte distributions acquired from different sources. In the future, we plan to augment FatFind with more sophisticated quantitative analyses.

3.4. Adipocyte Detection Algorithms

Our approach to adipocyte detection exploits the fact that an adipocyte in aqueous suspension typically forms a circular shape. Thus, we focus on finding circular and elliptical objects in our digital cellular images, as detailed in this section.

3.4.1. Two-dimensional Hough transform

The initial implementation of FatFind used variants of the two-dimensional Hough transform [20] to identify circles in target images. This approach had two serious limitations. First, the transform could only find circles of a specified radius, and thus needed to be run hundreds or thousands of times per image to find circles of different radii. Second, the transform could not reliably detect adipocytes whose shape had been deformed from a circle, which is a common occurrence in practice.



Total objects: 1699, Processed objects: 38, Dropped objects: 0, Displayed objects: 30

Fig. 5. Investigation phase

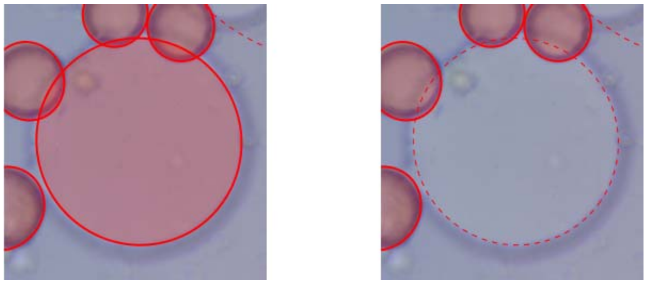


Fig. 6. Modifying search results

3.4.2. Fast ellipse extraction

The current implementation of FatFind employs a fast ellipse extraction (FEE) algorithm [21]. This technique is efficient and robust, and can locate overlapping and partially-occluded cells. We adopted an open-source implementation from LTI-Lib [22], and were impressed by the quality of the implementation and the level of technical support.

3.4.3. Edge Detection

The fast ellipse extraction algorithm requires binary edge images as input. To extract ellipses from our digital cell images, edge detection is a necessary first step. LTI-Lib includes an implementation of a Canny-like edge detector that uses color contrast gradients rather than grayscale contrast [23]. The color contrast gradient edge detector gave superior results over a standard Canny grayscale edge detector.

3.4.4. Multi-Resolution Processing

The fast ellipse extraction algorithm was initially designed to work on relatively low-resolution images such as those produced from sensors in robot navigation. Our digital cell images were acquired at a significantly higher resolution than those for which FEE was designed. In our images, FEE did not correctly detect large features, although small features were being correctly detected. Simply scaling the images down would allow the algorithm to find the large features, but would discard valid small features.

Therefore, we extended FEE to operate on a multi-scale image pyramid, where the original image was successively smoothed and sub-sampled. FEE was applied to each level of the pyramid and the results were subsequently merged

3.4.5. Overlap suppression

The ellipse merge operation has one potential problem. It is not sufficient to simply take a union of the extracted ellipses from all pyramid levels, because FEE can generate multiple detections for the same adipocyte. This problem is shown in Fig. 7.

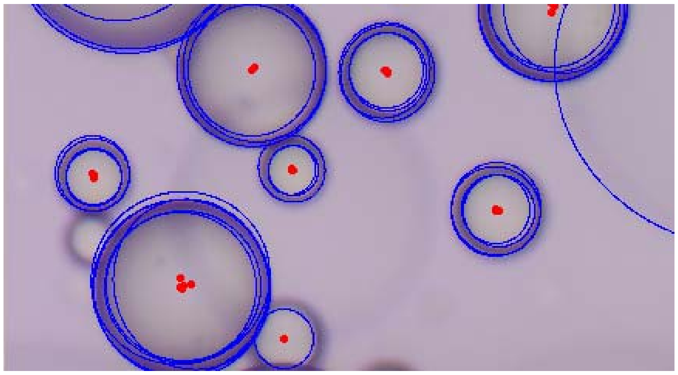


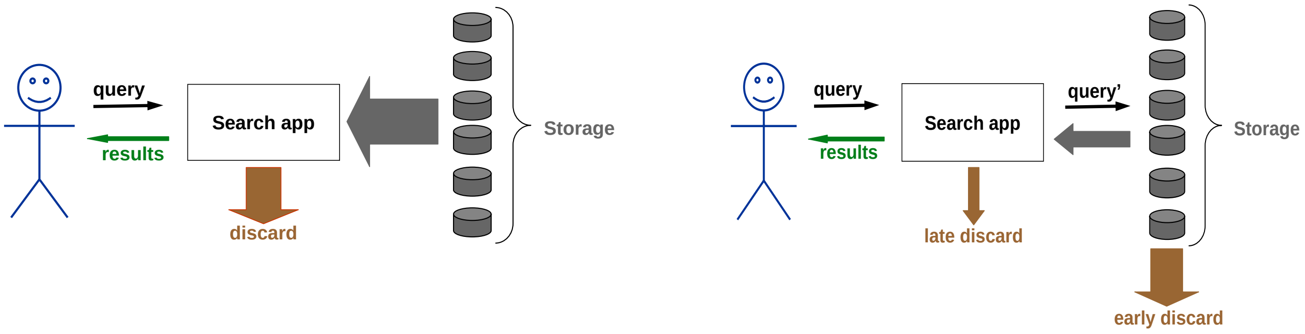
Fig. 7. Lack of overlap suppression.

To address this problem, we added an overlap suppression stage. This is a common post-processing technique in computer vision for object detection. Multiple detections with a high degree of overlap are assumed to correspond to a single adipocyte in the cellular image. Our current implementation for overlap suppression implicitly assumes that the detected adipocytes are ellipses with relatively low eccentricity (i.e., they are approximately circular). Should overlap suppression be required for scenarios where this assumption is violated, we could implement a more sophisticated overlap suppression scheme.

4. Diamond Search Platform

FatFind is an application built on the Diamond platform for interactive search. As mentioned earlier, Diamond is a distributed storage architecture that enables efficient interactive exploration of complex, non-indexed data. Such data frequently occurs in the form of images in the pharmaceutical and health care domains. When an index is not available, as is the case with rich data, brute force search is the only current option. Today, scanning a large volume of data is typically so slow that it is only performed in the context of well-planned data mining; only rarely is it attempted interactively. Diamond aims to improve the efficiency of brute force search so that an interactive approach becomes feasible. The key to achieving this efficiency is *early discard*.

4.1. Early Discard



(a) Without early discard

(b) With early discard

Fig. 8. Brute-force search

Fig. 8(a) illustrates the control and data flow in a typical brute-force search operation. Each data item passes through a pipeline from the disk surface, through the disk logic, over a local or network interconnect to the host computer. The search application can reject some of the data before presenting the rest to the user. There are two problems with this design. First, the system is unable to take full advantage of object-level parallelism at the storage nodes. Second, data must be shipped through the entire pipeline before being discarded in the final stages. This is undesirable because the huge volume of irrelevant data may clog the interconnect or host processor.

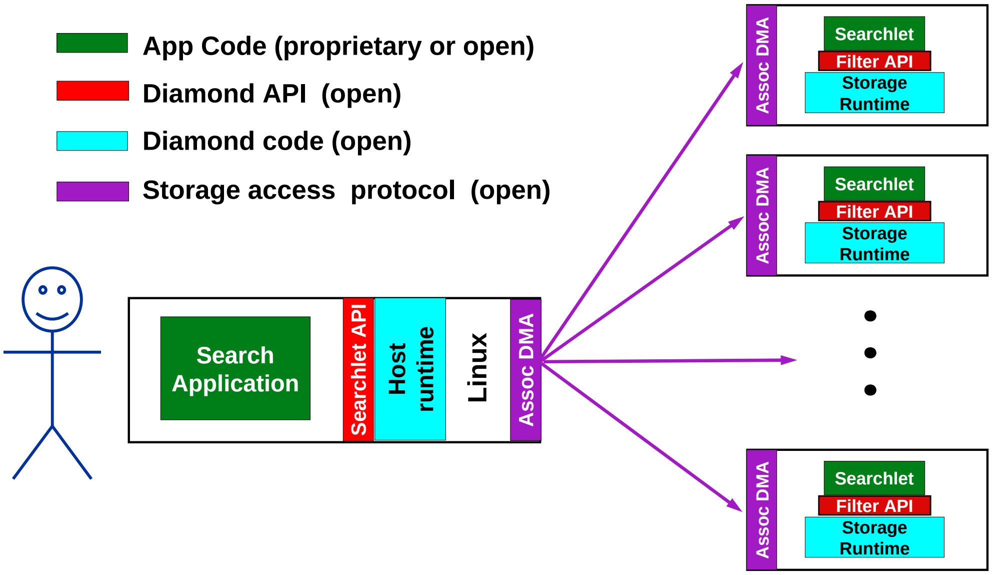
Early discard, shown in Fig. 8(b), is the idea of rejecting irrelevant data as early in the pipeline as possible. This improves scalability since it eliminates a large fraction of the data before it is sent over the interconnect. Since the knowledge needed to recognize irrelevant data is domain-specific, early discard requires application code to be executed close to storage.

Ideally, early discard would reject all of the irrelevant data without eliminating any of the desired data. This is impossible in practice for two reasons. First, the computational resources close to storage may be insufficient to perform all of the necessary (potentially expensive) application-specific computations. Second, there is a fundamental trade-off between false-positives (irrelevant data that is not rejected) and false-negatives (good data that is incorrectly discarded) [24]. The best one can do in practice is to tune an early discard algorithm to favor one at the expense of the other. Different applications may make different trade-offs in this space.

4.2. Diamond Architecture

As shown in Fig. 9, Diamond cleanly separates domain-specific application code from a domain-independent runtime system that underpins a wide range of search applications. Three lightweight application programming interfaces (APIs) define the external interfaces of Diamond.

The Searchlet API separates application code from Diamond code on a host system close to the user, typically a desktop. Above this API is the domain-specific graphical user interface (GUI) as well as domain-specific code to perform late discard (as shown in Fig. 8(b)). Through the Searchlet API, the application presents Diamond with a piece of code called a *searchlet* that is customized for the current query. The searchlet contains all of the domain-specific knowledge needed for early discard, and can be viewed as a proxy of the application that executes within the storage back-end. Diamond distributes the searchlet to each node in the storage back-end. Conceptually, each new query requires a new searchlet; in practice, many queries may only involve transmission of new searchlet parameters.



The FatFind user interface and application runs on the host system, in the box marked “Search Application”. The FatFind search code runs at the storage nodes, in the boxes marked “Searchlet”.

Fig. 9. Diamond architecture.

A typical searchlet is composed of stages called *filters*, each of which can independently discard objects. At each storage node, Diamond iterates through local objects in a system-determined order and presents them for evaluation through the Filter API. Diamond is completely ignorant of the details of this evaluation; all it cares about is the return value, which indicates whether to discard the object or to pass it on to the next filter. Only objects that pass through all filters in a searchlet are forwarded to the front-end. The Associative DMA API abstracts network transport and flow control, enabling Diamond searches that span storage nodes varying considerably in compute power, capacity, network connectivity and other performance characteristics.

4.3. Diamond and FatFind Interaction

FatFind follows the typical Diamond application model, which splits development into two parts: a domain-specific front-end client application, which runs on the user's machine; and a set of domain-specific filters, which run in a distributed fashion close to storage. For FatFind, we developed the front-end application from scratch and implemented a filter designed specifically for finding and categorizing circles and ellipses in high-resolution microscopy images.

Because FatFind is built on top of the Diamond infrastructure, FatFind shares a general workflow design with other Diamond applications. Roughly, the Diamond workflow has two parts: data storage and interactive search.

First, data is stored into a data storage system, at a layer somewhere below Diamond. Each backend storage node has access to a part of this data. From the perspective of Diamond, data in the storage system is accessible but immutable, and stored at a domain-specific granularity. For

FatFind, the data consists of individual high resolution cell microscopy images. Diamond works exactly at the level of granularity specified by the domain -- it can neither aggregate objects together, nor split objects into smaller pieces.

Second, a user interacts with a domain-specific front-end system to perform interactive search. Diamond takes advantage of the immutable nature of the data to parallelize execution and cache results, thus improving performance of iterative interactive search.

5. Evaluation

5.1. User Experience

From a user's point of view, FatFind enables her to:

1. Quickly quantitate adipocyte sizes.
2. Select specific sizes of adipocytes in an image or a group of images.
3. Search for adipocytes of a particular size and description from a large image database.

There were 3 major advantages in using Diamond as compared to the manual approach used earlier. First, this method enabled quickly finding the adipocyte size of interest which was difficult before. In the past, it could only be done in either of two ways, both of which are equally tedious. One way was to manually sift through the large image collection and visually estimate the distribution by eye. The other was to determine the distribution for every sample manually, enter the distributions into a spreadsheet or database, and then perform search and lookup. The second benefit of using Diamond was that it enabled direct and unbiased measurement of adipocytes automatically, as opposed to drawing a manual trace of each adipocyte and then measuring the trace. Finally, using Diamond allowed subpopulations of adipocytes to be detected and selected depending on the search parameters chosen.

Measuring adipocytes is complicated by a few issues. Large adipocytes are fragile which causes many of them to lyse during preparation and release lipids. Though every effort was made to remove these lipids, they tend to bind the cover slip or slide leading to artifacts that appear similar to adipocytes and can confuse the user or the program. In practice, FatFind was able to distinguish the lipid droplets from real adipocytes because the lipid droplet is too far out of the

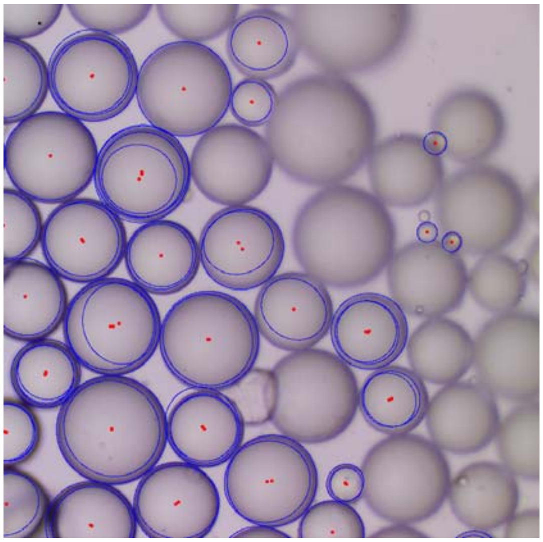


Fig. 10. Focal plane effect.

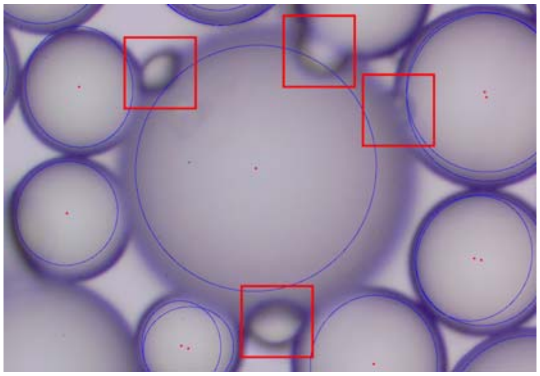


Fig. 11. Refraction effects.

focal plane for its edges to be detected. A human using a microscope typically distinguishes real adipocytes from lipid droplets by changing the focus. The focal plane selected for photography was a best guess by the investigator. In the resulting image, some adipocytes are in a different plane of focus and do not appear sharp, as shown in Fig. 10. Another factor complicating the issue was that adipocytes act like miniature lenses due to their almost perfectly round shape and almost complete transparency, as shown in Fig. 11. This phenomenon may be the reason why concentric rings within the adipocyte were observed, and had to be distinguished from the true border of the adipocyte.

Since the ellipse extraction algorithms relies solely on the quality of detected edges, the ability to find cells suffers significantly when edge detection is difficult. This is seen most often when a cell is significantly outside the focal plane of the microscope lens, or when anomalous refraction effects occur. In both cases, the cell visually appears to be “blurry” to the researcher’s eye. More work is needed in this area. Currently, FatFind uses a generic edge detector, so it is possible that sharpening or some simple domain-specific image processing may be enough to improve the range of detected cells. Any improvements to the underlying edge detection task will likely result in enhanced performance of the system as a whole. In general, FatFind reliably finds and quantitates adipocytes that are confirmed by the user to be intact adipocytes and not lipid droplets from other lysed adipocytes or other out of focus particles.

5.2. Performance

FatFind was designed to run on full-size images from the original adipocyte study. Each image contained 4500x3000 pixels, was stored in RGB color, and compressed using JPEG. Table 1 shows the time for our servers to decompress an image, build a pyramid, perform edge detection, find ellipses using the FEE algorithm, and eliminate overlapping circles. Even though the initial latency of a query was relatively high (typically 15 seconds), the overall throughput was also high, since we had eight machines running in parallel. Other than server parallelism, we did not explore any parallelism within a filter. Because of our use of image pyramids, it is likely that we could take advantage of multicore hardware to easily find a performance gain.

In our environment, the front-end runs on a machine with 1 GB of RAM, a 3.6 GHz Intel Pentium 4 processor, and a standard SATA hard disk. Eight back-end servers each have 4 GB of RAM, two 3.8 GHz Intel Pentium 4 Xeon processors, and multiple 10K RPM SCSI disks.

As an object passes through a searchlet, filters can append arbitrary attributes to it. These attributes are the primary mechanism of inter-filter communication. In FatFind, compressed objects are passed through a decoder filter that examines the compressed data and generates an uncompressed representation of the image. A current implementation limitation of Diamond is that filters can append but not remove attributes. As a result, a single 2MB JPEG image grows to over 55MB by the time it reaches the client. This does not pose an issue on our internal gigabit-speed network, but does cause problems on networks running at 100Mb/s or slower.

Operation	Time (ms)
Load and decompress	486.3
Pyramid build	327.0
Edge detection	7663.0
Fast ellipse extraction	2737.4

Table 1: Searchlet execution time.

Our future work involves the implementation of two complementary features: attribute garbage collection and a mechanism for a client to request attributes on demand. By eliminating the transfer of unnecessary attributes, a FatFind search would generate only a few hundred KB of network traffic. Only when a user requests would the system transmit the full information about a particular result.

6. Future Directions And Conclusion

The development of the FatFind application is a first step towards general browsing of microscopy image data sets. In the pharmaceutical and related industries, screening of compounds and biologics, numbering in the thousands to millions and more, is a significant area of focus. Screening is automated through the use of robotic and imaging machinery. Reactions of interest take place in small wells embedded within a plate. The number of wells may vary from less than 100 to several thousand. There are several types of screens currently used. Among these screens are High Content Screening (HCS) and High Throughput Screening (HTS). HTS performs reactions at the molecular level, while HCS highlights cellular response. Often HCS is associated with functional proteomics and genomics. HCS may be contrasted with HTS by examining the amount of data and computation derived and applied, respectively, to each well. HTS may yield one to several hundred data points from a single readout or time series of readouts.

As seen with the FatFind application, raw HCS data are typically 2-D or 3-D images. As “a picture is worth a thousand words” an HCS image may contain megabytes of information or “millions of words”. While a few hundred arithmetic operations may be used to fit a curve in an HTS well, the image processing required to interactively extract information from HCS data is shown in this article to greatly benefit from parallel computation.

Typical HCS software is designed for slow batch processing of specific assay protocols. Interactive browsing of the images is limited to the previously computed characteristics of these same images. Browsing and mining numerical HTS data has reached a high level of sophistication with many commercial platforms available. Because HCS machinery has evolved in a decentralized and dedicated fashion, browsing the raw data, fusing the information with other data sources and mining the resulting aggregation was not seen as feasible. The Diamond platform provides a ready data mining and browsing framework as demonstrated above with adipocyte data and FatFind.

In order to move towards more sophisticated browsing and analysis capabilities for HCS, future directions may involve building descriptors of other cellular characteristics such as cellular events, nucleic perturbations as well as changes in the cell cytoplasm. As a large collection of descriptors and experience with them evolve, it will be possible to build a general, interactive browsing capability like SnapFind [25]. Here, queries by example may be built allowing users to select an image or part of an image. This is used to retrieve images with similar features. Iterative improvement of the query may be performed by looking at a small return set and giving the user the opportunity to train the query through accepting or rejecting each returned image. This feedback is then used to sharpen the query.

Developing an image data mining platform for HCS on par with the speed of HTS is now possible. However, the significantly higher data density contained within an HCS platform has the potential to create new information and knowledge that may be much greater than HTS.

References

[1] G. Ailhaud, “Adipose tissue as a secretory organ: from adipogenesis to the metabolic syndrome,” *C R Biol* 2006; 329; (8), 570-7; discussion 653-5.

[2] L. Laviola, S. Perrini, A Cignarelli, and F. Giorgino, “Insulin signalling in human adipose tissue,” *Arch Physiol Biochem* 2006; 112; (2), 82-8.

[3] H. Sell, J. Eckel, and D. Dietze-Schroeder, “Pathways leading to muscle insulin resistance--the muscle--fat connection,” *Arch Physiol Biochem* 2006; 112; (2), 105-13.

[4] Anon. “Adipocyte sizes and numbers in lean and obese subjects,” *Nutr Rev* 1984; 42; (4), 159-60.

[5] C. Couillard, P. Mauriege, P. Imbeault, D. Prud'homme, A. Nadeau, A. Tremblay, C. Bouchard, and J.P. Despres, “Hyperleptinemia is more closely associated with adipose cell hypertrophy than with adipose tissue hyperplasia,” *Int J Obes Relat Metab Disord* 2000; 24; (6), 782-8.

[6] P. Imbeault, S. Lemieux, D. Prud'homme, A. Tremblay, A. Nadeau, J.P. Despres, and P. Mauriege, “Relationship of visceral adipose tissue to metabolic risk factors for coronary heart disease: is there a contribution of subcutaneous fat cell hypertrophy?” *Metabolism* 1999; 48; (3), 355-62.

[7] J. Smith, M. Al-Amri, P. Dorairaj, and A. Sniderman, “The adipocyte life cycle hypothesis,” *Clin Sci* (Lond) 2006; 110; (1), 1-9.

[8] V. van Harmelen, T. Skurk, K. Rohrig, Y.M. Lee, M. Halbleib, I. Aprath-Husmann, and H. Hauner, “Effect of BMI and age on adipose tissue cellularity and differentiation capacity in women,” *Int J Obes Relat Metab Disord* 2003; 27; (8), 889-95.

[9] T. Bjornheden, B. Jakubowicz, M. Levin, B. Oden, S. Eden, L. Sjostrom, and M. Lonn, “Computerized determination of adipocyte size,” *Obes Res* 2004; 12; (1), 95-105.

[10] H.C. Chen, and R.V. Farese, Jr., “Determination of adipocyte size by computer image analysis,” *J Lipid Res* 2002; 43; (6), 986-9.

[11] M. Di Girolamo, S. Mendlinger, and J.W. Fertig, “A simple method to determine fat cell size and number in four mammalian species”, *Am J Physiol* 1971; 221; (3), 850-8.

[12] J. Hirsch, and E. Gallian, “Methods for the determination of adipose cell size in man and animals,” *J Lipid Res* 1968; 9; (1), 110-9.

[13] Y.D. Tchoukalova, D.A. Harteneck, R.A. Karwoski, J. Tarara, and M.D. Jensen, “A quick, reliable, and automated method for fat cell sizing,” *J Lipid Res* 2003; 44; (9), 1795-801

[14] B.J. Maroni, R. Haesemeyer, L.K. Wilson, and M. DiGirolamo, “Electronic determination of size and number in isolated unfixed adipocyte populations,” *J Lipid Res* 1990; 31; (9), 1703-9.

[15] D. Lemonnier, “Effect of age, sex, and sites on the cellularity of the adipose tissue in mice and rats rendered obese by a high-fat diet,” *J Clin Invest* 1972; 51; (11), 2907-15.

[16] L. Sjostrom, P. Bjorntorp, and J. Vrana, “Microscopic fat cell size measurements on frozen-cut adipose tissue in comparison with automatic determinations of osmium-fixed fat cells,” *J Lipid Res* 1971; 12; (5), 521-30.

[17] A.H. Bakker, F.M. Van Dielen, J.W. Greve, J.A. Adam, and W.A. Buurman, “Preadipocyte number in omental and subcutaneous adipose tissue of obese individuals,” *Obes Res* 2004; 12; (3), 488-98.

[18] M.P. Stern and F. Conrad, “An automated, direct method for measuring adipocyte cell size,” *Clin Chim Acta* 1975; 65; (1), 29-37.

[19] S.K. Fried and N. Moustaid-Moussa, “Culture of adipose tissue and isolated adipocytes,” *Methods Mol Biol.* 2001;155:197-212.

[20] R. C. Gonzales and R. E. Woods: Digital Image Processing. Prentice Hall, New Jersey 2002

[21] E. Kim, M. Haseyama, H. Kitajima. “Fast and Robust Ellipse Extraction from Complicated Images,” *Proceedings of the IEEE International Conference on Information Technology and Applications*, 2002.

[22] Lehrstuhl für Technische Informatik, Aachen University of Technology. http://ltilib.sourceforge.net/doc/html/classlti_1_1fastEllipseExtraction.html.

[23] A. Cumani, "Edge Detection in Multispectral Images", Technical Report, Istituto Elettrotecnico Nazionale "Galileo Ferraris", 1989

[24] R. Duda, P. Hart, and D. Stork, *Pattern Classification*, Wiley, 2001.

[25] L. Huston, R. Sukthankar, D. Hoiem, and J. Zhang. SnapFind: Brute Force Interactive Image Retrieval. *Proceedings of International Conference on Image Processing and Graphics*, 2004.