Correspondence Matching for the LUMIS2 Camera System

Tyler O'Neil
tmoneil@ucsd.edu

Abstract

The UCSD Computer Vision Group along with the Scripps Institute of Oceanography developed a multispectral imaging system for monitoring the health of coral reef called LUMIS2. As a small part of this project, image registration was employed for visualizing and interpreting the massive amount of data produced by the unique system. This report is an overview of one approach to registering images of different spectra taken of a scene with large depth variation using techniques borrowed from optical flow and relatively simple image hacks like histogram matching.

1 Introduction

The sophistication of image registration is frequently hard to conceptualize because humans effortlessly match corresponding points between the images from their eyes; however, the task is non-trivial and complicated by a number of issues. First of all, distortions introduced by the imaging system like lens imperfection and refraction can make two images of the same scene appear distorted. Additionally, lighting inconsistencies like specularities and fluorescence (in the case of a multispectral system) can make regions appear bright in one image and dark in the other, and therefore complicate correspondence matching. Finally, scenes with large depth variations can have occluded regions that are only visible in one image but not the other. The human visual system subconsciously corrects for all these difficult problems, but any algorithm needs to take them into account.

The basic flow of image registration is often as follows: optionally correct for image distortions, perform some level of normalization in order to make both images appear to have the same brightness, find correspondences between the two scenes, use the correspondences to define a transformation from one image to the other, and finally apply the transformation. All of these stages have application-specific solutions, so it was necessary to take each into consideration for the LUMIS2 system. For instance for the first step of correcting distortions, the LUMIS2's underwater environment introduces distortions due to refraction in addition to normal lens distortions. Correcting for this distortion will make registration easier and the resulting stack of registered images more representative of the actual scene. Additionally for the second step of reconciling brightness inconsistencies, image registration for LUMIS2 will have to deal with coral fluorescence at different frequencies. Fluorescence and the lack of it can make two scenes appear very different, and we'll need to consider it in order to find correspondences between the two images. For finding correspondences and defining a transform from one image to the other, occlusions will be severe in our scenes of close up shots of coral. Any correspondence-finding algorithm needs to be able to deal with large patches that are not visible in the other image. The rest of this paper is dedicated to overcoming these obstacles.

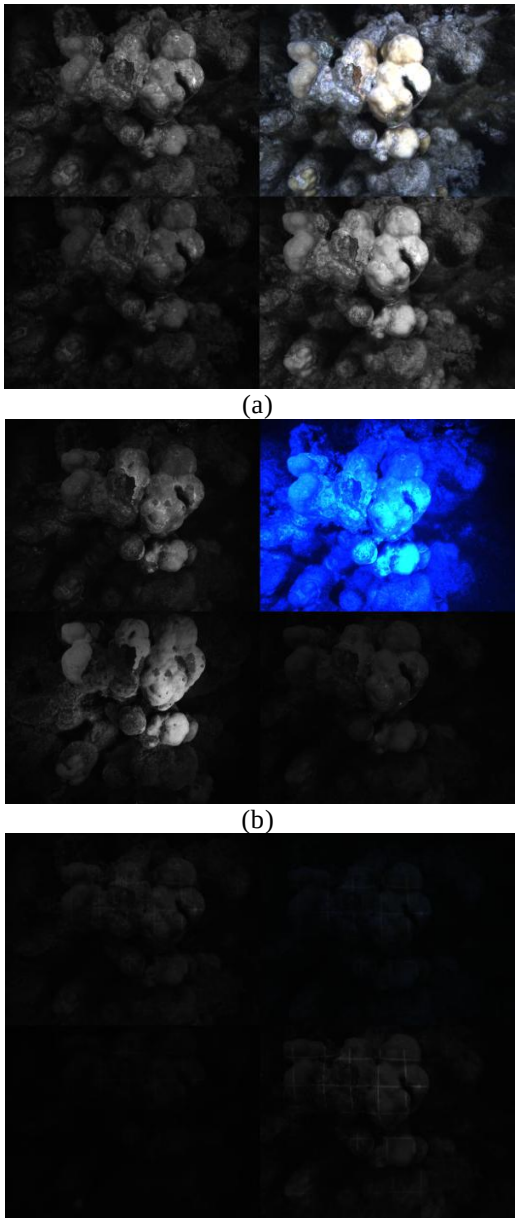


Figure 1: Images taken from by LUMIS2 with white light (a), blue light (b), and only pattern (c).

The final objective is an algorithm that will take the three filtered images under blue light and register them with the white light non-filtered image. In figure 1, this would correspond to taking the upper left, lower left, and lower right of (b) and matching it with the upper right of (a). Additionally, this system needs register images as fast as possible in order to scale to data sets of hundreds of high resolution images.

2 Background on LUMIS2

The LUMIS2 is an underwater camera system consisting of a projector and four cameras operating at different frequencies housed behind a thick translucent polymer. Each camera operates at a different spectrum: green (510-530nm), no color filter, red (665-695nm), and orange (568-588nm). The system takes 3 image sets at a time: one with white light, one with blue light, and one with only light from the projector. The images are taken

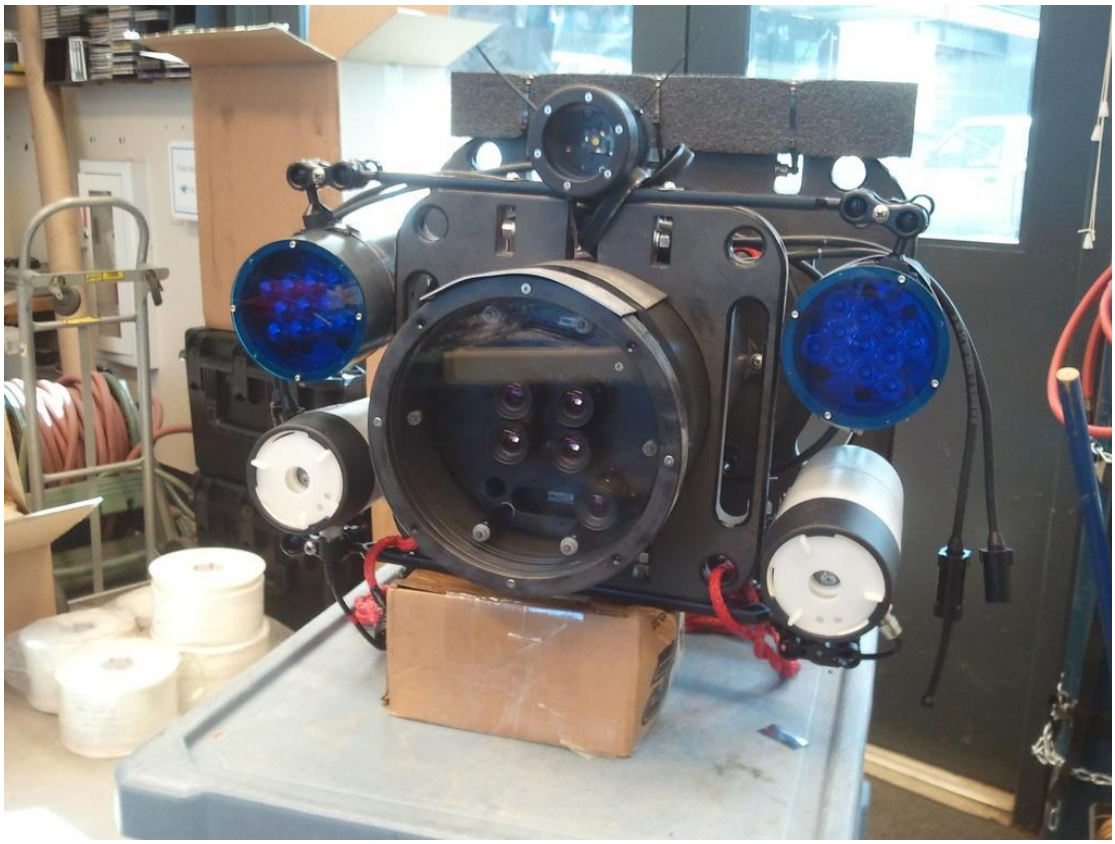


Figure 2: LUMIS2

120 milliseconds apart for a total of 12 images. The Field of view is 42 degrees in the horizontal and 33 degrees in the vertial. See Figure 1 for an example sequence.

3 Compensating for Distortions

There are two main categories of distortion: transversal distortion which is only a function of a point's radial coordinate, and non-transversal distortion which is a function of the radial coordinate and the 3D location of the point being projected [1]. Traditionally, correcting distortion only deals with transversal distortion as a result of imperfect lens curvature. The non-linear transform of the pixels is usually represented with the following formulas [2]:

$$x_{corrected} = x(1 + k_1r^2 + k_2r^4 \ldots.)$$

$$y_{corrected} = y(1 + k_1r^2 + k_2r^4 \ldots.)$$

where r is the radial coordinate and k_n is a property of the lens found through camera calibration. LUMIS2's imperfect lenses should be corrected with this method to make registration results reflect depth (and possibly assist registration).

In addition, the underwater system also has distortion caused by refraction that should be corrected. From snell's law:

$$n_1 \sin\theta_1 = n_2 \sin\theta_2$$

where n_1 and n_2 are functions of the materials on either side of the interface (water and air in

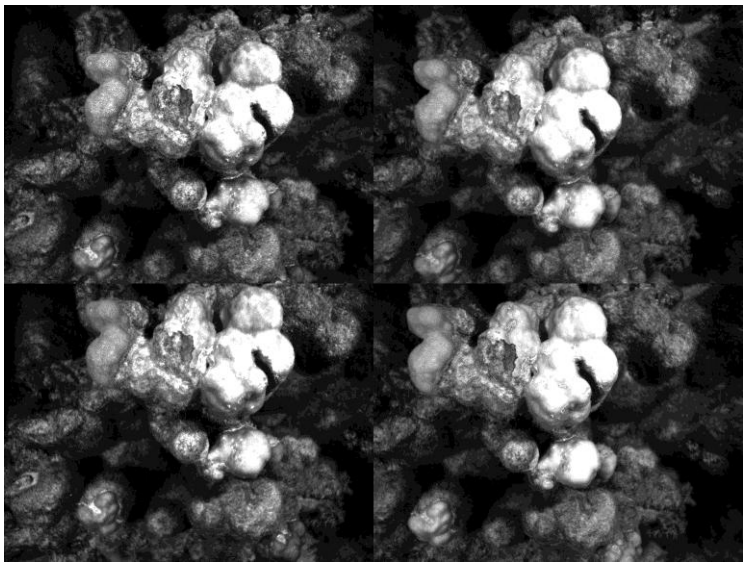


Figure 3: Result of histogram matching from Figure 1a.

our case), and θ_1 and θ_2 are the incoming and outgoing angles. So as an object is moved closer to the interface and θ_2 changes, the angle at which light hits the camera will change, so the distortion is clearly non-transversal. Correcting non-transversal distortion is extremely difficult, however, so the next question was whether or not one can model refraction as a typical transversal radial distortion.

3.1 Experimental Results of Modeling Refraction as Transversal Radial Distortion

A series of underwater images of calibration patterns were taken with the LUMIS2 and calibrated with the Bouguet calibration toolbox [3], and then the reprojection error was analyzed. The re-projection error was very low between .5 and 1 pixel error. However, reprojection error suffers from the problem of 'over-fitting' to the current depth of the scene. That is, if we changed the depth of the scene, then we may need an entirely different set of radial distortion parameters. I ran the same reprojection error test on five random but representative images of coral with 10-20 widely distributed points for each, and I received the same results. For this application's limited depth variation and small field of view (42 degrees in the horizontal), refraction is modeled extremely accurately with normal radial distortion.

4 Handling Fluorescence

The goal is to take the filtered blue light images and register them with the unfiltered white light images, however the color filters, difference in illumination and fluorescence complicate the task. In fact, I found that this challenge was too great (see Figure 1). Histogram matching and LoG filtering did not seem to improve results, so instead the problem was simplified by solving for the transformations between all the white light images and assuming it is the same for blue light images. So instead of finding the T(filtered blue -> unfiltered blue), I found T(filtered white -> unfiltered white) and assumed they are the same. To correct for jitter between sequences, a corrective distortion is calculated between the unfiltered blue light image and the unfiltered white image. This way, the task of compensating for different image filters is separated from the problem of different lighting, while removing the problem of fluorescence all together.

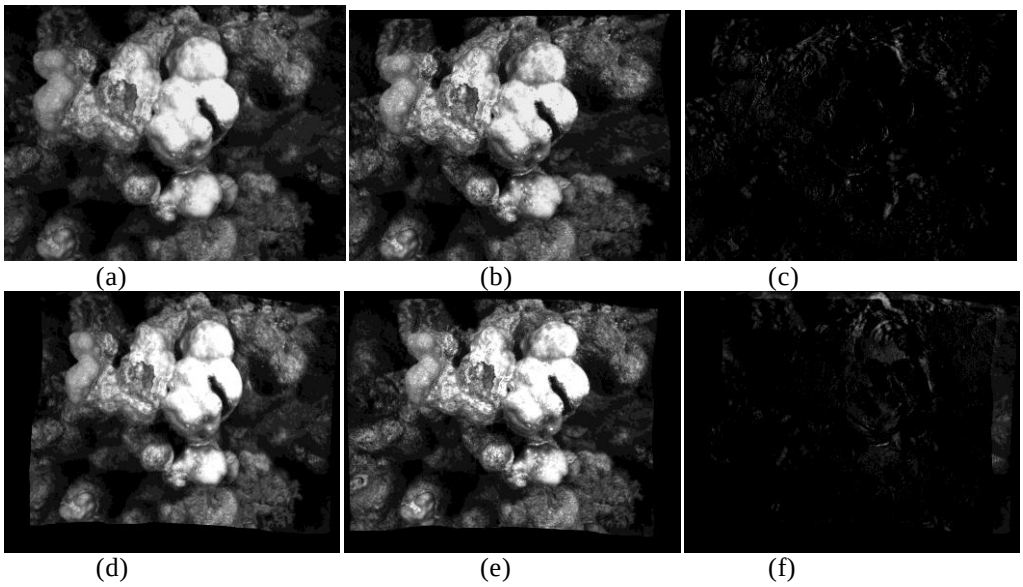


Figure 4: Result of thin spline: left image (a), right image (b), and absolute difference (c). Result of elastic: left image (d), right image (e), and absolute difference (f).

After experimenting with bandpass filtering and histogram matching, I found that histogram matching worked well with all the white light images (see figure 3) and continued using it successfully for registration.

5 Registration

Finding correspondences and solving for a transformation can be separated or rolled together in an iterative refinement algorithm. In my experience, algorithms that iterated both steps had much better matching results, so I'll discuss the two processes together under the banner of 'registration.'

5.1 Okutami-Kanade Algorithm

The first technique tried was a sparse method that uses the epipolar geometry of the four cameras to limit the search space for corresponding points. The algorithm builds upon basic stereo algorithms that take a point in one image and scan the other image finding a point that minimizes some function of 'correspondence-ness.' By using rectification, we can be sure that the corresponding points lie along the x axis. Okutomi and Kanade expanded on this idea by noting that $d = BF/z$ where B is the baseline between the images, F is the focal point, and z is the depth of the point [4]. Using this property, we can find correspondences between multiple images by minimizing the energy function of $I_1(x,y)$, $I_2(x+d_1,y)$, $I_3(x+d_2,y)$, etc, where every d_i is a function of the same depth z. In other words, we can simultaneously search all four images by searching for the best depth by using the calibrated baseline, B and focal point F. By combining all four views, I was able to get extremely accurate results. For a sparse collection of features found by a Harris edge detector, 90-100% match could be expected.

Despite the good results of the algorithm, performance crippled this algorithm. On a 1.8 GHz Intel Core 2Duo with 2 GB of RAM, points were collected at a rate of roughly 10 correspondences/second. This limited the number of correspondences to under a thousand, and for a scene with lots of depth like the coral images, I could not define a rich enough transform to map one image to another.

The methods for finding a sufficient transform used were thin plate spline and an

elastic transform.

5.1.1 Thin Plate Spline

Thin plate spline is a popular method for registering two scenes with a sparse number of correspondences [5]. Rather than defining a rigid transformation, thin-plate spline interpolates the interior points by using the depth of its neighbor correspondences. The effect is a transformation that models a flat surface to a surface with peaks and valleys defined by the sparse correspondences already found.

To test the efficacy of this method, I solved for the transform, mapped one image to the other using it, and then found the absolute difference of a region of interest of the two images in a range of 0-255. The results averaged over 4 random scenes were 7.3441 intensity change per pixel. The results can be seen in Figure 4. The code was provided by the toolbox bUnwarpJ [6].

5.1.2 Elastic transforms

Elastic deformation with spring meshes is another way of modeling the transformation from two images. The technique overlays an image with a virtual mesh of points and springs connected them. The sparse correspondences are then used to calculate the deformation of these springs that minimizes the error in the correspondences as well as the 'tension' in the springs. [7]

I used the same method for efficacy as before and received slightly worse results of 8.9535 intensity change per pixel. See Figure 4.

5.2 Optical Flow

Since sparse correspondence did not perform up to expectations, it was necessary to obtain dense correspondences. Of the many strategies for finding dense correspondence, I found optical flow to yield the best results [8].

Optical flow algorithms make two key assumptions about an image: brightness is consistent and there is very little change between two images. Using these two constraints, optical flow tries to minimize the following objective function:

$$E(u,v) = \sum (I_1(p+w) - I_2(p))^2$$

where p is an image point and w is a transformation composed of velocity vectors. The Lucas Kanade algorithm is a well known algorithm that solves for the velocity vector field in patches over the entire image. However, Lucas Kanade cannot find correspondences in blank regions with no discernable features (the aperture problem). To compensate for this problem, the objective function is expanded to:

$$E(u, v) = \sum ((|I_1(p+w) - I_2(p)|^2) + \alpha(|\nabla u|^2 + |\nabla v|^2))$$

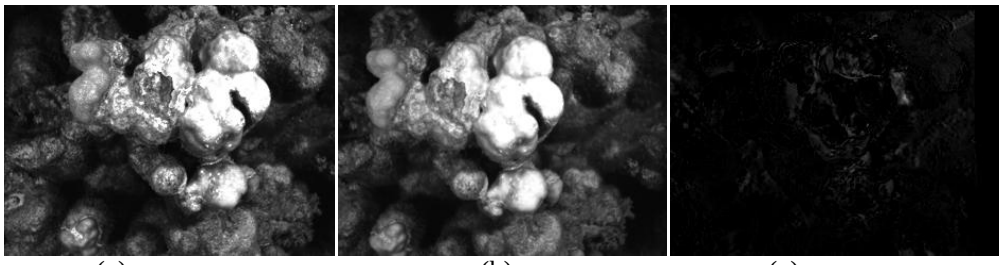
where u and v are velocity vectors, and ∇u and ∇v represent their gradients. So the $|\nabla u|^2 + |\nabla v|^2$ adds the additional constraint that there is some degree of smoothness in the vector field w. Finally α is a weight that shows the relative importance of the two constraints.

Because of the relatively short baselines of the coral images, and the relatively constant depth of the scene, optical flow lends itself well to this application. Although the distance isn't subpixel (as Lucas Kanade assumes), a pyramidal approach of 8 images resized by $.75^i$ captured the movement.

For this part of the project, I relied on Ce Liu's optical flow code [9] that provides an efficient implementation of the above algorithm and yields extremely accurate results of 4.2171 pixel differences per image on the same training set as before. See figure 6.

5.2.1 Performance

The code was modified slightly in order to allow the user to provide initial guesses of the velocity vector field, however, the algorithms performance only improved by about



(a) (b) (c)

Figure 5: Result of Lucas-Kanade: left image (a), right image (b), and absolute difference (c).

15%, even when the initial guess was the output of a previous run. The intuition for the cause is that most of the motion is captured in the 'higher' levels of the pyramid, which are very quick to compute. The majority of the computation is done at the lower levels of the image pyramid where the vector fields are refined. At these lower refinement stages, one cannot rely on estimates, so this computation is inescapable.

For the same reason as above, constraining Lucas Kanade to only create warps along epipolar lines will probably do very little to make the algorithm faster. At the low levels of the image pyramid where fine tuning takes place and most of the computation occurs, one usually cannot rely on extremely accurate epipolar lines (especially with the residual effects of refraction). Therefore, this avenue of optimization was not pursued.

In the end, the Lucas Kanade approach gives dense correspondences of the four scenes in about a minute on a 1.8 GHz Intel Core 2Duo with 2 GB of RAM. Of all the approaches tried, not only was this the most effective, but it was also the fastest.

6 Conclusion

The LUMIS2 presents unique challenges for calibration including refraction, differences in illumination, and scenes with large variations in depth. Additionally, registration needs to be as fast as possible to scale to large datasets. After further analysis, the issues of refraction and illumination could be prevented with relatively simple solutions like histogram equalization. However, a variety of registration techniques needed to be surveyed to find a dense enough map of correspondences. Images with large depth variation and a small baseline lend themselves extremely well for an optical flow algorithm. This approach has the added benefit of being relatively fast.

7 References

[1] Tali Treibitz, Yoav Y. Schechner, Clayton Kunz & Hanumant Singh, "Flat Refractive Geometry," oral in CVPR (2008)

[2] D. C. Brown, "Decentering distortion of lenses," *Photogrammetric Engineering 32* (1966): 444-462.

[3] J. B. Bouquet, "Camera Calibration Toolbox," http://www.vision.caltech.edu/bouguetj/calib_doc/. (accessed 5 Jun. 2011).

[4] M. Okutomi, T. Kanade, "A Multiple-Baseline Stereo," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 15, no. 4, pp. 353-363, Apr. 1993.

[5] A. Ardeshir Goshtasby, "2-D and 3-D Image Registration for Medical, Remote Sensing, and Industrial Applications," Wiley Press, 2005.

[6] I. Arganda-Carerras, "bUnwarpJ," http://biocomp.cnb.uam.es/~iarganda/bUnwarpJ/ (accessed 5

Jun. 2011).

[7] S. Saalfeld, "Elastic Alignment and Montage,"

http://pacific.mpi-cbg.de/wiki/index.php/Elastic_Alignment_andMontage. (accessed 5 Jun. 2011).

[8] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision." Proceedings of Imaging Understanding Workshop, pages 121-130, 1981.

[9] C. Liu. "Beyond Pixels: Exploring New Representations and Applications for Motion Analysis." Doctoral Thesis. Massachusetts Institute of Technology. May 2009.