Mid-Semester Project Report

Alex Watras

April 22, 2016

1 Intro

Laparascopic surgery is a method of minimally invasive surgical procedure where the entire surgery is performed through a small incision. To avoid making a larger incision, the surgeon must rely on a live camera feed for vision of the surgical area. Currently, the cameras used for laparascopic surgery have a single camera at the end of a probe. Due to the limited field of view (FOV) of the single camera, The surgeon can not keep the entire surgical area in frame at any given time.

To solve this problem, We will look at replacing the camera used for laparascopic surgery with a camera array. By replacing the single camera with an array, we can potentially greatly increase the FOV of our vision system, but we introduce new challenges and open up new possibilities for our problem.

We will assume that we have a 2x2 array of cameras attached to the surgical port that we are using to survey the surgical area. We have to now consider how we can best visualize our scene to help the surgeon.

For this project we have several goals.

1. Combine multiple video feeds from a 2x2 video camera array into a single mosaicked video feed.

2. Re-project images to a new viewpoint to allow surgeons to focus on specific portions of the surgical area.

3. Give surgeons depth perception by either real stereo vision, or approximated stereo vision using image morphing and view morphing techniques

2 Related Work

There are many existing methods for recreating a 3D scene from 2D images. Structured light methods such as the methods proposed by [5],[3], and the method used by the Microsoft kinect use a light source to project a pattern onto the scene. If we know the location of the light source relative to the camera, we can use basic triangulation techniques as outlined in [5] to determine a depth map of the scene.

Structure from motion algorithms such as are used in [8] do not need any external light source. Instead, they attempt to match points between multiple images and use those to estimate the relative position of the two cameras. Once those positions have been calculated, the same triangulation methods can be used to calculate the depth mapping of the scene.

On this project, we will attempt a method more similar to [10] where we can cut down on computation times by using a camera array with known relative pose to simplify the calculations necessary for 3D scene reconstruction.

3 Technical Solutions

The first challenge on this project is using an image mosaicking algorithm at each frame of the video to combine the 4 video feeds into a single feed. To achieve this, we can use a fairly standard Image mosaicking algorithm. First we need to find corresponding feature points in the two images. There are several existing algorithms and implementations of this. SIFT offers good robust feature detection. However, it is computationally very expensive. By using another algorithm such as SURF or ORB, we may be able to cut down computation times.

Once we have our corresponding feature points. We use a RANSAC method to compute the best homography for warping the images. Now that we are dealing with a 2x2 array of images, we have to decide how we want to warp images onto each other.

Finally, we apply our homography, and blend the resulting images together. We may choose to sequentially warp new images onto the existing mosaic. Alternatively, we could try to perform warping procedures in parallel to reduce computation time.

Once we have completed the image mosaicking step, we start looking at view morphing. We want to be able to control a virtual viewpoint for the mosaicked scene. There are several ways we may be able to do this. Initially we considered using a stereoscopic algorithm to generate a 3d depth map of the scene. However, this would only allow us to use scene points that we found in at least two cameras. Instead, we will use view morphing to try and morph the image between different possible image mosaics.

The concept of view morphing is that given a complete correspondence of pixels in two images, we can morph the image into an intermediary state by sliding each pixel along from the starting image to the result image. Using one of these methods, we can simulate any viewpoint in between our cameras locations while still utilizing all of the captured pixels. We may also be able to use a view morphing algorithm to simulate a stereoscopic 3D view by creating virtual viewpoints corresponding to each of the viewer's eyes.

Our final goal is to be able to do all of this in real time. So once we have all of these capabilities in place, we want to look into speeding up these algorithms so that they can be performed on live streaming video feeds. We will attempt to look at where computations can be sped up or skipped in order to facilitate view morphing on a live video stream.

4 Milestones Achieved

We have already created scripts to stitch together multiple images in both matlab and python. The scripts will stitch the images together and give a step by step run down of computation times, so that we can easily identify which steps in the process are too computationally expensive.

We have implemented a basic image reprojection model to attempt to synthesize a novel viewpoint for the scene we have captured. Unfortunately, one of the assumptions on image stitching algorithms is that all points lie approximately on a single plane. Because of this assumption, generating virtual 3D Views becomes impossible. However, this simplifies the process of generating novel viewpoints. In order to generate a novel viewpoint, we look at our assumption that all of the features lie in a single plane. The homography we use in the image stitching process creates a single plane in the 3D space that all of our image pixels can be considered to lie in.

5 Remaining Milestones

The next step is to apply view morphing techniques to the stitched images. The existing stitching algorithms we have will only morph the image into the viewpoints of the four real cameras, however, using view morphing techniques, we should be able to take these four mosaics and use them to create new view points from any virtual viewpoint between the camera locations.

Once we can create a virtual viewpoint, we can attempt to approximate human stereoscopic vision by creating two virtual viewpoints two simulate the viewpoint from each eye.

Finally, we will look at how we can speed up our algorithms in order to allow for using them on streaming video.

6 Resources

1. Chen, S. E., & Williams, L. (1993). View Interpolation for Image Synthesis. Acm Siggraph, 27, 279288. http://doi.org/10.1145/166117.166153

2. C.Zitnick, S.B.Kang, M.Uyttendaele, S.Winder, & R.Szeliski. (2004). High-Quality Video View Interpolation Using a Layered Representation. ACM Transactions on Graphics, 23(3), 600608

3. Gupta, M., & Nayar, S. K. (2012). Micro Phase Shifting Conventional Phase Shifting Projected images Micro Phase Shifting. IEEE Conference on Computer Vision and Pattern Recognition, 813820.

4. Huang, H., Kao, C., Lin, Y., & Hung, Y. (2000). Disparity-Based View Interpolation for Multiple Perspective Stereoscopic Displays. Virtual Reality, 3957.

5. Lanman, D., & Taubin, G. (2009). Build Your Own 3D Scanner : 3D Photography for Beginners. ACM SIGGRAPH 2009 Courses, 94. http://doi.org/10.1145/1665817.1665819

6. Nain, S., Science, I., Cheng, T., Taipei, I., National, E., & Technology, I. (1998). Disparity-Based View Morphing- Rendering A New Technique for Image-Based, 916. http://doi.org/10.1145/293701.293703

7. Seitz, S. M., & Dyer, C. R. (1995). Physically-valid view synthesis by image interpolation. Proceedings IEEE Workshop on Representation of Visual Scenes (In Conjunction with ICCV95), 1825. http://doi.org/10.1109/WVRS.1995.476848

8. Snavely, N., Seitz, S. M., & Szeliski, R. (2006). Photo tourism. ACM Transactions on Graphics, 25(3), 835. http://doi.org/10.1145/1141911.1141964

9. Szeliski, R. (2010). Computer Vision : Algorithms and Applications. Computer, 5, 832. http://doi.org/10.1007/978-1-84882-935-0

10. Yang, R., Welch, G., & Bishop, G. (2002). Real-time consensus-based scene reconstruction using commodity graphics hardware. Proceedings - Pacific Conference on Computer Graphics and Applications, 2002-Janua, 225234. http://doi.org/10.1109/PCCGA.2002.1167864