Stereo Image Capture and Interest Point Correlation for 3D Modeling

Andrew Crocker, Eileen King, and Tommy Markley
Department of Math, Statistics, and Computer Science
St. Olaf College
1500 St. Olaf Avenue, Northfield MN 55057
crocker@stolaf.edu, kinge@stolaf.edu, markley@stolaf.edu

Abstract

In modeling St. Olaf's Regents Hall of Natural Sciences, 3500 pairs of images were collected with two identical Canon Rebel cameras with 18mm lenses and 25-35 percent overlap between pairs. Quality of the images was improved by the use of a tripod and precision mount for stability and consistency, a small aperture and long shutter speed to correct depth of field blur, and a two-second shutter delay to keep the cameras stable during capture. Modified functions from the OpenSURF library were used to locate corresponding points between two images and indices of the matches were compared to transitively matching points across two *pairs* of images. A least-squares minimization (beginning with a guess at camera location from the building's blueprints) can then be used to successively approximate camera location using these correspondence points, information about camera calibration, and the physical location of three anchor points relative to a chosen origin.

1 Image Collection, Storage, and Processing

This document details the work done by Team BigFish on the January 2012 continuation of the St. Olaf College computer science department's ongoing work in 3D vision and modeling. This year's project was the beginning of a 3D model of the interior of the College's Regents Hall of Natural Sciences, a 200,000-square-foot building of four floors; Team BigFish's work was in was the capturing and handling of image data from strategic locations throughout Regents Hall. This work can be further broken down into image collection, storage, and processing.

1.1 Image Collection

1.1.1 Equipment and Setup

All photos were taken with a pair of Canon EOS Rebel T1i cameras. These two 15-megapixel units came equipped with standard 18-55 mm Canon lenses, set to 18 mm to ensure the widest angle possible for optimal area coverage. The cameras were shooting in RAW to minimize data loss through JPEG compression; all auto features with the exception of auto focus were switched off, and all other camera settings were set to be the same between the cameras. With identical cameras, images could be captured that were also identical except for the shift that occurs between two images of a stereo pair.

The crux of data collection was the simultaneous capture of these images to make a pair. Some obstacles in taking capturing useful data such as bad lighting and light reflection were difficult (if not impossible) to avoid, but depth of field blur was a problem with a solution. To overcome the blur generated by an out-of-focus image, the aperture on the cameras was set to the maximum value of f22. To accommodate the proportionally small amount of light let in by such a small aperture, shutter speed was set for relatively long exposure times: two to four seconds on average, depending on the lighting conditions, with some exposures of up to twenty seconds in dimly lit areas. For such long exposures, it became very important that the cameras remained steady during the capture process to avoid unnecessary motion blur.

Through use of a heavy duty tripod, cameras were kept immobile during the capture period. As an added precaution, pictures were taken with a two-second time delay, which ensured that the photographer’s hands pressing buttons didn’t affect the camera’s steadiness. In order to maintain a constant orientation relative to one another, the cameras were securely mounted side by side on a sturdy metal bar that was in turn fastened to the top of the tripod (Figure 1). This was all set up on a daily basis as charging the camera batteries necessitated disassembling the rig, charging, reassembling, and repeating the calibration process to get accurate camera orientation information. The assembly could then be used for data collection, which involved gathering image pairs as well as making an estimate of the camera position inside of the global artificial coordinate system (to be later narrowed down to a much greater degree of accuracy using least-squares

minimization software developed for this process).

Figure 1: The camera rig, consisting of a heavy-duty tripod and Jasper Engineering precision mounting bar bearing two Canon EOS Rebel T1i 15-megapixel cameras.

1.1.2 Coverage

This minimization process required an estimate of the camera's location accurate to within two meters, relative to the geometric precision and calibration team's chosen origin: the lower-right corner of the second-floor fishtank (see Figure 3). Blueprints for the building proved to be an invaluable tool in this process. The floor plans used a coordinate system based on the concrete support pillars placed throughout the building, and data collection was planned around the location of these reference points. Since the blueprints gave detailed measurements of every part of the building and each page of the document was given a specific scale, it was possible to accurately calculate the location of the cameras by positioning the tripod strategically within the coordinate system mapped on the plans.

Images were then methodically gathered of the hallways, atria, and study areas on all four floors of the building. Each camera position required between 8 and 20 image pairs to achieve the desired coverage of the area visible from that point; photos had to have sufficient overlap to ensure that corresponding points could be found, so overlap from one image to the next is approximately one-fourth to one-third.

1.2 Data Storage

As the SD cards in the cameras filled up, images had to be moved to a file system accessible to other teams. To this end, they were renamed, converted from .CR2 format to the much more useful .ppm file type, and grouped by floor and then by date of capture (to ensure that they would be easily matched with the correct calibration data). These 3500 pairs of images were captured over a period of nine days, with most of the data collection taking place in the final seven days. Due to a lack of time for taking images from specific locations and measuring those positions accurately, there were very small parts of the building such as corners and occluded regions behind furniture that were not included in the data set. This data is being used in the second-semester senior capstone class and will no doubt be used in future department work in computer vision.

1.3 Image Processing

After collecting all the data, our secondary task was to process the images and pull important values from them. These values came in the form of corresponding points between the images, and accurate camera locations for each of the photos. Once we had this information, we could pass it to other groups to help in the creation of a 3D model of Regents Hall. However, it was the automated detection of these corresponding points which was the focus of the second half of our part in the project.

1.3.1 SURF

Detection of matching points between images was done using functions from an open-source interest point detection library called OpenSURF (Speeded Up Robust Features). SURF achieves its eponymous speed boost in two key ways: the use of integral images and a reversed approach to the construction of scale-spaces. The integral image is a representation wherein the value of any particular pixel is the sum of the intensities (black and white) of the pixels above and to its left, inclusive; it can be relatively quickly calculated and its use speeds up later operations by allowing the area of any rectangle to be calculated with only four array accesses [1].

The second point is the truly revolutionary aspect of the SURF method: rather than creating a scale-space in the typical fashion (convolving the image with the Gaussian kernel, resizing, and repeating), the SURF approach is to instead resize a box filter approximating the Gaussian (figure 2). Enlarging a filter takes significantly less time than shrinking an image (with negligible loss of accuracy in the results), especially considering that scales don't depend on each other and could therefore be run in parallel for even further improvement [1].

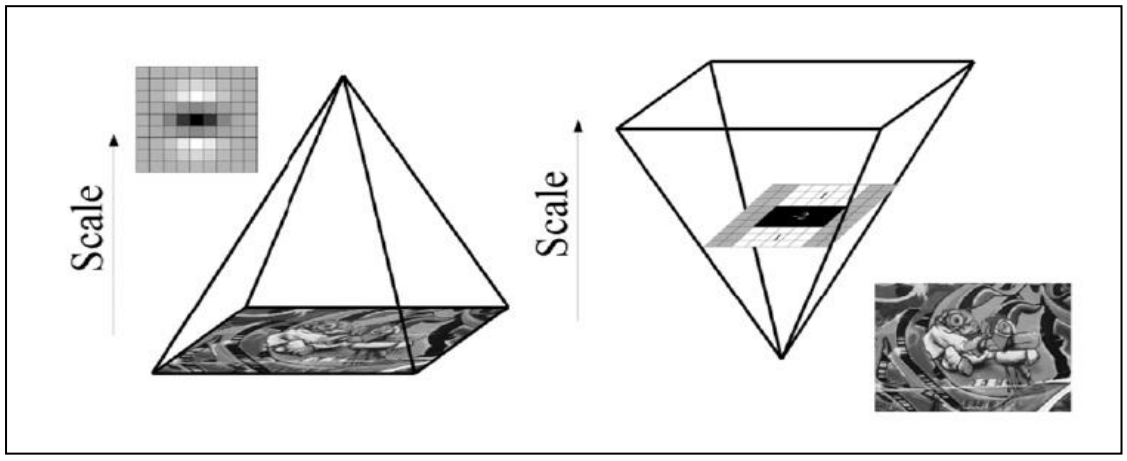

Figure 2: A distinguishing feature of SURF is that it resizes its filter rather than the image, which requires far fewer steps in computation [1].

1.3.2 Eriol

Creation of correspondence files began with an image viewing program called Eriol, a code base developed in previous years. Key features of this program included the ability to display two different images in two windows, to zoom in and out quickly, and to mark small colored points on the images. An early attempt at location of corresponding points involved marking them by hand using different letters of the alphabet to create points in different colors; however, this was too tedious and imprecise to be practical with a data set of this size.

1.3.3 Creating an Integrated Interface

Integration of the functionality of the OpenSURF library created the ability to automatically find (and draw in different colors) matching points between two images; SURF's matching points were stored in a vector to allow removal of points determined visually to not be good matches (warped due to being through or reflected on glass, "matching" points on visual corners created by obstruction rather than objects that actually intersect, points that are the right part of, for example, the wrong ceiling tile, etc.). Additionally, rather than having a fixed ratio of comparison between points to determine matches, a loop was added to this part of the process to start with a low ratio (strong matches) and slowly increase until the desired number of matches had been found, ensuring that images were linked only by the strongest correspondences.

Finally, work was begun on finding matches not just between two images but between two *pairs* of images. The first step in the matching process is the detection (and storage in respective vectors) of interest points in two images; next, pairs of the indices of matching points are stored in additional vectors. These vector indices allow quick matching of points through transitive comparison of integers rather than a complicated process of trying to compare points in more than two images at a time. We then find up to ten points

that match across all four images (four is the minimum number required to be able to precisely locate the position of the cameras), and these points are displayed so that they can be checked before being written into a correspondence file.

Figure 3: Automatically-detected matches across two pairs of images of the same object taken at different angles. Dots of the same color in each image represent matching points and are connected by white lines.

1.3.4 Results and Future Plans

The most pressing need in making this program a truly valuable tool is increasing the speed; finding matching points between four full-size images as in figure 3 generally takes between one and three minutes. Parallelizing the code would go a long way toward speeding up the process. Also useful would be the ability to read and display points from a correspondence file so that a person could run large batches and return to simply visually check them later. Finally, exploration needs to be done of creating tools to help keep track – over vast numbers of images – of which images have already been matched and which images are likely to be of the same area but from different angles.

Regarding image collection, future plans would include restructuring the collection process to make processing more efficient. Images were captured facing in all directions from a given location, which took less time to capture but more time to organize and process. Location of correspondence points could be done more quickly if long sequences of images all faced the same direction. Development of tools to better keep track of the approximate camera location – perhaps by triangulating signals, or even by simply being able to mark the location on an electronic copy of the blueprint – would also eliminate a major source of frustration and potential human error.

References

[1] C. Evans. Notes on the OpenSURF library. CSTR-09-001, University of Bristol. January 2009.