

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[‡]
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.

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. Significant alteration in body mass involves alterations in both adipocyte volume and number. Adipocyte-derived factors are significantly increased in obesity and represent good predictors of the development of type 2 diabetes [1][2]. Moreover, the increase in fat mass has been strongly correlated to the size of the adipocytes, especially in females [3][4]. 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 [5][6][7], 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 infrastructure that enables this capability is Diamond, a distributed storage system that enables efficient interactive exploration of complex, non-indexed data, detailed in Section 4; the particular techniques for locating and measuring adipocytes are presented in Section 0.

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 [5][6][8]. In contrast, both the FatFind application and the

Diamond platform on which it runs are open source software that specifically address the previously mentioned needs.

2. DATA COLLECTION

In this work, we prepared adipocytes from collagenase treatment of fat mass. A small drop of the live adipocyte suspension was 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 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.

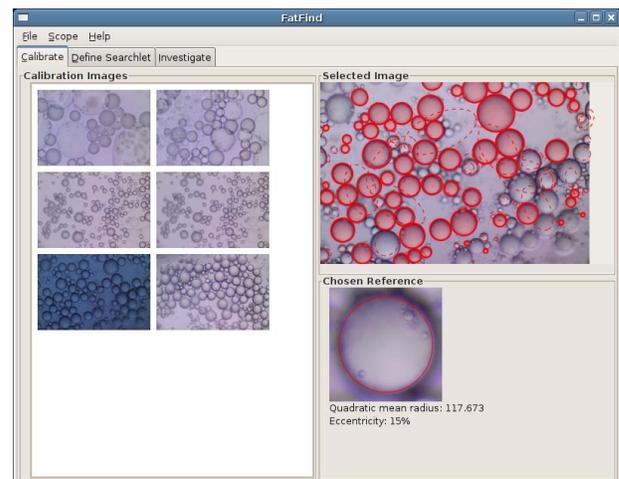


Fig. 1. Calibration phase.

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 [5][6][8], 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. FatFind must give researchers the freedom to specify and adjust the way in which a search is conducted.

Workflow

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

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*.

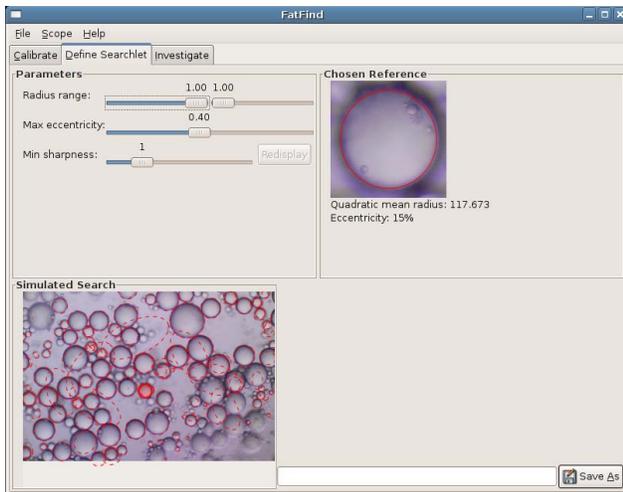


Fig. 3. Search definition phase.

Define search

After the reference adipocyte is chosen, the researcher begins defining the search as shown in Fig. 3. 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. 3), which illustrates matching adipocytes located in the calibration image. As the sliders are moved (Fig. 2), 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.

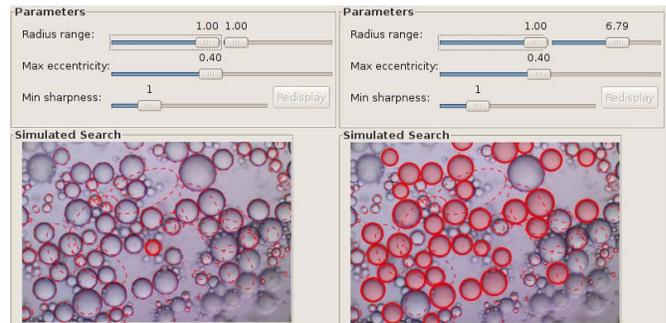


Fig. 2. Search preview with radius range adjustment.

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.

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.

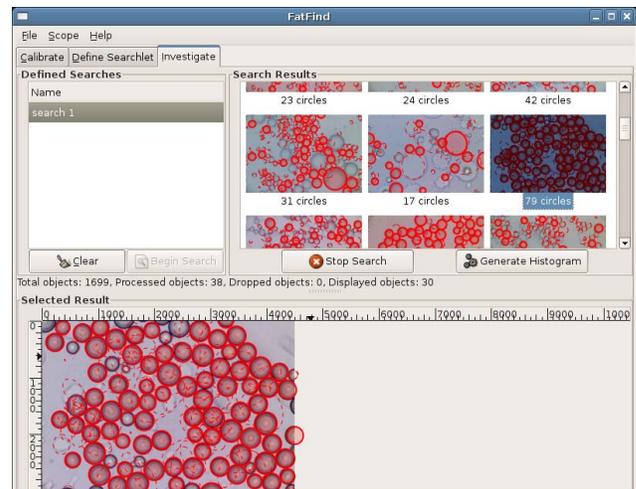


Fig. 4. Investigation phase.

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.

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. 4. 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.

- Cells can be interactively defined using the mouse.
- Cells can be deleted.
- Cells can be toggled between the dashed and filled states, 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.

Derive statistics

Once adipocytes have been found in the given image, FatFind can generate statistics, such as the histogram of detected adipocyte sizes. Quantitative measurements enable researchers to visualize changes in adipocyte distributions acquired from different sources.

Adipocyte Detection Algorithm

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.

Fast ellipse extraction

The current implementation of FatFind employs a fast ellipse extraction (FEE) algorithm [9]. This technique is efficient and robust, and can locate overlapping and partially-occluded cells. We adopted an open-source implementation from LTI-Lib [10].

Multi-Resolution Processing

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 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.

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. To address this problem, we added an overlap suppression stage, where multiple detections with a high degree of overlap are assumed to correspond to a single eclipse.

4. DIAMOND

FatFind is an application built on the Diamond platform for interactive search. As mentioned in the Introduction, Diamond is a distributed storage architecture that enables efficient interactive exploration of complex, non-indexed data. The key to achieving this efficiency is *early discard*.

Early Discard

Fig. 5 top illustrates the control and data flow in a typical brute-force search operation. 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. 5 bottom, 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.

Diamond Architecture

As shown in Fig. 6, Diamond cleanly separates domain-specific application code from a domain-independent runtime system that underpins a wide range of search applications. Three lightweight

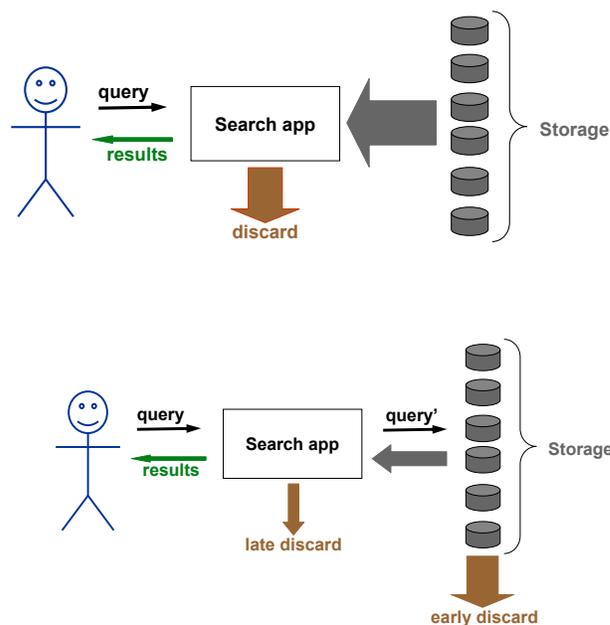


Fig. 5. Brute-force search

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. 5 bottom). 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. Only objects that pass through 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.

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 for finding and categorizing ellipses, which run in a distributed fashion close to storage. Diamond parallelizes execution and caches results, thus improving performance of iterative interactive search.

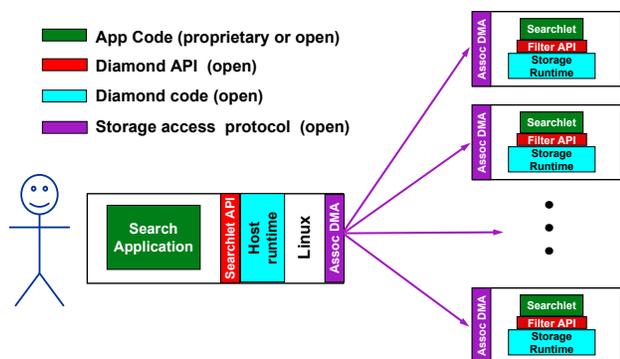


Fig. 6. Diamond architecture. The FatFind user interface runs on the host system, in the box marked “Search Application”. The FatFind search code runs at the storage nodes, in the boxes marked “Searchlet”.

5. EVALUATION

User Experience

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

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

There are 3 major advantages in using Diamond. First, it enables efficient finding of the adipocyte size of interest. In the past, it could only be done in either of two equally tedious ways. One was to manually sift through the large image collection and visually estimate the distribution. The other was to determine the distribution for every sample manually, enter it into a spreadsheet or database, and then perform search and lookup. The second benefit of using Diamond is that it enabled direct and unbiased measurement 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.

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 multiple hardware threads 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.

6. FUTURE DIRECTIONS

The development of the FatFind application is a first step towards general browsing of microscopy 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. The Diamond platform provides a ready data mining and browsing framework as demonstrated above with adipocyte data and FatFind. 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 [11]. 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.

Table 1. FatFind Searchlet execution time.

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

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] Anon. “Adipocyte sizes and numbers in lean and obese subjects,” *Nutr Rev* 1984; 42; (4), 159-60.
- [4] 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.
- [5] 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.
- [6] H.C. Chen, and R.V. Farese, Jr., “Determination of adipocyte size by computer image analysis,” *J Lipid Res* 2002; 43; (6), 986-9.
- [7] 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
- [8] 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.
- [9] 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.
- [10] Lehrstuhl für Technische Informatik, Aachen University of Technology. http://tlib.sourceforge.net/doc/html/classlti_1_1fastEllipseExtraction.html.
- [11] 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.
- [12] A. Goode, M. Chen, A. Tarachandani, L. Mummert, R. Sukthankar, C. Helfrich, A. Stefanni, L. Fix, J. Saltzman, M. Satyanarayanan, “Interactive search of adipocytes in large collections of digital cellular images”, CMU-CS-06-177.