Preprint

Constructing Image Mosaics Using Focus Based Depth Analysis

Mohammad Helala Faisal Z. Qureshi
University of Ontario Institute of Technology, Oshawa, ON L1H 7K4 Canada
{Mohammad.Helala,Faisal.Qureshi}@uoit.ca

Abstract

Image alignment techniques have gained popularity for constructing image mosaics from video sequences. These image alignment techniques, however, have a hard time dealing with motion parallax, which limits their applicability. This paper studies image mosaicing in the presence of motion parallax and develops a new algorithm for generating view dependent image mosaics from low-flying aerial video sequences exhibiting strong parallax effects. Specifically we develop an energy minimization framework that computes a dense depth map of the scene from a sequence of images (captured by an uncalibrated camera following an unknown trajectory), which in turn can be used to generate a panoramic mosaic through view interpolation. We evaluate our algorithm on real and synthetic aerial video sequences and show that the proposed algorithm can construct high quality image mosaics even in the presence of strong parallax.

1. Introduction

Image mosaicing is the process of stitching together a set of overlapped images to generate a *larger* image. It overcomes the Field-of-View (FOV) limitations of physical cameras; multiple overlapped images can be combined to generate a panoramic image providing an overview of the entire area. Traditional mosaicing techniques constrain camera motion to straight lines [28, 19] or assume a planar scene [23, 22, 2] to construct perceptually plausible image mosaics. These techniques in particular have a hard time dealing with input imagery that contains strong parallax effects. In the presence of parallax, for example, these methods exhibit undesirable artifacts, such as ghosting, blurring, object repetition, etc. [28, 9].

Constructing a high-quality, artifact-free image mosaic from a video sequence exhibiting strong parallax is a challenging problem that finds its use in several applications such as video editing [18], video re-targeting [26], scene understanding [3], aerial imagery [9] etc. Recent image mosaicing techniques attempt to deal with parallax by 1) restricting the camera motion [1], 2) using adaptive manifolds [16], 3) dividing the scene into planar sub-scenes [6], or 4) recovering dense depth maps [4, 12, 28, 9]. Image mosaicing is especially relevant within the context of aerial imagery, where one is oftentimes faced with the challenge of visualizing and understanding the image data captured over vast areas by one or more aerial vehicles. For high-altitude image capture [15], the perceived depth variation between different scene structures is negligible and the motion parallax is barely noticeable. Here we can assume the scene to be planar, greatly simplifying mosaic construction. On the other hand, depth variation is very noticeable for images captured by an aerial vehicle flying at low altitudes, say a helicopter flying over a metropolis comprising high-rise, multi-story buildings. These video sequences exhibit strong motion parallax, which leads to occlusions between successive frames. Naïve image stitching alone is not sufficient to construct high-quality image mosaics in this case.

Within the context of image mosaics, depth map recovery appears an important first step for dealing with object occlusions due to motion parallax. [12, 28, 9], for example, use plane sweep stereo to assign a depth label to each pixel that is matched between a sequence of input images. Occluded regions, which are areas containing mismatched pixels, are filled in using various heuristics. Still the existence of thin structures that induce sudden depth changes or texture-less (homogeneous) regions where matching fails lead to noisy depth estimates [13]. Plane sweep stereo also has a hard time dealing with sequences captured over large distances [9].

This paper proposes a new image mosaicing framework that constructs view-specific mosaics from low-flying aerial videos. The proposed method comprises of two phases: 1) pre-processing and 2) mosaic generation (Figure 1). During the first phase, Structure from Motion (SFM) is used to estimate camera trajectory and sparse depth information from the input imagery. Next the estimated camera trajectory is used to group input images into spatially local clusters containing overlapping images. For each cluster, plane sweep stereo is used to estimate dense depth information. We recast plane sweep depth assignment within the sparse cost

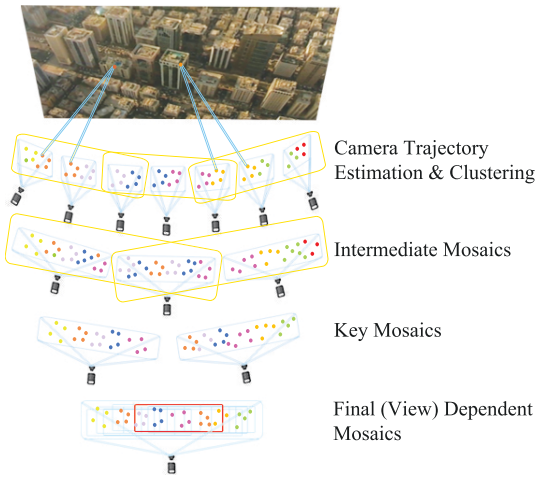

Figure 1. Overview of our proposed method.

volume filtering framework introduced by [8]. An *intermediate* image mosaic is constructed for each cluster and a second phase of trajectory estimation, clustering and plane sweep stereo over these image mosaics refines the depth assignments and creates another set of mosaics, called *key* mosaics. Final mosaic construction, or the second phase, consists of defining a reference camera and projecting intermediate and key mosaics onto the reference camera. We formulate this projection procedure as an energy minimization problem, whose solution selects the best patch (the most fronto-parallel) from each intermediate or key mosaic to be included in the final mosaic. We evaluate the proposed method on both real and synthetic data and the results appear promising.

The contributions of this work are threefold. First, we present a new method for creating image mosaics from low-flying aerial footage that exhibits strong motion parallax. Our method can deal with arbitrary camera motions and can construct image mosaics from previously unseen viewpoints through view interpolation (within reason, of course). Second, we develop an energy minimization extension for the plane sweep stereo algorithm, that appears well-suited for dealing with scene irregularities due to sudden depth changes. Lastly, we also formulate image stitching (for final mosaic construction) as an energy minimization problem. We would like to emphasize that our method eschews *a priori* scene information, such as camera calibration, scene depth information, etc.

2. Related Work

Mosaic construction usually goes through three main steps: frame registration [23, 24], intensity blending, and compensation for motion parallax and illumination change. Several techniques have been proposed to construct a video mosaic, which can be classified in static or dynamic [11]

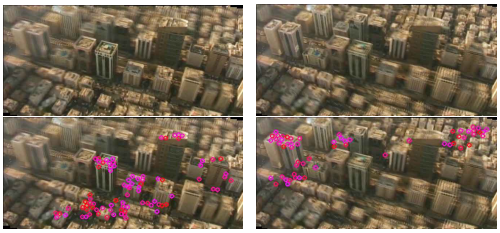

Figure 2. 3D points returned by the SFM routine are used to identify the sparse set of sub-volumes defined by [8]. (Top row) Average images for sweep planes at depths 50 (Left) and 58 (Right). (Row 2 and 3) SFM points shown as Red circles define salient regions around their locations. Here white pixels belong to the salient regions for depths 50 (Left) and 58 (Right).

based on the existence of moving objects in the scene. Alkaabi et. al. [1] present an iterative corner extraction and matching method for aligning images in two-dimensional mosaics. Peleg et. al. [16] investigate manifold mosaicing for more general cases of camera motion. Steedly et. al. [22] divide video frames into key or intermediate frames based on the amount of overlapping. Their method estimates orientation homographies using extracted keypoints which are then used to register frames to a common mosaic. Marzotto et. al. [14] investigate global mosaic registration with high resolution. Their technique constructs a mosaic by chaining homographies over a graph which has the registered frames as vertices and homographies as edges. All of the above neglect the existence of parallax effects and assume planar scenes for constructing the mosaic.

Motion parallax can cause objects misalignment or ghost effects in the output mosaic. Gorges et. al. [6] present a technique for handling parallax effects by decomposing the scene into planar regions using stereo information. These regions are then tracked and warped into the output mosaic by avoiding the occluded regions resulting from parallax. Zhi and Cooperstock [28] incorporate the idea of virtual dense sampling for dealing with parallax problems. This technique assumes the existence of several stationary cameras viewing a scene with a known pose information. A virtual camera with a wide FOV is defined by interpolating the poses of the source cameras and a mosaic image is defined by depth estimation using a plane sweep algorithm. The idea of time warping is investigated by Rav-Acha et. al. [19] for constructing mosaics from images captured by a translational camera. This technique works on space time volumes and estimates the motion and shape parameters to provide depth invariant alignment between successive video frames.

Recent advances in aerial imagery applications rely on Unmanned Aerial Vehicles (UAVs) for capturing images. Most of these applications require the construction of aerial maps of wide-FOV [25, 17, 27, 29]. For example, Rinner et. al. [25, 17] demonstrate a technique that constructs

an aerial mosaic map for monitoring a fire drill using images captured by multiple UAV vehicles. Yuping et. al. [27] perform mosaicing by using a reference image map to register a UAV image sequence. Zhu et. al. [29] addressed the parallax problem for translational dominant airborne cameras. Their technique constructs stereoscopic mosaics using a parallel perspective representation on rectified images.

In summary, recent techniques try to overcome the parallax effects by either restricting the camera motion (usually a translational motion model) or assuming a scene with small depth variations. Our proposed technique addresses these issues by analyzing scene depth.

3. Phase 1: Data Processing

As stated earlier the proposed technique consists of two phases. The first phase processes raw imagery (captured from an uncalibrated camera moving along an unknown trajectory) and constructs intermediate and key mosaics, which are subsequently used in the second phase to construct several view-specific mosaics. We now describe the first phase.

3.1. Plane Sweep Stereo

Plane sweep makes use of the "depth of focus" property for scene depth estimation. It discretizes the 3D space into a set of sweep planes that lie parallel to the image plane of a reference camera [12, 28, 9]. For each sweep plane an average image I is constructed by projecting the source images $\mathcal{I} = \{I_i | i = 1..n\}$ onto it. It is easy to project a source image onto the sweep plane at a depth, say d, by finding the homography that relates the source image to the sweep plane at that depth. These homographies can be constructed via some suitable SFM technique [21]. Indeed we use SFM to estimate the camera trajectory and sparse 3D scene structure.

An object in the scene will be in focus (i.e., appear sharp) only when it is projected onto the plane whose depth is equal to the depth of this object. Consequently, focus analysis can help us estimate scene depth. A common way to identify if a particular patch in the source images is in focus at a certain depth d is to project these patches onto the sweep plane at that depth and compute the color variations in these projections. If the patch is in focus at depth d then the color variations for these projections will be negligible.

Plane sweep is computationally expensive, so we must identify a parsimonious set of depth or disparity planes to consider. We employ Expectation-Maximization (EM) to estimate the disparity range from the 3D points returned by our SFM routine [12]. The disparity range is iteratively subdivided to select a set of disparities for which sweep planes will be considered. This process continues until the source camera projections onto the successive sweep planes show little change as determined by a predefined threshold.

3.1.1 Sparse Cost Volume Filtering

The plane sweep algorithm recovers a dense depth map for the scene by defining a label assignment problem $l : I \rightarrow L$, which assigns to each pixel p in the image I, a disparity label from the set of disparities L selected through the iterative subdivision mentioned before. For each sweep plane l, we project all source images in $\mathcal{I}$. This constructs a 3D space image V that has $V(x, y, l) = \{I_i^l(x, y)\}_{i=1}^n$, I_i^l is the projection of source image $I_i \in \mathcal{I}$ to plane l, and $I_i^l(x, y)$ is the projected color at location (x, y). In order to solve this label assignment problem, we extend the sparse cost volume filtering method of [8] to multi-view plane sweep stereo. We select this method for the following reasons: 1) local Cost Volume Filtering (CVF) [10] provides a fast alternative to traditional global energy minimization methods with comparable accuracy; 2) CVF employs Edge-Aware Filtering (EAF) [7] which we think essential to handle the large color variance of multiple views; thereby preserving the intensity changes of a selected guidance image from $\mathcal{I}$; 3) The method of [8] provides up-to 4 times speedup over the traditional CVF method while providing similar accuracy.

A cost volume $C(x, y, l)$ is defined to store the cost of assigning a label l to each pixel (x, y). Each slice in the cost volume corresponds to a sweep plane, and the cost at location (x, y) is defined by extending the formula used in [8] for multi-views as,

$$C(x, y, l) = \sum_{i=1}^{n} ((1 - \beta)(\min(d_1, \gamma_1)) + \beta \min(d_2, \gamma_2)) \tag{1}$$

where $d_1 = |\bar{I}^l(x, y) - I_i^l(x, y)|$, $d_2 = |\nabla_x \bar{I}^l(x, y) - \nabla_x I_i^l(x, y)| + |\nabla_y \bar{I}^l(x, y) - \nabla_y I_i^l(x, y)|$ and $\beta \in [0, 1]$. γ_1 and γ_2 are user defined thresholds. $\bar{I}^l(x, y)$ is the average of the projected colors at $V(x, y, l)$, ∇ represents the horizontal or vertical gradient at the projected image location $I_i^l(x, y)$, and $\nabla \bar{I}^l(x, y)$ is the average gradient defined similar to $\bar{I}^l(x, y)$.

The method of [8] defines a set of sparse sub-volumes inside the cost volume $C(x, y, l)$. This method relies on the assumption that each sweep plane can be partitioned into visible and invisible regions. the visible regions are the locations where scene objects are in focus or appear sharp. The invisible regions are out of focus or distorted locations. The sub-volumes are defined as salient 3D cuboids in the cost volume that surrounds the visible regions from all sweep planes. To locate the sub-volumes, [8] initially applies feature matching to find a set of keypoints with known disparity labels l'. The keypoints are used as seeds inside the cost volume, and sub-volumes are defined around them using two parameters u and r. The u parameter controls the expansion of a sub-volume in the label space by considering the sweep planes with $l - l' < u$. The r parameter is a

fraction (usually set to 0.3) that defines the width and height of a sub-volume in the image space as $r \times I_{\text{width}}$.

In this work, we use the 3D points returned by SFM as seeds (see Figure 2) and define a set of sub-volumes S using [8]. EAF is then applied and restricted to the sub-volumes by processing each slice l in a sub-volume $S_i \subset C$ as follows,

$$S'_i(x, y, l) = W_{I_{\text{guide}}(x,y)} \otimes S_i(x, y, l), \tag{2}$$

where $\otimes$ denotes the convolution operator, $S'_i(x, y, l)$ represents the filtered costs, and $W_{I_{\text{guide}}(x,y)}$ are weights calculated for each pixel (x, y) using a guidance image. In this work, we use the guided filter [7] for defining the weights, where the guided image is defined as the projection of the middle image from the source images $\mathcal{I}$. This filtering step defines a set of filtered sub-volumes S'. As a final step, a *winner-takes-all* strategy is used to assign each pixel (x, y) in the reference camera image plane, a disparity label $l^*_{(x,y)}$, such that,

$$l^*_{(x,y)} = \arg\min_l \{S'_i(x, y, l)\ \forall S'_i \in S' \text{ and } (x, y) \in S'_i\}. \tag{3}$$

3.1.2 Occlusion Handling and Hole Filling

It is easy to recover the mosaic (i.e., the reference image I_{ref}) given the depth information for each pixel. For each pixel $p \in I_{\text{ref}}$, we copy the color at the corresponding location of the fronto-parallel sweep plane at depth l^*_p. In order to locate occluded regions due to parallax effects or noisy estimation, we apply a *forward-backward consistency check* similar to [10]. Here, we reapply equations 2 and 3 using a guidance image (different from the previously selected one) chosen from the source images $\mathcal{I}$. This results in another depth map defined for the the reference camera image plane. Pixels with inconsistent label assignment are defined as occluded. These pixels are not copied into the reference image, resulting in gaps. Similar to [9, 28], we employ a row scanning algorithm that gives precedence to nearer objects to fill the gaps present in the reference image.

3.2. Intermediate and Key Mosaics

We are interested in constructing a dense depth map of the scene, as it is possible to construct perceptually correct image mosaics from previously unseen view points via view interpolation given depth information of the scene. In the previous sections, we introduced our version of the plane sweep stereo algorithm that casts depth assignment within the sparse cost volume filtering framework. We noticed, however, that the plane sweep algorithm fare poorly on longer image sequences, i.e., image sequences that are captured by a moving camera over large areas. Projections of source images taken far from the reference camera (that sets up the plane sweep) are sensitive to calibration errors.

Figure 3. Intermediate (top row) and key (bottom row) mosaics generated from the Dubai sequence. The circle represent the SFM key points.

We solve this issue by borrowing the idea of key and intermediate mosaics that was proposed by [22]. Here the input image sequence is grouped into clusters of overlapping images. It is straightforward to construct such a grouping by relying upon the estimated camera trajectory. We construct a mosaic for each cluster using the plane sweep formulation described above. We call these mosaics *intermediate* mosaics. Next we use the intermediate mosaics to estimate a (virtual) camera trajectory and use it to group the intermediate mosaics into clusters of overlapping images. Another set of mosaics, called the key mosaics, are constructed for each of these clusters using the plane sweep stereo. For each intermediate and key mosaic, we also keep track of the locations of the (SFM) 3D points that were used in the previous section to select salient regions. We refer to these locations as keypoints. Figure 3 shows example intermediate and key mosaics, plus their keypoints, for the Dubai sequence.

We shall see in the next section that given the intermediate and key mosaics, plus the associated depth maps and the locations of the key points enable us to generate gap free mosaics via view interpolation.

3.3. Phase 2: View-Specific Mosaic via Stitching

Now we discuss the second phase of the proposed method. Given a set of intermediate and key mosaics, along with their depth maps and the key points, we are interested in generating image mosaics from a previously unknown viewpoint. For the remainder of this section, we assume that $M = \{M_1, M_2, \cdots, M_k\}$ denotes the set of intermediate and key mosaics and $P_j = \{\mathbf{p}_j^1, \mathbf{p}_j^2, \cdots, \mathbf{p}_j^{k_j}\}$ denotes the set of key points for the j^{th} mosaic.

We construct the final mosaic for a given viewpoint as follows:

1: Construct a reference camera for the viewpoint and define an image plane for this reference camera.
2: Compute the homography that projects each mosaic M_j onto the image plane.
3: For each mosaic M_j, use the homography computed in

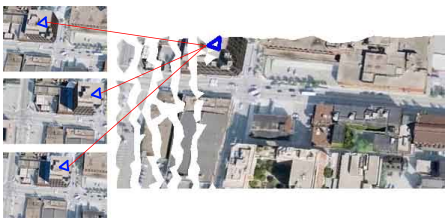

Figure 4. Energy based stitching algorithm for final view-specific mosaic generation using intermediate and key mosaics. Typically each triangular region in the reference image has multiple fill candidates in the intermediate and key mosaics.

the previous step to project its keypoints to the image plane.

4: Let $\mathbf{v}_j^i$ be the projection of the i^{th} keypoint in the j^{th} mosaic onto the image plane, we use the set $\{\mathbf{v}_j^i\}$ to construct a 2D Delaunay triangulation that divides the image plane into triangular segments (Figure 4). The goal now is to fill in these triangular segments using the available intermediate and key mosaics. We cast triangle filling as an energy minimization problem $f : T \rightarrow M$, which assigns each triangle $t \in T$ the source mosaic $M_j \in M$ from which it will be filled. Intuitively, if a triangle $t \in T$ is assigned a mosaic M_j then the triangle should be filled in using the corresponding triangular patch t^j (computed via inverse homography) from that mosaic.

We now define the data and the smoothing costs for the assignment problem mentioned above.

$$E^{\text{f}}(f) = E_d^{\text{f}}(f) + E_s^{\text{f}}(f) \quad (4)$$

$$E_d^{\text{f}}(f) = \sum_{t \in T} \left(\alpha_1 Pose(t, t^j) + \alpha_2 Area(t, t^j)\right) \quad (5)$$

$$E_s^{\text{f}}(f) = \sum_{\substack{t,t' \in T \\ t' \in Adjacent(t)}} \omega Boundary(t, t'), \quad (6)$$

where α_1, α_2 and ω are used defined parameters. $Pose(t, t^j)$ returns the angle between the optical axis of the reference camera (whose image plane contains t) and the optical axis of the camera for mosaic M_j, which contains t^j, $Area(t, t^j)$ returns the difference in the areas of the two triangles t and t^j, and $Boundary(t, t')$ function returns the sum of squared difference of pixel colors of the shared boundary between two adjacent triangles t and t'. The set $Adjacent(t)$ consists of the three triangles that share an edge with t. By definition $Boundary(t, t')$ is zero when the two triangles come from the same key or intermediate mosaic. Remember also that t and t^j are related by the homography that projects M_j to the image plane.

The data cost term (Equation 5) encourages triangles to be filled from mosaics that are most fronto-parallel to the reference camera. This is achieved by minimizing the distance between the pose of selected mosaics to the reference camera and choosing a mosaic with a filling area that best match the area of the corresponding triangle in t. The smoothness cost term (Equation 6) encourages neighboring triangles to be selected from the same mosaic and matches the boundary pixels between two adjacent triangles.

4. Results

The proposed algorithm is tested on synthetic and real video sequences. The synthetic sequences are captured by "simulating" an aerial vehicle flying over the city of Toronto within the Google Earth environment. [9] also make use of Google Earth as tool for studying aerial mosaics. The real video was recorded by an airborne camera flying over the city of Dubai. We do not have the camera calibration or scene depth information for these sequences.

Our main focus is to demonstrate that the proposed technique is able to construct high-quality aerial mosaics in the presence of motion parallax; therefore, both synthetic and real video sequences that we use to evaluate our technique exhibit strong motion induced parallax. Table 1 provides an overview of the two tests, recording for each input sequence its length and dimensions, the depth range estimates returned by the EM, and the number of sweep planes constructed. For cost volume filtering, we use the same parameters used in [10, 8]. We also set the sub-volume parameters $r = 0.3$ and $u = 4$. We empirically set the values of $\{\alpha_1, \alpha_2, \omega\}$ parameters to $\{0.5, 0.5, 1\}$, and cluster size $= 3$ for both intermediate and key mosaics. In the future, we plan to study the sensitivity of our method to these parameters.

Table 2 shows a quality comparison of the generated mosaics against selected ground truth frames, by calculating the average sum of square error in the frequency domain. We select the frequency domain as it encodes shape information and is sensitive to distortions. The compared methods include: 1) Our formulation of plane sweep stereo using sparse cost volume filtering; 2) the common plane sweep stereo with GraphCut (GC) optimization; 3) the GraphCut method in 2) while using Clustering Views for Multi-view Stereo (CMVS) [5] patches as salient regions in the cost volume and penalizing (Adding extra cost = 100) other regions. The mosaic quality comparison is performed on a dataset that includes the first four frames of the Toronto-3 and Dubai sequences, and the two frames of the teddy and cones dataset [20].

4.1. Toronto3 Sequence

The Toronto3 sequence is a 2060 frames video sampled at 30 frames per second by a "simulated" aerial vehicle following a curved path over the city of Toronto (Figure 5(a)). We uniformly select 168 frames and use these to construct

Seq.	#frames	Dimensions	Depth Range	#planes
T3	168_{2060}	772×436	[18,40]	25
Du	30_{180}	640×310	[40,90]	25

Table 1. Test sequences. T3 refers to the Toronto3 sequence. Du refers to the Dubai sequence. The depth range for each sequence is estimated via EM. Column 2 refers to the number of frames used to construct the aerial mosaics. The number in the subscript indicates the total length of the video sequence, which we sample uniformly to select the frames used for mosaic creation.

Our Method	Common GC	GC+CMVS Patches [5]
0.014	0.032	0.028

Table 2. A quality comparison of the generated mosaics of our method, common plane sweep stereo with GraphCut (GC) optimization, plane sweep stereo with GraphCut (GC) using the CMVS patches as salient regions. For each comparison, we measure the average sum of square error by comparing against ground truth images in the frequency domain. we can see that our formulation of salient regions provides the lowest error values.

the mosaic. Similar to Toronto2 sequence, motion parallax is clearly visible here as well. Figure 5 (top row, right) show four sample input frames, where row 2 and 3 show the generated aerial mosaics from two different viewpoints. The proposed method is able to construct an artifact-free aerial mosaic from an uncalibrated image sequence that exhibits strong motion parallax.

4.2. Dubai Sequence

We also evaluate our method on a 180 frames real video sequence captured by an airborne camera flying over the city of Dubai. We uniformly select 30 frames from this sequence to construct the aerial mosaic. This sequence also exhibits strong motion parallax; Dubai too has many high-rise buildings. Figure 6 (top row) show three sample input images. In addition to parallax, vehicular traffic is also visible in this sequence. Still the proposed method is able to construct aerial mosaics from this video sequence (Figure 6 (row 2 and 3)). Again we show two aerial mosaics from different view points.

4.3. Conclusion

In this paper we explored a new video mosaicing framework that can handle parallax effects in aerial video sequences. The framework does not constrain the camera motion or assume a planar scene. Our method is able to generate image mosaics from previously unseen viewpoints through view interpolation. To this end we have developed an extension to the plane sweep algorithm for multi-views that casts depth label assignment within sparse cost volume filtering. Furthermore, we make use of the sparse depth information computed using SFM techniques to guide the depth label assignment during plane sweep. We also cast the image stitching step responsible for creating the final

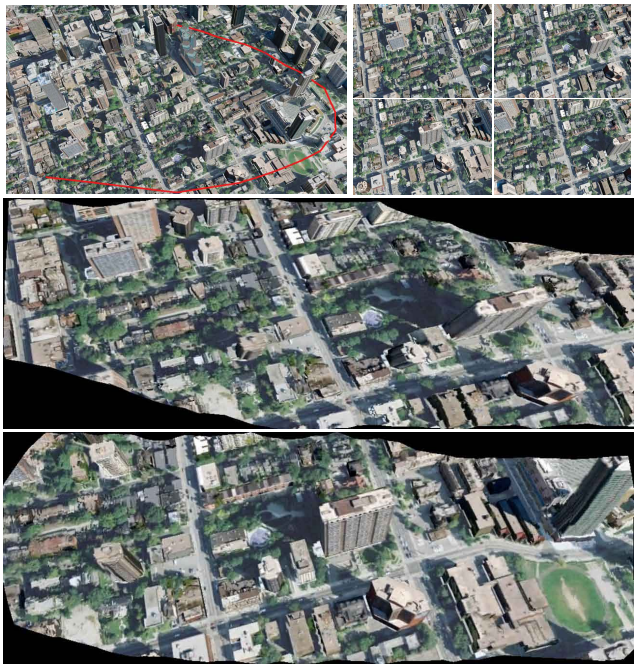

Figure 5. Toronto3 synthetic sequence. 2060 frame video was recorded using Google Earth software along the trajectory (Red) shown above. We evenly sampled this video to get 168 images, which were used to construct the two mosaics shown here. Each mosaic is taken from a different view point. *This figure is best viewed in color.*

view-specific mosaic as an energy minimization problem.

The proposed framework is evaluated on near-ground aerial footage (both synthetic and real) exhibiting strong parallax, and the results suggest that the technique is suitable for generating artifact-free mosaics from such footage. Vehicular traffic is visible in one of the sequences; however, we make no claims that the proposed method can handle imagery containing moving objects. We plan to address this shortcoming in the future. We are also interested in high-performance implementation of our scheme, which is urgently needed as we deal with larger input sequences.

References

[1] S. Alkaabi and F. Deravi. Iterative corner extraction and matching for mosaic construction. In *CRV*, pages 468 – 475, Victoria, BC, Canada, May 2005. 1, 2

[2] M. Brown and D. Lowe. Recognising panoramas. In *ICCV*, pages 1218–25, Nice, France, Oct. 2003. 1

[3] L. Chen, Y. Lai, and H. Liao. Video scene extraction using mosaic technique. In *Proc. ICPR*, pages 723–726, Hong Kong, Aug. 2006. 1

[4] R. Collins. A space-sweep approach to true multi-image matching. In *CVPR*, pages 358–363, San Francisco, CA, USA, 1996 Jun. 1

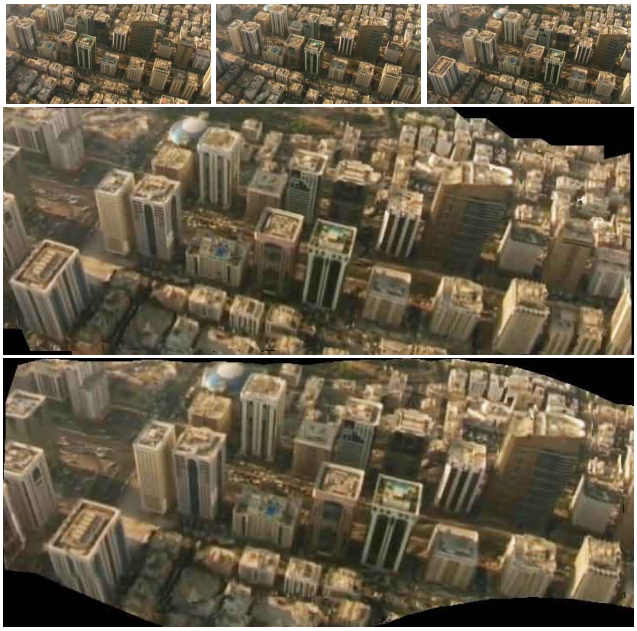

Figure 6. Dubai sequence consists of 180 frames. We uniformly selected 30 frames to construct the mosaics shown here. The first row show three input images, and the second and the third row show two aerial mosaics constructed by the proposed method. Here each aerial mosaic is from a different view point.

[5] Y. Furukawa and J. Ponce. Accurate, dense, and robust multiview stereopsis. *IEEE TPAMI*, 32(8):1362–1376, 2010. 5, 6

[6] N. Gorges, M. Hanheide, W. Christmas, C. Bauckhage, G. Sagerer, and J. Kittler. Mosaics from arbitrary stereo video sequences. In *Pattern Recognition*, volume 3175 of *Lecture Notes in Computer Science*, pages 342–349. Springer Berlin / Heidelberg, 2004. 1, 2

[7] K. He, J. Sun, and X. Tang. Guided image filtering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(6):1397–1409, June 2013. 3, 4

[8] M. A. Helala and F. Z. Qureshi. Accelerating cost volume filtering using salient subvolumes and robust occlusion handling. In *ACCV*, Nov 2014. 2, 3, 4, 5

[9] M. A. Helala, L. A. Zarrabeitia, and F. Z. Qureshi. Mosaic of near ground uav videos under parallax effects. In *ICDSC*, pages 1–6, Hong Kong, Nov. 2012. 1, 3, 4, 5

[10] A. Hosni, C. Rhemann, M. Bleyer, C. Rother, and M. Gelautz. Fast cost-volume filtering for visual correspondence and beyond. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(2):504–511, 2013. 3, 4, 5

[11] M. Irani, P. Anandan, and S. Hsu. Mosaic based representations of video sequences and their applications. In *ICCV*, pages 605–611, Boston, MA, USA, June 1995. 2

[12] S. Kang and R. Szeliski. Extracting view-dependent depth maps from a collection of images. *Int. J. Comput. Vision*, 58(2):139–163, Jul. 2004. 1, 3

[13] A. Kowdle, N. Snavely, and T. Chen. Recovering depth of a dynamic scene using real world motion prior. In *ICIP*, pages 1209–1212, Florida, USA, Oct. 2012. 1

[14] R. Marzotto, A. Fusiello, and V. Murino. High resolution video mosaicing with global alignment. In *CVPR*, volume 1, pages 692–698, Los Alamitos, CA, USA, June 2004. 2

[15] E. Molina and Z. Zhu. Persistent aerial video registration and fast multi-view mosaicing. *IEEE Transactions on Image Processing*, 23(5):2184–2192, May 2014. 1

[16] S. Peleg, B. Rousso, A. Rav-Acha, and A. Zomet. Mosaicing on adaptive manifolds. *IEEE TPAMI*, 22(10):1144 – 1154, Oct. 2000. 1, 2

[17] M. Quaritsch, R. Kuschnig, V. Mersheeva, D. Wischounig-Strucl, S. Yahyanejad, E. Yanmaz, G. Friedrich, H. Hellwagner, C. Bettstetter, and B. Rinner. Collaborative uavs for aerial reconnaissance in rescue scenarios. In *Austrian Robotics Workshop*, Innsbruck, Austria, May 2011. 2

[18] A. Rav-Acha, P. Kohli, C. Rother, and A. Fitzgibbon. Unwrap mosaics: a new representation for video editing. In *ACM SIGGRAPH*, pages 1–11, New York, NY, USA, Aug. 2008. 1

[19] A. Rav-Acha, Y. Shor, and S. Peleg. Mosaicing with parallax using time warping. In *CVPRW*, page 164, Washington, DC, USA, june 2004. 1, 2

[20] D. Scharstein and R. Szeliski. High-accuracy stereo depth maps using structured light. In *CVPR*, volume 1, pages 195–202, 2003. 5

[21] N. Snavely, S. Seitz, and R. Szeliski. Modeling the world from internet photo collections. *Int. J. Comput. Vision*, 80(2):189–210, nov 2008. 3

[22] D. Steedly, C. Pal, and S. Szeliski. Efficiently registering video into panoramic mosaics. In *ICCV*, volume 2, pages 1300–1307, Beijing, China, Oct. 2005. 1, 2, 4

[23] R. Szelisk. Video mosaics for virtual environments. *IEEE Comput. Graph. Appl.*, 16(2):22–30, Mar. 1996. 1, 2

[24] H. Wallin, C. Christopoulos, and F. Furesjo. Robust parametric motion estimation for image mosaicing in the mpeg-7 standard. In *ICIP*, volume 2, pages 961–964, Thessaloniki, Greece, Oct. 2001. 2

[25] D. Wischounig-Strucl, M. Quaritsch, and B. Rinner. Prioritized data transmission in airborne camera networks for wide area surveillance and image mosaicking. In *CVPR*, volume 1, pages 692–698, Colorado Springs, USA, June 2011. 2

[26] T. Yen, C. Tsai, and C. Lin. Maintaining temporal coherence in video retargeting using mosaic-guided scaling. *IEEE Trans. on Image Processing*, 20(8):2339–2351, Aug. 2011. 1

[27] L. Yuping and G. Medioni. Map-enhanced uav image sequence registration and synchronization of multiple image sequences. In *CVPR*, pages 1–7, Minneapolis, Minnesota, USA, June 2007. 2, 3

[28] Q. Zhi and J. Cooperstock. Toward dynamic image mosaic generation with robustness to parallax. *IEEE Trans. on Image Processing*, 21(1):366–378, 2012. 1, 2, 3, 4

[29] Z. Zhu, A. Hanson, and E. Riseman. Generalized parallel-perspective stereo mosaics from airborne video. *IEEE TPAMI*, 26(2):226–237, Jan. 2004. 2, 3