Parallax-Free Registration of Aerial Video *

Daniel Crispell Joseph Mundy Gabriel Taubin
Brown University, Providence, RI, USA
daniel_crispell@brown.edu

Abstract

Aerial video registration is traditionally performed using 2-d transforms in the image space. For scenes with large 3-d relief, this approach causes parallax motions which may be detrimental to image processing and vision algorithms further down the pipeline. A novel, automatic, and online video registration system is proposed which renders the scene from a fixed viewpoint, eliminating motion parallax from the registered video. The 3-d scene is represented with a probabilistic voxel model, and camera pose at each frame is estimated using an Extended Kalman Filter and a refinement procedure based on a popular visual servoing technique.

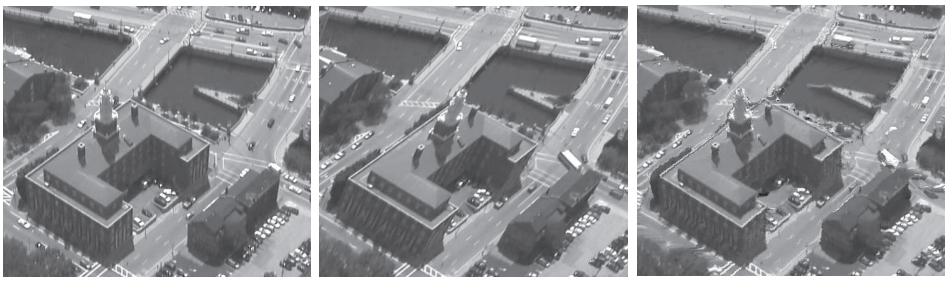
1 Introduction

Video registration is an important problem in aerial surveillance applications. When imaging scenes that can be approximated as planar, 2-d image transformations generally suffice for this purpose. When imaging a scene with significant 3-d structure, however, 2-d registration techniques lead to errors caused by motion parallax. Many imaging systems [3] require precise video registration in order for higher level image processing such as foreground detection and tracking to be accurately performed. In order to operate correctly in highly non-planar environments such as urban or mountainous landscapes, scene geometry must be accounted for in some way. A novel, fully automatic 3-d registration system is proposed based on a probabilistic voxel model of the scene's geometry and appearance, with camera pose recovery formulated as a Kalman Filtering problem. The system operates online, meaning each image is registered as soon as it is available, with no knowledge of future data. A unique pose estimate refinement step using visual servoing techniques in conjunction with imagery generated using the probabilistic voxel-based scene model is also presented.

2 Prior Work

Image registration is a fundamental problem in many applications such as surveillance, geographic information systems (GIS), medical imaging, and mosaic creation. Because the registration system computes and utilizes information about the underlying 3-d scene and camera pose, it is also related to work in the fields of 3-d modeling and automatic camera calibration.

*This material is based on work supported by DARPA grant no. NBCH1060013.



(a) (b) (c)

Figure 1: (a) A region of the first frame of the "Steeple St." sequence. (b) The 70th frame, registered using a 2-d ground-plane registration. (c) The 70th frame, registered using the proposed system.

2.1 2-d Image and Video Registration Techniques

A comprehensive survey of image registration techniques was presented by Zitovia in 2003 [19]. Many traditional methods of image registration assume that the scene is approximately planar and/or the scene is being viewed from a single viewpoint. Under these assumptions, a 2-d image transformation (a homography for projective cameras) can be used to map pixels from one image to another. Transformations can be computed based on matching feature points [16], or via direct comparisons of pixel values in the two images.

When the planarity assumption is violated and the viewpoint is not fixed, parallax motions are induced. While most 2-d registration algorithms simply treat the motions as outliers, Rav-Acha et al. [13] showed that small parallax motions could be predicted based on previous frames, and ignored in the registration process. Another method for dealing with parallax motions without 3-d information is to relax the assumption of a global image transformation and define a locally-varying map. Caner et al. [1] estimate the parameters of a spatially varying filter which maps pixels in one image to a base image.

2.2 Utilizing 3-d information

If information about the 3-d scene is available, it can be used to produce more robust registration results. Two relevant application domains are orthorectification and model-based rendering.

In the GIS field, digital elevation models (DEM's) are routinely used in order to produce imagery orthographically rendered from above. Zhou et al. [18] provided a study of orthorectification methods for urban terrain, and Zhou and Chen [17] presented methods for forested areas. Satellite imagery can also be used to refine existing DEM's using stereo methods, increasing the accuracy of the orthorectification [9, 7].

General polygonal meshes can also be used to render scenes from novel viewpoints through texture mapping. In order to implement such a system, both the 3-d polygonal mesh and projection of the mesh into the input images must be known. Sawhney et al. [14] align edge features in the image to projected edges of a (fixed) 3-d model in order to optimize camera pose and render the scene from a viewpoint chosen interactively.

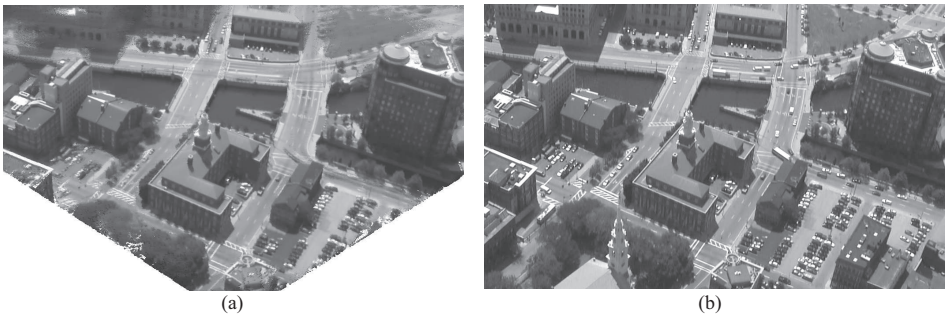
2.3 Camera Calibration

It is assumed that the internal parameters of the camera are known, but its pose is not. The full calibration is needed in order to relate images of the scene to the 3-d model and must be computed automatically. There have been many publications on the topic of automatic calibration since Maybank and Faugeras presented their work [10] in 1992. Many algorithms, including noncausal structure from motion (SFM) techniques [6, 12], perform well but are not suitable for online systems because they optimize parameters for all images in a sequence simultaneously using bundle adjustment. Simultaneous Localization and Mapping (SLAM) systems such as those presented by Davidson [4] and Chiuso et al. [2] require real-time estimation of camera position and 3-d points. Typically, both SFM and SLAM algorithms rely on feature detection and matching/tracking to relate images to one another. The proposed system does not compute any feature points, but rather uses all information available in the image to optimize the camera pose and 3-d model.

3 The Voxel Model

The probabilistic voxel model proposed by Pollard and Mundy [11] for use in 3-d change detection is used to accumulate information about the scene. Each voxel X is associated with both an appearance model and an occupancy probability $P(X \in S)$, which stores the probability that a world surface lies within X. It is assumed that for each pixel (i, j) in an image of the scene, the intensity $I_{i,j}$ is produced by an unoccluded voxel $V_{i,j} \in S$. The probability of a voxel X producing an intensity I in an image pixel given that $X = V$ is represented by a mixture of Gaussian density distribution.

Given a new image of the scene, the occupancy probability and appearance model parameters of the voxel X are updated according to the equations given by Pollard and Mundy [11]. Intuitively, for each pixel in the new image, a ray is cast into the scene which intersects some set of voxels. The voxels whose appearance models indicate that they are likely to have produced the intensity at the pixel have their occupancy probability increased accordingly, and vise-versa. Each voxel along the ray then has its appearance model updated using the pixel's intensity, weighted by the likelihood of the voxel being visible to the camera.



(a) (b)

Figure 2: (a) An expected image generated from the point of view of camera 70 of the "Steeple St." sequence. Note that moving vehicles on the streets do not appear in the expected image because of their low probability to exist at any given location. (b) The original image.

3.1 Expected Image Generation

Given a voxel model and a camera viewing the scene, the expected value of the produced image can be determined. Each camera ray passes through a set of voxels, R. The expected value of the intensity I_R associated with R can be calculated as a weighted average of the expected intensities $E[I|V=X]$ at each voxel $X \in R$:

$$E[I] = \sum_{X \in R} E[I|V=X] \frac{P(X \in S)\,\mathrm{vis}(X)}{W}, \quad W = \sum_{X \in R} w_X \quad . \tag{1}$$

The term $\mathrm{vis}(X)$ represents the probability that voxel X is visible from the camera's viewpoint, and is calculated as:

$$\mathrm{vis}(X) = 1 - \prod_{X' < X} (1 - P(X')) \quad , \tag{2}$$

with $X' < X$ denoting voxels in the set R occurring before X, i.e. closer to the camera center. The expected value $E[I|V=X]$ is the expected value of the mixture of Gaussians distribution.

4 Camera Optimization

The change detection algorithm presented by Pollard and Mundy assumed fully calibrated cameras, an impractical assumption for an online video registration system. It is assumed, however, that the internal parameters of the camera have been calibrated and that an estimate of the ground plane relative to the camera is known (e.g. from onboard altitude and attitude sensors). This estimate is needed only for the first frame of the video to initialize the voxel model. In order to reliably estimate relative camera pose for each frame, an Extended Kalman Filter (EKF) is implemented, with the state x at time step k representing the camera motion relative to time step $k-1$. A novel extension of a popular visual servoing technique is used to refine the Kalman filter state estimate at each time step.

4.1 Representation of 2-d and 3-d Transformations

The 2-d general affine matrix group $GA(2)$ is used to represent image homographies, and the special Euclidian matrix group $SE(3)$ to represent camera motions. Drummond and Cipolla [5] showed that by using Lie Algebra representations, 3-d information about the world is implicitly embedded into the 2-d image transformations. Although the goal is to accurately handle non-planar scenes, an assumption is made that the camera motion between two successive frames is sufficiently small to be approximated by a 2-d homography induced by a dominant world plane $\Pi = \begin{bmatrix} \hat{n}_x & \hat{n}_y & \hat{n}_z & d \end{bmatrix}^T, \|\hat{n}\| = 1$. Disregarding degenerate cases, Π can be represented using three parameters:

$$\theta = \tan^{-1} \frac{\hat{n}_x}{-\hat{n}_z}, \quad \phi = \tan^{-1} \frac{\hat{n}_y}{-\hat{n}_z}, \quad d_z = \frac{-d}{\hat{n}_z} \tag{3}$$

(See Figure 3). The Lie group $SE(3)$ has an associated Lie algebra $se(3)$, which is spanned by the so-called $SE(3)$ *generator matrices* E_i, $i \in \{1,2\ldots6\}$. The six $se(3)$ bases correspond to translation in x, y, and z, and rotation about the x,y, and z axes, respectively.

Likewise, the Lie group $GA(2)$ has an associated Lie algebra $ga(3)$ which is spanned by the $GA(2)$ generator matrices G_i, $i \in \{1,2\ldots6\}$. The six $ga(2)$ bases correspond to shift in x, shift in y, rotation, scaling, shear at 90°, and shear at 45°, respectively. Using these bases, the vectors $\vec{x} \in \Re^6$ and $\vec{z} \in \Re^6$ are defined, representing infinitesimal 3-d Euclidean and 2-d affine transformations, respectively. Using the dominant world plane parameterized by θ, ϕ, and d_z, The Jacobian matrix which maps infinitesimal changes in the camera pose to changes in the induced homography can then be defined, i.e. $J_{i,j} = \frac{\delta \vec{z}_i}{\delta \vec{x}_j}$.

$$J = \begin{bmatrix} 1/d_z & 0 & 0 & 0 & 1 & 0 \\ 0 & 1/d_z & 0 & -1 & 0 & 0 \\ \frac{\tan(\phi)}{2d_z} & \frac{-\tan(\theta)}{2d_z} & 0 & 0 & 0 & 1 \\ \frac{-\tan(\theta)}{2d_z} & \frac{-\tan(\phi)}{2d_z} & -1/d_z & 0 & 0 & 0 \\ \frac{-\tan(\theta)}{2d_z} & \frac{\tan(\phi)}{2d_z} & 0 & 0 & 0 & 0 \\ \frac{-\tan(\phi)}{2d_z} & \frac{-\tan(\theta)}{2d_z} & 0 & 0 & 0 & 0 \end{bmatrix} \tag{4}$$

The derivation of this matrix is not presented here but is very similar to one presented by Drummond and Cipolla [5], with the main difference being they assume that the world plane normal lies in the YZ plane, and thus use two plane parameters only. Note that columns 3 through 5 are approximate only, because in general a full projective image transformation is needed to model changes caused by translation along the camera axis and rotation around an axis other than the camera's principal axis.

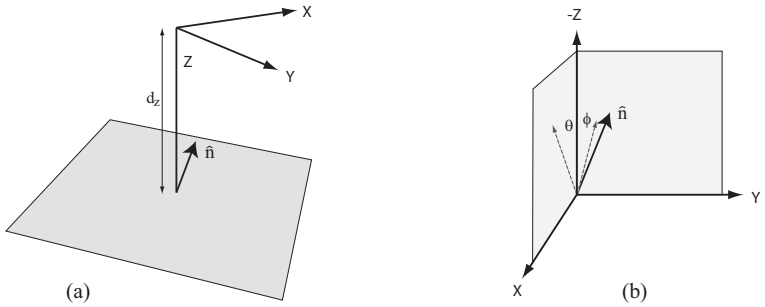


Figure 3: (a) The dominant world plane is shown in the camera coordinate system. (b) The plane normal $\hat{n}$ is projected onto the X-Z and Y-Z planes, giving the plane parameters θ and ϕ.

4.2 Kalman Filter Formulation

The Extended Kalman Filter (EKF) is an extension of the Kalman Filter that allows the filter to be applied to non-linear processes, and processes with non-linear measurement functions. Unlike the standard Kalman Filter, the EKF does not give provably optimal results due to the fact that the random variables are no longer normal after undergoing non-linear transformations. Despite this fact, it is widely used for a variety of applications and performs well for processes that are close to linear on the scale of the time increments [15]. The filter assumes that the system state, x_k, is a function f of the previous state x_{k-1} and an input u_k, plus a zero mean random variable w_k. The filter estimates a

state $\hat{x}_k$ and error covariance P_k of the estimate at time step k using two steps. The first step predicts the current state and covariance based on the previous state estimate $\hat{x}_{k-1}, P_{k-1}$ and the control input u_k.

$$\hat{x}_k^- = f(\hat{x}_{k-1}, u_k) + w_k, \qquad P_k^- = A_k P_{k-1} A_k^T + Q_{k-1} \tag{5}$$

A_k is the Jacobian matrix $\frac{\delta f}{\delta x}$, and Q_{k-1} is the covariance matrix of w. The prediction is then updated based on a measurement vector z_k. It is assumed that z_k is a function h of x_k plus a normal, zero mean random variable v_k with covariance R_k.

$$\hat{x}_k = \hat{x}_k^- + K_k(z_k - h(\hat{x}_k^-)), \qquad P_k = (I - K_k H_k) P_k^- \tag{6}$$

H is the Jacobian matrix $\frac{\delta z}{\delta x}$, and K is the Kalman gain, calculated as

$$K_k = P_k^- H_k^T (H_k P_k^- H_k^T + R_k)^{-1} \quad . \tag{7}$$

A concise derivation of the update and gain functions can be found in the report by Welch and Bishop [15].

An EKF is used to estimate the change in camera pose from one frame to the next. The state vector $x \in \Re^6$ contains the coefficients associated with the six $se(3)$ basis matrices. A linear (in the space of se(3)) motion model is used, so that the function f used in Equation 5 is defined simply as $x_k = x_{k-1}$, and A is the 6×6 identity matrix, I_6. The measurement vector $z_k \in \Re^6$ contains the coefficients associated with the six $ga(2)$ basis matrices which map frame $k-1$ to the current frame k. In order to estimate z_k, a multi-grid Levenberg-Marquardt minimization of the sum of squared differences between frame $k-1$ and frame k warped by z_k is used. The Jacobian matrix H is the Euclidian to affine Jacobian J defined in Equation 4. The state and measurement covariance matrices are assumed constant for all k, and the errors associated with the individual state and measurement elements are considered independent:

$$Q_k = \begin{bmatrix} \sigma_t^2 I_3 & \\ & \sigma_r^2 I_3 \end{bmatrix} \quad , \quad R_k = \sigma_h^2 I_6 \tag{8}$$

The standard deviations of the errors in the state translation, state rotation, and homography measurement coefficients are represented as σ_t, σ_r, and σ_h, respectively.

4.3 Refinement using Visual Servoing and Expected Images

The *a posteriori* estimate $\hat{x}_k$ may contain errors due to the non-planarity of the scene, perspective components of the homography ignored by the affine model, linearization of the measurement function h via the Jacobian matrix H, and noise. Data from previous images accumulated in the voxel model is used to produce a refined estimate, $\hat{x}_k^+$. An expected image is rendered from the viewpoint defined by $\hat{x}_k$ using Equation 1. If $\hat{x}_k$ matches the true state x_k, the homography bringing the expected image and frame k into alignment should be the identity ($z_k^+ = 0$). If it is not, the inverse Jacobian J^{-1} is used to move the estimate towards the correct state, and a new expected image is generated using the adjusted state $\hat{x}_k^+$. The processes is repeated until the adjustments to $\hat{x}_k^+$ fall below a fixed threshold, or a maximum number of iterations is reached. Once the estimate converges, the voxel model is updated with information from the current image and the

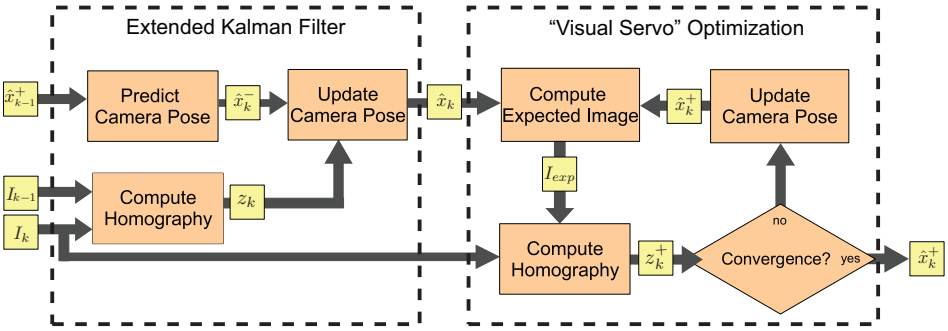


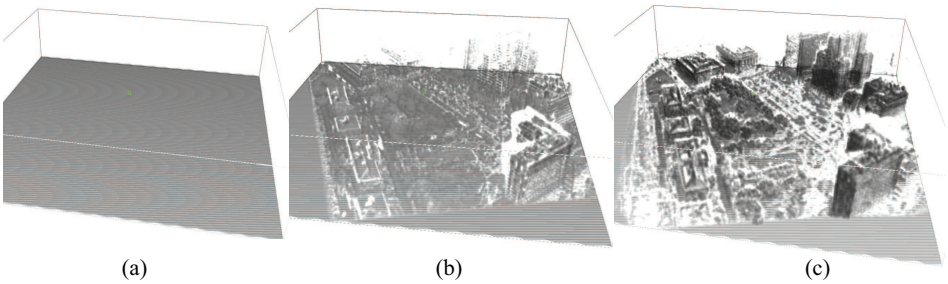
Figure 4: Flowchart of the camera pose optimization algorithm.

refined state estimate. The refined state $\hat{x}_k^+$ is used in place of $\hat{x}_k$ as a prediction for the next iteration of the Extended Kalman filter.

Note that this refinement process is essentially a novel application of Drummond and Cipolla's [5] visual servoing algorithm to camera calibration. Rather than providing feedback to a physical servoing system, it is the camera estimate that is being adjusted. Since it is not possible to capture real images from the estimated viewpoints, data from previous frames is used to predict them.

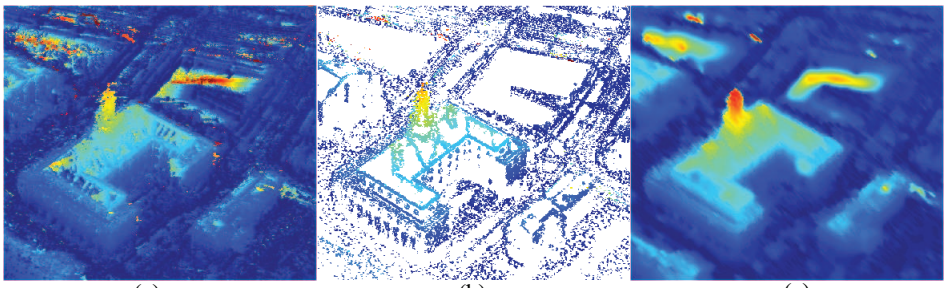
4.3.1 System Initialization

The refinement process assumes that the voxel grid has already been populated with enough information to generate reasonably accurate expected images. In order to allow the system to "bootstrap" itself, an estimate of the ground plane is provided upon initialization. The occupancy probabilities are then initialized using a normal distribution as $P(X) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{d^2}{2\sigma^2}}$, where d is the distance of the voxel center to the plane, and σ is a parameter set based on the certainty of the ground plane estimate. Because of this planar initialization, registration errors due to parallax can be seen in the first few frames until the occupancy probabilities converge (Figure 5).



(a) (b) (c)

Figure 5: Volume renderings of the voxel occupancy probabilities for the "downtown" sequence. The higher a voxel's occupancy probability, the more opaque it is drawn. (a) frame 0 (b) frame 25 (c) frame 100.



(a) (b) (c)

Figure 6: (a) The generated heightmap for a frame of the "Steeple St." sequence. (b) The heightmap values associated with confidence values above threshold. (c) Using the high confidence heightmap values as boundary conditions, a smoothed heightmap is generated using the heat equation.

5 Registered Frame Rendering

Once the camera pose for a frame is determined, the next task is to render the registered image. Registered images are essentially re-renderings of the original frames from a stationary camera. The rendering algorithm can be thought of conceptually as three steps:

1. Generate a voxel heightmap from the virtual camera viewpoint
2. Backproject data from optimized camera into the voxel grid
3. Reproject data to the virtual camera.

The goal of Step 1 is to determine the most likely voxel $\hat{X}$ that produces the intensity at each pixel in the registered image. If R is the camera ray corresponding to the pixel,

$$\hat{X} = \arg\max_{X \in R}(P(X \in S)\,\mathrm{vis}(X)) \quad , \quad c_{\hat{X}} = P(\hat{X} \in S)\,\mathrm{vis}(\hat{X}) \tag{9}$$

where $c_{\hat{X}}$ is the corresponding confidence associated with voxel estimate $\hat{X}$. A heightmap is then generated which contains the z value of the most likely voxel $\hat{X}$ at each pixel, and a confidence map which holds the corresponding $c_{\hat{X}}$ values.

Pixels with low confidence tend to be noisy and need to be filtered. The smoothing is formulated as a heat equation problem, using the heightmap pixels whose corresponding confidence values are above a threshold as the boundary values. Alternatively, the smoothing can be formulated as a least-squares fitting problem which uses all confidence values as weights. As can be seen in Figure 6, areas of homogeneous intensity tend to be associated with low confidence values. Typically, heightmap values at textured regions and edges are propagated to the homogeneous areas.

For each pixel in the registered image, a corresponding pixel in the original image can be found using the position of the corresponding $\hat{X}$, and the camera estimate $\hat{x}_k^+$. It is possible, however, that $\hat{X}$ is occluded in the original image. This case is detected by thresholding the visibility probability $\mathrm{vis}(\hat{X})$ from the point of view of the optimized camera. If $\mathrm{vis}(\hat{X})$ falls below the threshold, the expected intensity (Equation 1) is used in place of a pixel value from the original image.

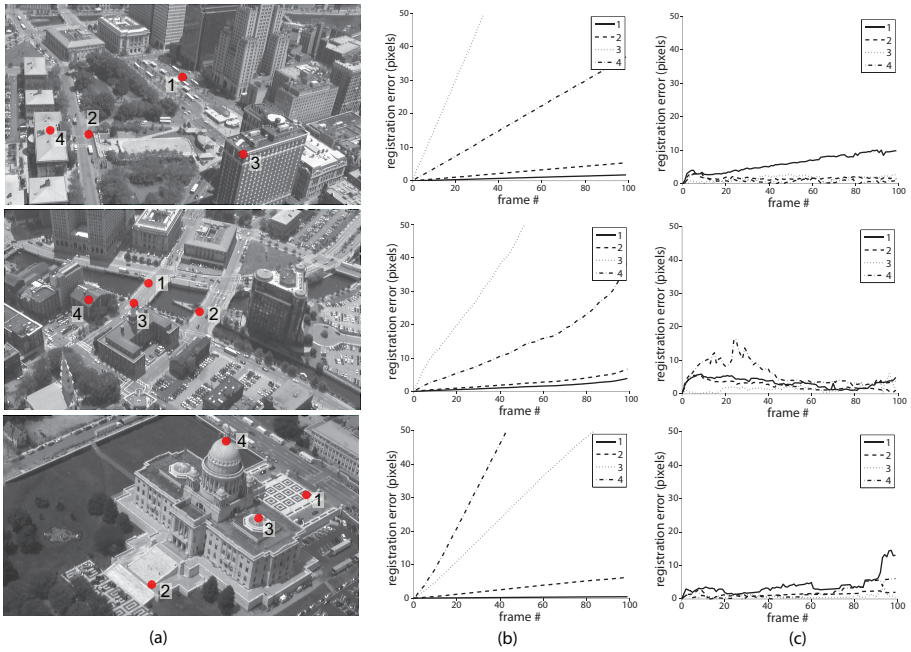


Figure 7: Column (a): first frames of three sequences (from top to bottom): "downtown", "Steeple St.", and "capitol". Four test points are marked in each. Column (b): registration error of the test points using 2-d ground plane registration. Column (c): Registration error with proposed system.

6 Results and Future Work

The system was tested on three aerial videos as shown in Figure 7. The videos are greyscale, with resolution 1280×720 and captured at 30 fps. The "Steeple St." sequence contains every tenth frame of the original sequence. Using 2-d ground plane registration (based cameras and ground plane manually calibrated to obtain ground truth), points near the ground plane are registered with high accuracy. Points off of the ground plane, however, exhibit large parallax motions as the camera changes viewpoint. Using the proposed registration system, points both on and off the ground plane are registered with accuracy comparable to the ground points in the 2-d case. Ground-plane estimates on the order of five meter accuracy were provided to the initialization procedure. As can be seen in Figure 7, the registration error is in general much lower using the proposed system.

Some rendering artifacts do exist in the registered videos which do not exist in the 2-d registration. Future work will involve removing these artifacts. Further work will also be focused on implementing the system to run in real-time. The nature of the current implementation makes it an ideal candidate for implementation on the GPU, which could potentially provide order of magnitude speed-ups. Another potential improvement in efficiency could be realized by storing the voxel data in a more efficient manner. Currently, the data for each voxel is stored on disk. Most of the voxels in a typical model, however, converge quickly to very low occupancy probabilities $P(X \in S) \approx 0$. An efficient data structure [8] could provide large savings in storage and allow regions with fine structural detail to be modeled with increased resolution.

References

[1] Gulcin Caner, A. Murat Tekalp, Guarav Sharma, and Wendi Heinzelman. Local image registration by adaptive filtering. *IEEE Transactions on Image Processing*, October 2006.

[2] A. Chiuso, P. Favaro, H. Jin, and S. Soatto. "mfm": 3-d motion from 2-d motion causally integrated over time. In *Proceedings of ECCV*, 2000.

[3] DARPA. ARGUS-IS (BAA 07-23) Proposer Information Package. `http://www.darpa.mil/ipto/solicit/baa/BAA-07-23_PIP.pdf`, February 2007.

[4] A. J. Davidson. Real-time simultaneous localisation and mapping with a single camera. In *Proceedings of ICCV*, volume 2, pages pp.1403–1410, October 2003.

[5] Tom Drummond and Robert Cipolla. Application of lie algebras to visual servoing. *International Journal of Computer Vision*, 37(1):pp 21–41, 2000.

[6] Andrew W. Fitzgibbon and Andrew Zisserman. Automatic camera recovery for closed or open image sequences. In *ECCV*, pages 311–326, London, UK, 1998. Springer-Verlag.

[7] J. A. Goncalves and A. R. S. Marcal. Automatic ortho-rectification of aster images by matching digitial elevation models. *LNCS: Image Analysis and Recognition*, 4633/2007, 2007.

[8] Philippe Lacroute and Marc Levoy. Fast volume rendering using a shear-warp factorization of the viewing transformation. In *SIGGRAPH*, 1994.

[9] Sebastien Leprince, Sylvain Barbot, Francois Ayoub, and Jean-Philippe Avouac. Automatic and precise orthorectification, coregistration, and subpixel correlation of satellite images, application to ground deformation measurements. *IEEE Transactions on Geoscience and Remote Sensing*, 45(7), June 2007.

[10] S. Maybank and O. D. Faugeras. A theory of self-calibration of a moving camera. *International Journal of Computer Vision*, 8(2):pp.123–151, 1992.

[11] Thomas Pollard and Joseph Mundy. Change detection in a 3-d world. In *Computer Vision and Pattern Recognition*, 2007.

[12] M. Pollefeys, L. Van Gool, M. Vergauwen, F. Verbiest, K. Cornelis, J. Tops, and R. Koch. Visual modeling with a hand-held camera. *IJCV*, 59(3):pp.207–232, 2004.

[13] Alex Rav-Acha, Yael Pritch, and Shmuel Peleg. Online video registration of dynamic scenes using frame prediction. *LNCS Dynamical Vision*, 4358:151–164, 2007.

[14] H.S. Sawhney, A. Arpa, R. Kumar, S. Samarasekera, M. Aggarwal, S. Hsu, D. Nister, and K. Hanna. Video flashlights - real time rendering of multiple videos for immersive model visualization. In *Thirteenth Eurographics Workshop on Rendering*, 2002.

[15] Greg Welch and Gary Bishop. An introduction to the kalman filter. Technical Report TR 95-041, University of North Carolina at Chapel Hill, July 2006.

[16] Gehua Yang, Charles V. Stewart, Michael Sofka, and Chia-Ling Tsai. Registration of challenging image pairs: Initialization, estimation, and decision. *PAMI*, 29(11), November 2007.

[17] Gouqing Zhou, Chaokui Li, and Penggen Cheng. Unmanned aerial vehicle (uav) real-time video registration for forest fire monitoring. In *Proceedings of 2005 IEEE International Geoscience and Remote Sensing Symposium*, volume 3, pages 1803–1806, 2005.

[18] Guoqing Zhou, Weirong Chen, John A. Kelmelis, and Deyan Zhang. A comprehensive study on urban true orthorectification. *IEEE Transactions on Geoscience and Remote Sensing*, 43(9):2138–2147, September 2005.

[19] Barbara Zitová and Jan Flusser. Image registration methods: A survey. *Image and Vision Computing*, 21:977–1000, 2003.