Rectified Surface Mosaics

Robert E. Carroll Steven M. Seitz
University of Washington

Abstract

We approach mosaicing as a camera tracking problem within a known parameterized surface. From a video of a camera moving within a surface, we compute a mosaic representing the texture of that surface, flattened onto a planar image. Our approach works by defining a warp between images as a function of surface geometry and camera pose. Globally optimizing this warp to maximize alignment across all frames determines the camera trajectory, and the corresponding flattened mosaic image. In contrast to previous mosaicing methods which assume planar or distant scenes, or controlled camera motion, our approach enables mosaicing in cases where the camera moves unpredictably through proximal surfaces, such as in medical endoscopy applications.

1. Introduction

Mosaics enable capturing the appearance of an entire scene in a single image. Many techniques have been proposed in the literature, based on a range of projection models including both perspective [1, 2] and multi-perspective varieties [3–5]. A common feature of almost all known projection models is that they introduce distortions, as certain scene characteristics (e.g., linearity, parallelism, length, angles, etc.) are not preserved in the mosaic. While some distortions may be tolerable or even desirable for casual visualization tasks, they can pose problems when precision or measurement is required.

One approach that avoids distortions is to define the projection based on a known reference plane in the scene. In this case, the images can be *rectified* to align with the physical coordinates on the reference plane, and stitched together. The result is a mosaic that removes the effects of perspective distortion and preserves lengths and angles for points on the reference plane [6].

In this paper, we generalize the approach of plane rectification to allow distortion-free rectified mosaics for any *developable surface* (e.g., boxes, cylinders, cones, generalized cylinders, etc.). More generally, our framework supports any kind of parametric surface, in which case the resulting mosaic image has the same parameterization as the surface itself. This generalization has a number of possible uses, but is particularly motivated by endoscopy applications, i.e., building mosaics of internal organs for medical diagnostics.

Our approach takes as input a video from a camera moving through a tube or other surface, and produces as output a planar image that represents the surface texture unwrapped onto a plane. The form of the surface must be known in advance, but the camera path may be unknown and unconstrained. Towards this end, we seek to align the images using a global objective function that is parameterized by surface geometry and camera pose in every image. We minimize this objective function using a global space-time version of Lucas-Kanade registration [7, 8], and composite strips of the aligned images into a mosaic. As a side effect, the approach solves for camera pose.

1.1. Related Work

Rousso *et al*. introduced "pipe projection" as a method that allows the mosaicing of video containing forward motion [9]. While this approach is very related to our goals, the "pipe" in their case is not meant to correspond to a physical surface in the scene, but is rather used as a manifold that allows radial optical flow to be transformed into parallel optical flow on the pipe's surface. Their approach is most effective when the scene is distant, or in the special case when the camera is moving along the center axis of a perfect cylinder. This pipe projection method is therefore not well suited for cameras moving freely through proximal, physical pipes or other surfaces, as in the case of endoscopy video, and produces significant distortions in these cases.

In computing camera pose through a global optimization, our approach bears relation to work in computer vision and photogrammetry on structure from motion (SfM) and bundle adjustment [10]. Indeed, SfM could in principle be used to compute the camera motion, as an alternative to approach proposed in this paper. However, SfM methods require tracked features as input, which is particularly problematic for endoscopy video of anatomical surfaces, as such video tends to have few distinct features and large parallax between frames. Instead, we adopt an alignment approach

that simultaneously solves for correspondence and camera pose in order to build the mosaic.

The problem of estimating camera pose in endoscopy imagery has been explored in the medical imaging community. Notably, several researchers [11–13] have demonstrated the ability to track a camera moving through the lung, computing six degree of freedom pose. This line of work builds on the use of *virtual endoscopy* to generate synthetic renderings of a CT model of the lung acquired off-line, that can be compared to the images captured online for the purpose of registration. This use of virtual images has the advantage of being able to correct for drift that would otherwise occur in frame-to-frame image registration. However, the accuracy of the alignment depends not just on the accuracy of the CT scan, but also the models for surface reflectance, shading, and lighting, and the ability to create renderings that approximate the true appearance of the scene. In contrast, our approach avoids the need to render a virtual model, and instead compensates for drift problems using a global multi-frame simultaneous alignment. A disadvantage of our approach is that the global optimization is not well-suited for real-time applications.

Vercauteren *et al*. [14] present an endoscopic mosaicing technique where frame-to-frame motion can be approximated by rigid transformations. This approach is well suited for situations where the camera directly faces the surface being imaged and the surface is relatively flat. Seshamani *et al*. [15] addressed the problem of mosaicing of cylindrical structures, but assume the camera motion is completely axial, allowing the warping and allignment procedures to be done independently. In contrast our approach is aimed at applications where the camera moves arbitarily through a more general, yet static surface.

Some authors have explored the related problems of unfolding 3D geometry [16, 17], computing triangle texture maps from images registered with CT scans [18], and unwrapping a single image using a circular parameterization [19].

2. Overview

The input to our algorithm is a sequence of perspective views from a camera moving within a known type of surface. From this sequence we wish to output a mosaic which represents the texture of the surface. To do this we estimate the 6 DOF camera pose for each frame. With known pose, we can do an inverse projection of each frame onto the surface. The 2D mosaic is formed by flattening the surface onto a plane, which can be done without creating any distortions if the surface is developable.

Pose estimation is done by defining a warping function between neighboring video frames. This warp is based on an inverse projection of one image onto the mosaicing surface followed by a projection onto another image plane. The warp is a function of the pose parameters of both images and the surface parameters. We use this warp to define an intensity minimization between frames, using the framework of Lucas-Kanade alignment [2,7,8]. We would like to compare each image to at least its two neighbors in the sequence, but this results in two (likely inconsistent) pose estimations for each frame. The series of duplicate pose estimations are not readily combined into a single camera path, so we instead generalize the registration algorithm into a global minimization across all frames.

To compute a consistent pose for each image, based on warps to multiple other images, we simultaneously solve for every pose. We use a generalization of Lukas-Kanade that handles multiple warps with interdependent parameters, which in our case are the camera pose parameters. Additionally, we show that the iterative update for the global optimization can be constructed from elements of the standard Lucas-Kanade method. Our global formulation is similar to the space-time tracking framework proposed by Agarwala *et al*. [20], who also used global optimization for tracking, but in the context of tracking 2D contours in a sequence of images.

Furthermore, we show that our framework can be extended to handle correspondences between input images and the projection surface. If the surface contains one or more known features that can be identified in the video, the pose estimations can be constrained to maintain the correspondence. This is useful in medical applications when the vanishing point of a cylindrical structure can be found or when known fiducials are present.

3. Surface Projection Warp

The image warp we wish to solve for is modeled as the combination of an inverse perspective projection from one camera location onto the mosaicing surface, followed by the projection to another camera location. An arbitrary surface $\mathbf{S}$ in 3-space can be parameterized by two variables, (s,t), and the image plane is parameterized by (u,v).

The mapping from surface coordinate to image coordinates and vice-versa are given by the following expressions:

$$\begin{bmatrix} u \\ v \end{bmatrix} = \mathbf{P}\left(\begin{bmatrix} s \\ t \end{bmatrix}, \mathbf{X}\right), \tag{1}$$

and

$$\begin{bmatrix} s \\ t \end{bmatrix} = \mathbf{P}^{-1}\left(\begin{bmatrix} u \\ v \end{bmatrix}, \mathbf{X}\right). \tag{2}$$

where $\mathbf{X} = (x, y, z, \alpha, \beta, \gamma)$ contains the six-degree-of-freedom camera pose.

The relationship between the 3D surface point $\mathbf{S}(s,t)$ and its projection onto the image plane at $\mathbf{u} = (u, v, f)$ is described by

$$\mathbf{S} = \mathbf{x} + \mathbf{R}\mathbf{u}c \tag{3}$$

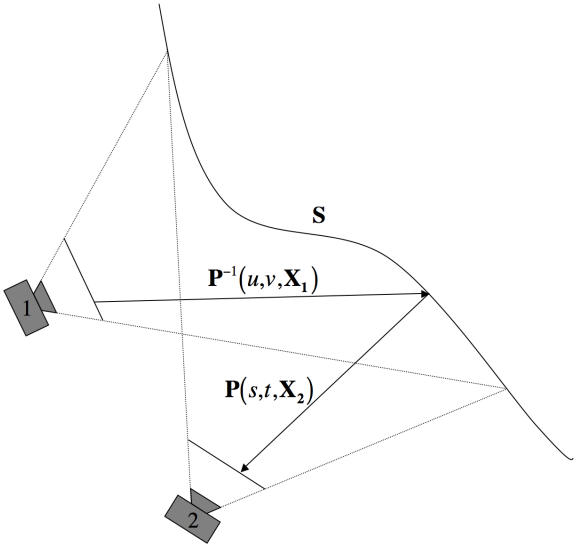


Figure 1. The surface projection warp is a combination of inverse projection onto the surface from camera 1, and projection back to the image plane of camera 2.

where $\mathbf{x} = (x, y, z)$ is the position of the camera, $\mathbf{R} = \mathbf{R}_x(\alpha)\mathbf{R}_y(\beta)\mathbf{R}_z(\gamma)$ is the rotation matrix representing the camera's orientation. The quantity $\mathbf{Ru}$ is the direction from the optical center to the 3D pixel location, adjusted to the coordinate system of the surface, and c is the scale factor representing the intersection of this surface along $\mathbf{Ru}$.

Solving Eq. (3) for $\mathbf{u}$ yields the following relation:

$$\mathbf{u} = \begin{bmatrix} u \\ v \\ f \end{bmatrix} = \mathbf{R}^{-1}(\mathbf{S} - \mathbf{x})/c \tag{4}$$

we can easily find c, since the focal length f is known, and thus we have the forward projection.

The inverse projection is defined by intersecting a ray from the camera with the surface. We define the projection such that the intersection with the smallest positive c is used if there are multiple ray-surface intersections. See the appendix for an example for a cylindrical surface.

Composing $\mathbf{P}^{-1}$ and $\mathbf{P}$ projects the first image through the surface into the coordinate system of the second image, giving the warp

$$\mathbf{W}(\mathbf{u}, \mathbf{X}_1, \mathbf{X}_2) = \mathbf{P}\left(\mathbf{P}^{-1}(\mathbf{u}, \mathbf{X}_1), \mathbf{X}_2\right). \tag{5}$$

4. Pairwise Pose Estimation

We wish to solve for a warp that minimizes the sum squared difference between two frames. That is, we wish to find $\mathbf{X}_1$ and $\mathbf{X}_2$ that minimizes the function

$$\sum_{\mathbf{u}} \left[I_1(\mathbf{u}) - I_2\left(\mathbf{W}(\mathbf{u}, \mathbf{X}_1, \mathbf{X}_2)\right)\right]^2. \tag{6}$$

We use a Lucas-Kanade style forwards additive[1] approach as described by Baker *et al*. [8]. Given a current estimate of $\mathbf{X}_1$ and $\mathbf{X}_2$, we wish to find iterative updates $\Delta\mathbf{X}_1$ and $\Delta\mathbf{X}_2$ that reduces the error function

$$\sum_{\mathbf{u}} \left[I_1(\mathbf{u}) - I_2\left(\mathbf{W}(\mathbf{u}, \mathbf{X}_1 + \Delta\mathbf{X}_1, \mathbf{X}_2 + \Delta\mathbf{X}_2)\right)\right]^2. \tag{7}$$

The closed form additive update for this equation is

$$\begin{bmatrix} \Delta\mathbf{X}_1 \\ \Delta\mathbf{X}_2 \end{bmatrix} = \mathbf{H}^{-1}\mathbf{b} \tag{8}$$

where $\mathbf{H}$ is the Hessian

$$\mathbf{H} = \sum_{\mathbf{u}} \left[\nabla I_2 \frac{\partial \mathbf{W}}{\partial(X_1, X_2)}\right]^T \left[\nabla I_2 \frac{\partial \mathbf{W}}{\partial(X_1, X_2)}\right], \tag{9}$$

and $\mathbf{b}$ is the residual [2]

$$\mathbf{b} = -\sum_{\mathbf{u}} \left[\nabla I_2 \frac{\partial \mathbf{W}}{\partial(X_1, X_2)}\right] \left[I_1(\mathbf{u}) - I_2(\mathbf{W}(\mathbf{u}, \mathbf{X}_1, \mathbf{X}_2))\right]. \tag{10}$$

Our warping function is the combination of two projections, so the Jacobian of the warp can be expressed in terms of the Jacobians of the projections

$$\frac{\partial \mathbf{W}}{\partial(X_1, X_2)} = \begin{bmatrix} \frac{\partial \mathbf{P}}{\partial \mathbf{s}} \frac{\partial \mathbf{P}^{-1}}{\partial \mathbf{X}_1} & \frac{\partial \mathbf{P}}{\partial \mathbf{X}_2} \end{bmatrix}. \tag{11}$$

In order to compute the Jacobian it is necessary that the surface be differentiable.

To improve convergence, we initialize the pose parameters based on the surface and the expected type of motion relative to that surface. For example, for a pipe we initialize the camera to be oriented axially, facing directly down the pipe. For a planar surface we initialize the camera to be facing the plane. All frames are given the same initial pose. For convenience, the world coordinates are chosen so that the initial pose is the zero vector. Depending on the surface there can be ambiguities in the warp, such as for a circular cylinder which is radially and axially symmetric. In these cases it may be desirable to partially constrain one of the frames, so as to prevent the minimization from searching among equivalent results. In the case of a cylinder, we constrain the first camera position to be at a particular depth and angle around the axis. The iterative update is run on a coarse to fine basis to improve convergence and performance.

[1] The forwards additive algorithm is computationally more expensive than the alternatives described in [8], but since the set of warps does not generally form a semi-group or group, the compositional algorithms are not applicable. Furthermore, the requirements for inverse additive approach are not satisfied.

5. Global Pose Estimation

The algorithm outlined in section 4 estimates two camera poses given two input images. To register a full video, we could apply the pairwise registration to each pair of consecutive images. However, this would result in *two* pose estimates at each frame, and they would likely be inconsistent. To obtain a single consistent pose at each frame, we reformulate the pairwise optimizations into one global optimization that minimizes the error between successive frames simultaneously.

The error function we wish to minimize is a sum of the pairwise errors,

$$\sum_{i=1}^{n-1} \sum_{\mathbf{u}} \left(I_i(\mathbf{u}) - I_{i+1} \left(\mathbf{W} \left(\mathbf{u}, \mathbf{X}_i, \mathbf{X}_{i+1} \right) \right) \right)^2 . \qquad (12)$$

As in section 4 the Jacobian for the warp between frame i and $i+1$ is given by

$$\mathbf{J}_i = \frac{\partial \mathbf{W}(\mathbf{u}, \mathbf{X}_i, \mathbf{X}_{i+1})}{\partial (\mathbf{X}_1, \mathbf{X}_2, ..., \mathbf{X}_n)}, \qquad (13)$$

the Hessian by

$$\mathbf{H} = \sum_{i=1}^{n-1} \sum_{\mathbf{u}} \left[\nabla I_{i+1} \mathbf{J}_i \right]^T \left[\nabla I_{i+1} \mathbf{J}_i \right], \qquad (14)$$

and the residual by

$$\mathbf{b} = - \sum_{i=1}^{n-1} \sum_{\mathbf{u}} \left[\nabla I_{i+1} \mathbf{J}_i \right] \left[I_i(\mathbf{u}) - I_{i+1}(\mathbf{W}(\mathbf{u}, \mathbf{X}_i, \mathbf{X}_{i+1})) \right]. \qquad (15)$$

The iterative update becomes

$$\begin{bmatrix} \Delta \mathbf{X}_1 \\ \vdots \\ \Delta \mathbf{X}_n \end{bmatrix} = \mathbf{H}^{-1} \mathbf{b}. \qquad (16)$$

Note that the Jacobian is sparse, as the parameters for each frame depend only on previous and next frame in the sequence. The Hessian for the global optimization is a $6n \times 6n$ square matrix. However, since only consecutive frames are compared the Hessian is sparse and banded, allowing Eq. (16) to be solved efficiently. The global Hessian and residual can be constructed from their pairwise counterparts, as is illustrated in figure 2.

We have derived a global optimization where each frame is compared to the immediately previous and next frames. However, this method is easily extended to compare each frame to any number of neighbors, at the cost of additional computational complexity. If we alter the global error to be

$$\sum_{i=1}^{n-1} \sum_{j=i+1}^{i+k} \sum_{\mathbf{u}} \left(I_i(\mathbf{u}) - I_j \left(\mathbf{W} \left(\mathbf{u}, \mathbf{X}_i, \mathbf{X}_j \right) \right) \right)^2 \qquad (17)$$

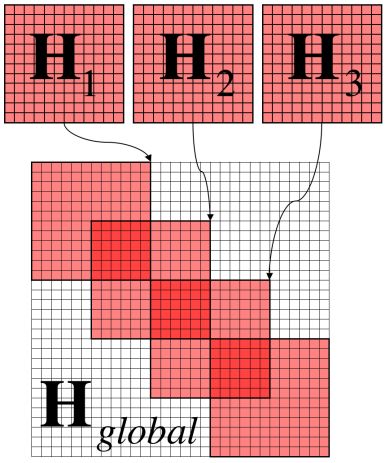


Figure 2. The banded Hessian matrix used in the global optimization is constructed from Hessians of the pairwise registration. The overlapping regions are summed.

the pose for each image will be determined by comparisons to k frames on each side. The value of k determines the width of the band in the global Hessian. The advantage of larger values of k is more reliable pose estimation that uses large and small motions for better local and global alignment. In the limit we compare each image to every other image and the objective function becomes

$$\sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \sum_{\mathbf{u}} \left(I_i(\mathbf{u}) - I_j \left(\mathbf{W} \left(\mathbf{u}, \mathbf{X}_i, \mathbf{X}_j \right) \right) \right)^2 , \qquad (18)$$

in which case the Hessian is completely filled in.

6. Surface Feature Constraints

It may be possible to detect certain features in the input images that are known to correspond to a particular location on the surface. For example, for the case of endoscopy inside of cylindrical surfaces, it is straightforward to detect the cylinders vanishing point as it appears as a dark spot in the video. We wish to constrain our optimization so that the location of the detected feature in each image coincides with the projection of its known surface location into that image. We can include these constraints into our optimization by adding the weighted squared distance of the two into the error function. The new global error is

$$\sum_{i=1}^{n-1} \sum_{\mathbf{u}} \left(I_i(\mathbf{u}) - I_{i+1} \left(\mathbf{W} \left(\mathbf{u}, \mathbf{X}_i, \mathbf{X}_{i+1} \right) \right) \right)^2 \qquad (19)$$

$$+ \sum_{i=1}^{n} \sum_{f} w \left| \mathbf{P}(\mathbf{s}_f, \mathbf{X}_i) - \mathbf{u}_f \right|^2$$

where $\mathbf{s}_f$ is the known location of a feature on the surface, $\mathbf{u}_f$ is the location of the feature in image i, and w is a

weighting term. Since the constraints are independent of other images, we can treat them individually. The constraint for features in an image is given by

$$\sum_f |\mathbf{P}(\mathbf{s}_f, \mathbf{X}) - \mathbf{u}_f|^2 \tag{20}$$

We can proceed as in section 4 to get the Hessian for the surface feature constraints

$$\mathbf{H} = \sum_f \frac{\partial \mathbf{P}}{\partial \mathbf{X}}^T \frac{\partial \mathbf{P}}{\partial \mathbf{X}}, \tag{21}$$

and the residual

$$\mathbf{b} = -\sum_f \frac{\partial \mathbf{P}}{\partial \mathbf{X}}^T \left[\mathbf{P}(\mathbf{s}_f, \mathbf{X}) - \mathbf{u}_f\right]. \tag{22}$$

The weighted Hessian and residual for each feature constraint can be added to their global counterparts in the same way as the pairwise pose estimation.

7. Mosaic Construction

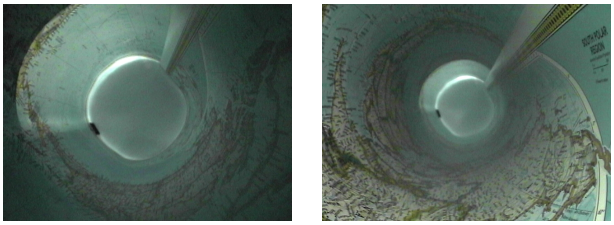
Once the camera pose is know for each image, the mosaic is constructed by extracting strips of the images, projecting them into the mosaic, and blending them together. Our approach is to traverse the video sequence, projecting onto new regions of the surface as they come into view. The size and shape of the strips are defined implicitly by yet unseen regions in the mosaic, which have come into view in the new frame. More generally we define a boundary in the image that determines potential strips, since the image edges tend to have more distortion and misredgistration. Finally, to remove any remaining seams we use gradient domain blending [21–23].

8. Experimental Results

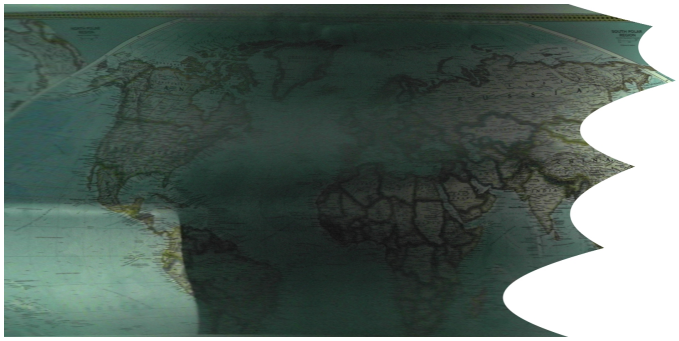
In this section we will demonstrate our algorithm on various datasets to produce scene visualizations that could not have been captured with a single physical camera. Runtime for each of these experiments was 20-30 minutes on a 3.2GHz Pentium IV.

8.1. World Map

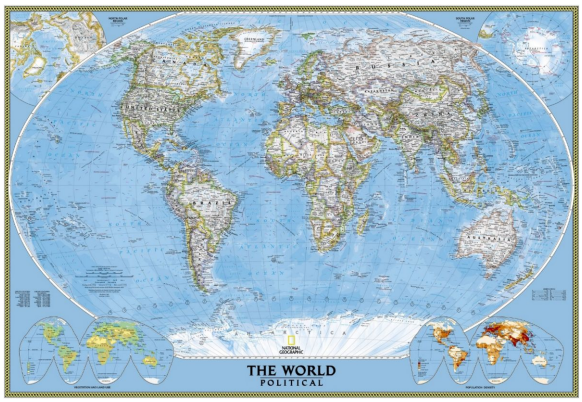
This experiment involved moving a camera down a rigid 10 inch diameter, 5 ft. long tube lined with a rolled up world map. The camera was inserted into the tube on a plastic tray. The video was taken with a consumer camcorder and the scene was unevenly lit, as can be seen in the input images. Along with a limited depth of field, these issues make the registration challenging. The resulting mosaic (Fig. 3) is constructed from strips taken from 400 images. The bottom of the map is cut off since it is not visible when rolled up. The mixed exposure in the mosaic is caused by uneven lighting conditions. Despite the low quality input video, the recovered mosaic closely matches the reference image, demonstrating the algorithm's capability for metric accuracy.



Input Images



Rectified Surface Mosaic

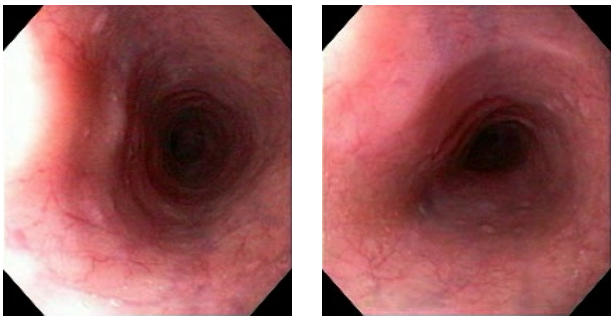


Reference

Figure 3. Video of a rolled up world map, mosaiced onto a cylindrical surface.

8.2. Esophageal Endoscopy

This example (Fig. 4) was made from 220 frames of an esophageal endoscopy procedure on a sedated patient. This is one of the motivations for our project, to create a diagnostic tool for precancer screenings. For this sequence a dark spot corresponding to cylinder's axial vanishing point was detected, and this constraint was used in the registration.



Input Images

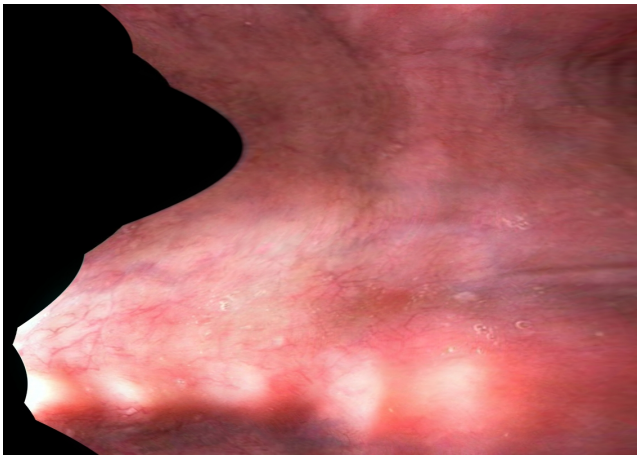


Figure 4. Mosaic from esophageal endoscopy video using a cylindrical surface. The dark spot is constrained to correspond to the cylinder's vanishing point.

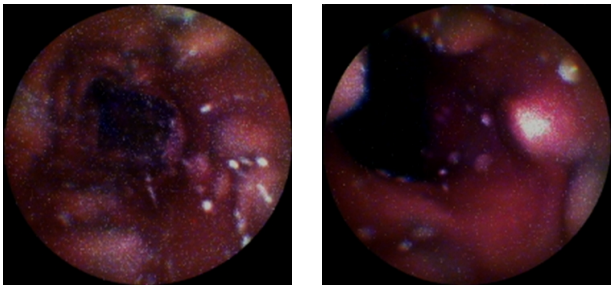
8.3. Model Esophagus

This mosaic (Fig. 5) was created as a proof of concept for the algorithm's use with an ultrathin Scanning Fiber Endoscope being developed at the University of Washington Human Photonics Lab [24]. The camera is unique in that it captures pixels sequentially from a spiraling optical fiber. The fiber scope was inserted into a model esophagus to capture a 250 frame sequence. Given the low quality, grainy input video our registration algorithm produces an impressive result.

9. Conclusion

This paper presented a technique for creating rectified image mosaics that avoids distortions and facilitates accurate measurement. The method employs a Lucas-Kanade style optimization to estimate camera pose in every frame, but applied globally over the sequence to address problems of drift and tracking error. Our work is motivated by applications in endoscopy and we show panoramic mosaics from esophagus videos. We also include a test on a cylindrical object that allows evaluation with respect to ground truth. The approach is shown to produce good results even with low quality input video.

Interesting directions for future work include developing faster optimization methods, and extending the approach to work in the presence of non-rigid motion. Our formulation can also be extended to include shape parameters, enabling mosaicing of unknown shapes.



Input Images

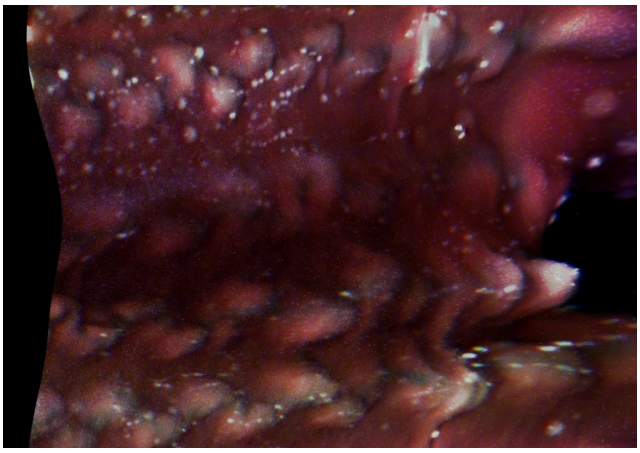


Figure 5. Mosaic of a model esophagus imaged using a Scanning Fiber Endoscope [24].

Acknowledgements

The support of Pentax and NSF grant IIS-0413198 are gratefully acknowledged. We wish to thank Eric Seibel for suggesting this problem and for helping to guide our research.

References

[1] S. E. Chen. Quicktime vr: an image-based approach to virtual environment navigation. In *SIGGRAPH*, pages 29–38, 1995.

[2] R. Szeliski and H.-Y. Shum. Creating full view panoramic image mosaics and environment maps. In *SIGGRAPH '97: Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, pages 251–258, 1997.

[3] D. N. Wood, A. Finkelstein, J. F. Hughes, C. E. Thayer, and D. H. Salesin. Multiperspective panoramas for cel animation. In *SIGGRAPH '97: Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, pages 243–250, 1997.

[4] A. Agarwala, M. Agrawala, M. Cohen, D. Salesin, and R. Szeliski. Photographing long scenes with multi-viewpoint panoramas. *ACM Trans. Graph.*, 25(3):853–861, 2006.

[5] S. M. Seitz and J. Kim. The space of all stereo images. *Int. J. Comput. Vision*, 48(1):21–38, 2002.

[6] R. Szeliski. Video mosaics for virtual environments. *IEEE Computer Graphics and Applications*, 16(2):22–30, 1996.

[7] B. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *Proceedings of the 7th International Joint Conference on Artificial Intelligence*, pages 674–679, 1981.

[8] S. Baker and I. Matthews. Lucas-kanade 20 years on: A unifying framework: Part1. Technical Report CMU-RI-TR-02-16, Robotics Institute, Carnegie Mellon University, Pittsburg, PA, July 2002.

[9] B. Rousso, S. Peleg, I. Finci, and A. Rav-Acha. Universal mosaicing using pipe projection. In *Proc. ICCV*, pages 945–952, 1998.

[10] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition, 2004.

[11] I. Bricault, G. Ferretti, and P. Cinquin. Registration of real and ct-derived virtual bronchoscopic images to assist transbronchial biopsy. *IEEE Trans. Med. Imaging.*, 17(5):703–14, Oct 1998.

[12] K. Mori, Y. Suenaga, J. Toriwaki, J. Hasegawa, K. Kataa, H. Takabatake, and H. Natori. A method for tracking camera motion of real endoscope by using virtual endoscopy system. volume 3978, pages 122–133. SPIE, 2000.

[13] J. P. Helferty, E. A. Hoffman, G. McLennan, and W. E. Higgins. Ct-video registration accuracy for virtual guidance of bronchoscopy. volume 5369, pages 150–164. SPIE, 2004.

[14] T. Vercauteren, A. Perchant, G. Malandain, X. Pennec, and N. Ayache. Robust mosaicing with correction of motion distortions and tissue deformations for in vivo fibered microscopy. *Medical Image Analysis*, 10(5):673–692, Oct 2006.

[15] Sharmishtaa Seshamani, William Lau, Gregory D. Hager. Real-time Endoscopic Mosaicking. In *MICCAI*, pages 355–363, 2006.

[16] P. Carrascosa, C. Capunay, E. M. Lopez, M. Ulla, R. Castiglioni, and J. Carrascosa. Multidetector ct colonoscopy: evaluation of the perspective-filet view virtual colon dissection technique for the detection of elevated lesions. *Abdominal Imaging*, Dec 2006.

[17] T. Truong, T. Kitasaka, K. Mori, and Y. Suenaga. Fast and accurate tract unfolding based on stable volumetric image deformation. volume 6143, pages 412–423. SPIE, 2006.

[18] L. Rai and W. E. Higgins. Image-based rendering method for mapping endoscopic video onto ct-based endoluminal views. volume 6141, pages 1–12. SPIE, 2006.

[19] J. R. Warmath, Z. Cao, P. Bao, A. J. Herline, and R. L. G. Jr. Semi-automatic staging system for rectal cancer using spatially oriented unwrapped endorectal ultrasound. volume 5744, pages 425–434. SPIE, 2005.

[20] A. Agarwala, M. Dontcheva, M. Agrawala, S. Drucker, A. Colburn, B. Curless, D. Salesin, and M. Cohen. Interactive digital photomontage. In *Proc. SIGGRAPH*, pages 294–302, 2004.

[21] A. Agarwala, A. Hertzmann, D. H. Salesin, and S. M. Seitz. Keyframe-based tracking for rotoscoping and animation. *ACM Trans. Graph.*, 23(3):584–591, 2004.

[22] P. Pérez, M. Gangnet, and A. Blake. Poisson image editing. *ACM Trans. on Graphics*, 22(3):313–318, 2003.

[23] R. Fattal, D. Lischinski, and M. Werman. Gradient domain high dynamic range compression. *ACM Trans. on Graphics*, 21(3):249256, 2002.

[24] E. J. Seibel, R. S. Johnston, and C. D. Melville. A full-color scanning fiber endoscope. In I. Gannot, editor, *Optical Fibers and Sensors for Medical Diagnostics and Treatment Applications VI. Edited by Gannot, Israel. Proceedings of the SPIE, Volume 6083, pp. 9-16 (2006).*, volume 6083 of *Presented at the Society of Photo-Optical Instrumentation Engineers (SPIE) Conference*, pages 9–16, Mar. 2006.

A. Cylindrical Pipe

The case of a cylindrical surface is derived by replacing the surface parameters (s, t) from section 3 with the cylinder's parameters (k, θ), where k is depth along the pipe's axis and θ is the angle around the axis. Then

$$\mathbf{S}(k, \theta) = \begin{bmatrix} r\cos(\theta) \\ r\sin(\theta) \\ k \end{bmatrix}$$

and relation between surface coordinate and image coordinate becomes

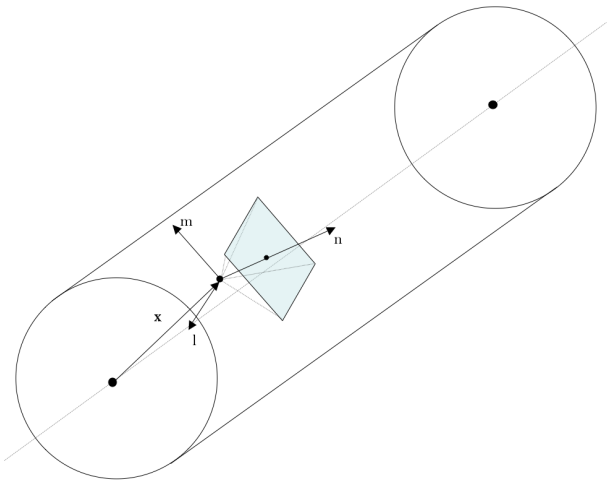


Figure 6. The pipe projection: $\mathbf{l}$, $\mathbf{m}$, and $\mathbf{n}$ represent the image's basis vectors in scene coordinates. $[\mathbf{l}\ \ \mathbf{m}\ \ \mathbf{n}] = \mathbf{R}$ represents the camera's orientation, and $\mathbf{x}$ is its position.

$$\begin{bmatrix} rcos(\theta) \\ rsin(\theta) \\ k \end{bmatrix} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \mathbf{R} \begin{bmatrix} u \\ v \\ f \end{bmatrix} c.$$

From this equation the mapping from (k, θ) to (u, v) is straightforward, but going from (u, v) to (k, θ) is a bit more tricky. If u and v are known we can solve for c by noting

$$r^2 = (r\cos(\alpha))^2 + (r\sin(\alpha))^2 = (c(\mathbf{Ru})_x + x)^2 + (c(\mathbf{Ru})_y + y)^2.$$

This gives us a quadratic in c. With known c it follows that $\alpha = \arctan((y + P_y c)/(x + P_x))$ and $k = z + (\mathbf{Ru})_z c$. Here it is assumed r is known, but this can be chosen arbitrarily as it is an ambiguity in the scene. Changing r will result in a stretching of the mosaic.