Low-Cost Orthographic Imagery

Peter Pesti
College of Computing
Georgia Institute of Technology
Atlanta, GA 30332-0765
pesti@gatech.edu

Jeremy Elson, Jon Howell,
Drew Steedly, Matt Uyttendaele
Microsoft Research
One Microsoft Way
Redmond, WA 98052-6399
{jelson, howell, steedly,
mattu}@microsoft.com

ABSTRACT

Commercial aerial imagery websites, such as Google Maps, MapQuest, Microsoft Virtual Earth, and Yahoo! Maps, provide high- seamless orthographic imagery for many populated areas, employing sophisticated equipment and proprietary image post-processing pipelines. There are many areas of the world with poor coverage where locals might benefit from recent, high-resolution orthographic imagery, but which do not fit into the schedules and scaling model of the big sites.

This paper describes MapStitcher, a system that orthorectifies and geographically registers imagery using only low-cost capturing equipment. MapStitcher combines manually-entered relationships between images and known ground references with a MOPs-based image-stitching technique that automatically discovers image-to-image relationships. Our image registration pipeline first extracts and matches feature points, then clusters images, then uses RANSAC-initialized bundle adjustment to simultaneously optimize all constraints over the entire image set. Simultaneous optimization balances the requirements of precise stitching and absolute placement accuracy. We used this technique to image a portion of the Skagit River Valley in the vicinity of the town of Concrete, WA (pop. 790) at 0.15 m/pixel. Our technique is more accurate than stitching followed by "rubber-sheeting" (deforming the stitched image into global coordinates), while it also requires less effort and produces a better-stitched composite than rubber-sheeting images separately.

Categories and Subject Descriptors

I.4 [**Image Processing and Computer Vision**]: Scene Analysis; J.2 [**Computer Applications**]: Physical Sciences and Engineering

General Terms

Photogrammetry

Keywords

mapping, aerial photography, automatic georegistration, feature matching, image stitching

ACM GIS 08, November 5-7, 2008. Irvine, CA, USA

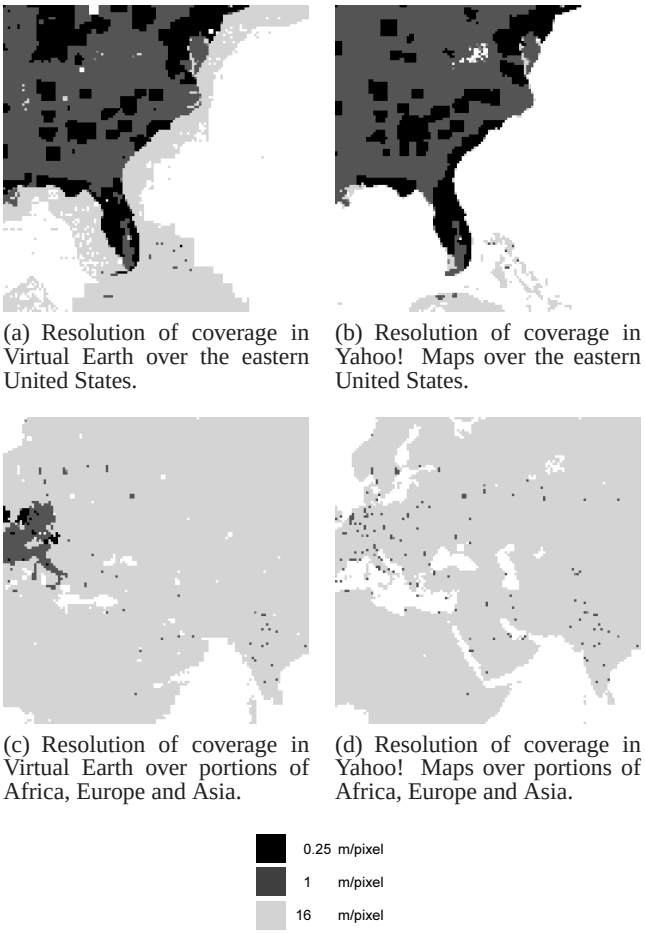
1. INTRODUCTION

Commercial aerial imagery sites, such as Google Maps, MapQuest, Microsoft Virtual Earth, and Yahoo! Maps, provide high-resolution seamless orthographic imagery for densely populated areas. To be able to image large areas in a cost-efficient manner, their techniques depend on special-purpose cameras mounted in gyro-stabilized mounts and flown in autopilot-equipped airplanes. Together, these components tightly constrain the parameters of the captured images, easing the task of post-processing the collection of images into a single orthorectified image mosaic. While allowing to amortize the cost of the system by imaging large areas, the equipment is also quite expensive; for example, the Vexcel UltraCam-D camera costs over half a million dollars. Furthermore, there are only a few competitors, and they tend to prioritize imaging populous markets. Users in small markets would also stand to benefit from access to recent, high-resolution geographically registered aerial imagery. However, it is beyond the means of small communities and other "long tail" users to purchase the expensive tools used by the large imaging operations. In addition, the post-processing pipelines used in the industry are proprietary, posing an additional barrier to entry for localized operations.

A quick survey of the image tiles available on public imagery sites reveals the lack of resolution for many regions of the Earth. For example, while most of the United States is covered at a 1 m/pixel resolution, with metropolitan areas imaged at 0.25 m/pixel (see Figure 1(a) and Figure 1(b), showing the eastern United States), other continents are mostly covered at 16 m/pixel (see Figure 1(c) and Figure 1(d), showing an area of the Earth bounded by the Equator (S), the Arctic Circle (N), 0°(W). and 90°E longitudes (E); some large cities and Western European countries have higher resolution coverage). Furthermore, the imagery update schedules of the big sites are independent of important changes in the environment, such as natural disasters, construction and demolition of roads, buildings and parking spaces. This paper describes a system designed to provide such imagery at a low cost of entry in a timely fashion: imagery is captured with a consumer-grade camera mounted on hardware-store plumbing pipe in a minimally-equipped light airplane, and post-processed with a generic pipeline that depends on a small amount of human annotation. While this approach has a higher cost per image of human annotation, the dramatically lower capital costs lead to lower overall cost for a small imaging project.

We contrast our approach with two simpler techniques for orthorectifying poorly-constrained aerial imagery.

The first approach is to simply manually annotate every captured image and then deform each image into place ("rubber-sheeting") with a tool such as MapCruncher [5]. MapCruncher scales well, allowing users to readily reproject existing maps, publishing multi-



(a) Resolution of coverage in Virtual Earth over the eastern United States.

(b) Resolution of coverage in Yahoo! Maps over the eastern United States.

(c) Resolution of coverage in Virtual Earth over portions of Africa, Europe and Asia.

(d) Resolution of coverage in Yahoo! Maps over portions of Africa, Europe and Asia.

Figure 1: Resolution of orthophoto coverage in large mapping websites.

Figure 2: Our operation: a consumer camera zip-tied to a PVC pipe protruding from a hand-flown Cessna 177.

Figure 3: Our 0.15 m/pixel composite aerial imagery, showing a portion of the Skagit River Valley near Concrete, WA, overlaid on a map of the area.

gigapixel images on the web in a client-bandwidth-friendly tiled format that interoperates with Microsoft Virtual Earth. Our experiments with this approach identified two problems: First, because the post-processing system only had information about global placement, relative inter-image placement often suffered, leading to obvious discontinuities at image boundaries. Second, where the images covered undifferentiated or entirely changed terrain, such as a construction site, there was no easy way to manually label the images with ground reference pairs.

In the second approach, the captured images are stitched into a single image of large extent using a modern photo stitching tool [2] that makes inter-image camera-pose estimates to reproject the images to eliminate boundary discontinuities. The resulting "panoramic" image represents a single theoretical image taken from a single logical viewpoint; this image is then related to ground references, and translated to a browser accessible user interface [5]. In practice, the lack of global constraints causes the photo stitcher to accumulate error and emit images that correspond to no real viewpoint of the original terrain.

This paper describes MapStitcher, an image orthorectification system that combines the two approaches above simultaneously. MapStitcher's stitching component discovers inter-image constraints. A human annotates a few images with ground reference constraints. Then MapStitcher estimates the pose of each image's camera by first initializing with RANSAC, a general technique for fitting a model in the presence of outliers. Then it uses bundle adjustment to minimize error across the entire constraint set, both relative and global. The resulting system is robust to poorly-constrained camera geometry, requires global constraints on only a small subset of images, and produces output with minimal image-boundary discontinuities.

We demonstrate MapStitcher by capturing imagery of the Skagit River Valley in the vicinity of the town of Concrete, Washington. Concrete's population of 790 has a long wait before major services will find it profitable to send a photo mission with expensive equipment. Our mission, in contrast, involved an ordinary four-seat Cessna ($160/hour rental, including pilot), three feet of PVC pipe, a consumer digital camera ($300), and two people: one pilot and one to operate the camera shutter and change the batteries (Figure 2). In post-processing, we identified 25 ground reference pairs, and used 60 photos to produce a 208 megapixel image at a resolution of 0.15 m/pixel (Figure 3).

2. RELATED WORK

The creation of aerial mosaics to form composite photomaps is described in [4]. Our method is analogous with the creation of semicontrolled mosaics, where ground reference pairs on a small number of images are combined with tie points between images to compute the transformation parameters. Our *stitch-first* control method is analogous with the creation of uncontrolled mosaics, and the *no-stitch* method is analogous with the creation of controlled mosaics. However, these digital mosaicking approaches only attempt to solve for rotation and translation parameters, assuming

vertical camera positions during image acquisition.

In order to perform digital mosaicking with less constrained cameras, the problem of estimating camera parameters must be tackled. Analytical aerotriangulation with simultaneous bundle adjustment aims to recover the 3D coordinates of object points, and the 3D location and exterior orientation parameters of all exposure stations [4]. These goals are similar to our objectives in our camera parameter estimation step. Using GPS to obtain *a priori* knowledge about the three-dimensional position of the exposure stations is a possible improvement [14]. Alternatives to bundle adjustment for solving the equations to estimate projection matrices and scene point locations are explored in [11]. For an introduction to 3D reconstruction of cameras and scene structure from photographs, we refer the reader to [7]. The problem of 3D scene reconstruction using bundle adjustment has also been explored recently in a computer vision context [1]. Bundle adjustment based methods [12] can benefit from initialization with RANSAC [6]. Specific techniques also exist for the estimation of interior [9] and exterior parameters [8] of cameras from line measurements, and for n-point camera pose determination [13].

3. GOALS OF AERIAL IMAGE COMPOSITION

Before describing MapStitcher's image processing pipeline, we first describe its design goals.

The pipeline should convert an input set of overlapping images, acquired individually, into a single virtual image that covers the same area. In constructing this composite image, we would like to simultaneously optimize for two goals. The first is geographic fidelity: features should have the correct shape in the composite image. For example, a straight road should not appear to curve in the image. The second goal is seamlessness: the boundaries between the input images should be invisible in the composite image. That is, there should not be visible discontinuities in features such as roads.

To ensure our system is practical and economical, we also have two usability goals. The first is that our pipeline should accept reasonably unconstrained input images—for example, it should not require pictures taken exactly straight down, or with cameras whose exact geometry or position is known. Such stringent requirements would significantly increase the cost of image acquisition. Our second usability goal is that the pipeline should require a minimum of user effort. A few hours of image acquisition should not be followed by weeks of manual post-processing.

In light of these goals, it is instructive to consider the weaknesses of other methods for generating a geographically accurate composite image. In this section, we will consider two that are commonly used in low-cost applications: individually "rubber-sheeting" each photo in the set, and rubber-sheeting a composite photo that was created with an image stitching tool. The main weakness of these methods is that they optimize for only one goal—geography or seamlessness—at a time.

The first method is exemplified by our previous work, MapCruncher [5], which can perform approximate Mercator reprojection of any image drawn to scale after being given a few correspondence points as exemplars. We call these points *ground reference pairs*—that is, correspondences between a pixel in an input image and a latitude and longitude in WGS84. MapCruncher has a simple interface, depicted in Figure 4, for specifying these pairs. Although surveying techniques (e.g., GPS) can be used, the fastest and easiest way is to establish ground reference pairs is to visually compare the newly acquired imagery with the existing imagery that is part of Microsoft Virtual Earth. We have found this technique useful because a typical use-case is overlaying recent high-resolution images on top of extant older or lower-resolution images. MapCruncher shows the user's images in one window and Virtual Earth in another.

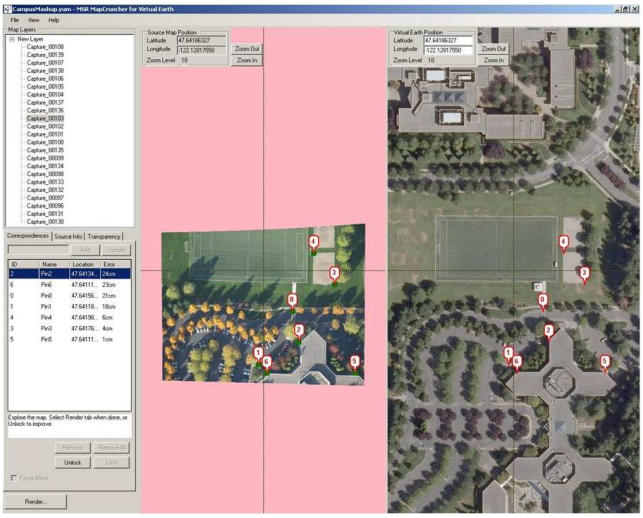


Figure 4: The user interface of both MapCruncher and MapStitcher. Users can specify ground reference pairs by finding the same feature in their own image and the standard Virtual Earth imagery. If the area has been manually surveyed, latitude and longitude can also be entered numerically.

Figure 5: When overlapping aerial images are rubbersheeted individually, discontinuities at the image boundaries are obvious.

MapCruncher was originally designed for use with maps. Our initial tests in using it for aerial image compositing were promising, but had two major drawbacks. First, MapCruncher considers the placement of each image individually, without global constraints. As a result, relative inter-image placement often suffers, causing obvious discontinuities at image boundaries, such as those shown in Figure 5. Second, where the images cover undifferentiated or entirely changed terrain, such as a new construction site, generation

Figure 6: A straight road, captured with 12 aerial photographs and mosaicked using an image stitcher. Without geographic constraints, the road appears to curve.

of ground reference pairs is difficult. The evaluation refers to this technique as *no-stitch*.

A second common approach is a two-step procedure. First, use a modern photo stitching tool [2] that makes inter-image camera-pose estimates and reprojects the images to eliminate boundary discontinuities. Next, rubber-sheet the mosaic to fit it to the depicted geography. In practice, we have found the lack of geographic constraints during the mosaic step causes the photo stitcher to accumulate error and emit images that correspond to no real viewpoint of the original terrain. For example, the mosaic shown in Figure 6 depicts about a mile of a straight north-south street, captured with a dozen individual photos shot from an airplane. Without geographic constraints, the stitcher incorrectly emits a (seamless) photo of a curving road. The evaluation refers to this technique as *stitch-first*.

4. THE MAPSTITCHER IMAGE PIPELINE

The MapStitcher image pipeline works by simultaneously combining user-specified geographic image constraints, similar to MapCruncher, and automatically generated image-stitching constraints, similar to a photo stitcher. With relatively little user effort, MapStitcher can convert a series of overlapping aerial images into a seamless, orthorectified, and geographically accurate composite. Users typically only need to specify a small number (e.g., 10) of ground reference pairs. For example, references might be set for only the first and last images in a series; the positions of intermediate images are estimated automatically using feature comparisons in the overlapping regions.

Image compositing is accomplished by first solving for the position and orientation of the camera at the moment each image was acquired. Then, each image is reprojected into an orthographic approximation and superimposed.

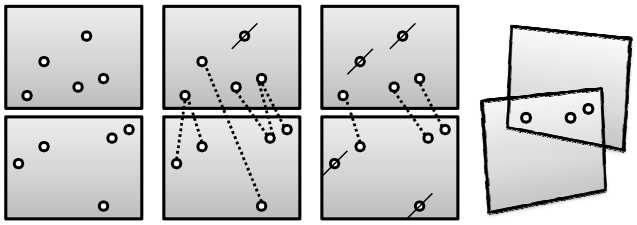
A homographic projection is used to model the view of the camera at each instant it acquires each image. Our model includes both *intrinsic* and *extrinsic* camera parameters. Intrinsic parameters are properties of the camera itself: currently just its focal length, captured in the F matrix. The extrinsic camera parameters are the translation and rotation, captured in the T and R matrices, respectively. In our model, a ground point (p_{ground}) is projected to an image point (p_{image}) according to the chained transformations:

$$q = F \cdot T \cdot R \cdot M_{pre} \cdot p_{ground},$$
$$p_{image} = M_{post} \cdot \left(\frac{q_x}{q_z} \quad \frac{q_y}{q_z}\right)',$$

where p_{ground} and q are 3D points represented as 4D homogeneous coordinates; F, T, R and M_{pre} are 4D matrices; M_{post} is a 2D matrix; and p_{image} is a 2D point. As a typical scene spans 10^{-6} equatorial circumferences in Mercator coordinates, the M_{pre} pre-transform matrix is used to scale the scene so that its size is comparable to the size of its projection on the camera's image plane, which has a largest dimension of 1.0. This scaling avoids rounding errors that lead to ill-conditioned optimizations. The M_{post} post-transform matrix ensures that the scene's projection is centered on the image plane. This centering is required to model the symmetry of the perspective projection around the center of the real camera's imaging surface.

The remainder of this section will describe how all of the camera parameters are estimated for each image acquired. Generally speaking, the procedure entails the following steps:

1. The user specifies ground reference pairs for a subset of the images to be stitched (Figure 4).
2. MapStitcher automatically finds common features in images that overlap (Section 4.1).
3. Each camera's model parameters are initialized to the "not estimated" state.
4. Iterate:
   (a) Initial estimates for camera model parameters are made for each camera in a "not estimated" state, that has sufficient ground reference pairs (Section 4.2).
   (b) Nonlinear optimization (bundle adjustment) is used to globally optimize the parameters of all cameras with estimates. Both the user-supplied ground reference pairs and constraints introduced by feature match pairs are used in this global optimzation step (Section 4.3).
   (c) Synthetic ground reference pairs are temporarily created where two images overlap, and at least one has a camera with a known model (Section 4.4). These are used to initialize camera parameter estimates in future iterations of Step 4a.
5. ... until there are no camera poses given new estimates in Step 4a.



(a) Feature points. (b) Feature matches. (c) Geometrically consistent feature matches after outlier rejection. (d) Transformed images based on established tie points.

Figure 7: Automatic establishment of feature match point correspondences between two images.

4.1 Automatic Extraction and Matching of Feature Points

MapStitcher uses Multi-Scale Oriented Patches (MOPs) [3] to identify corresponding features in the overlapping portions of adjacent images. MOPs can robustly identify features in common across images, even if they vary in scale, orientation and intensities.

The extraction of feature-matches is a five step process:

1. Interest points are identified (Figure 7(a)) on each image separately as local maxima of a "corner strength" function. The orientation of interest points is also computed.

2. The number of interest points is reduced for each image, while a uniform distribution of point locations on the image is maintained. The goal of this step is to reduce the total number of interest points, since the computational requirements for matching are superlinear.

3. A 64-dimensional feature descriptor vector is computed for each remaining interest point using the local image structure.

4. The lowest three non-zero wavelet frequencies of the feature vectors are used to create a three dimensional hash-table. This hash-table provides fast lookup for feature points. Fast approximate feature matching is performed by lookups in this hash-table: a set of approximately matching feature points are found – across all images – for each feature point. Some of the matches are eliminated as outliers using a simple heuristic (Figure 7(b)).

5. Finally, RANSAC is applied to remove additional outliers, by finding geometrically consistent feature matches (Figure 7(c)).

We refer the reader to [3] for specific details of the algorithm.

After the feature matching step is complete, MapStitcher has a list of *feature point matches* (Figure 7(d))—that is, pairs of points on overlapping photos that visually correspond to the same features on the ground.

4.2 Camera Parameter Initialization

Nonlinear estimation algorithms converge most reliably when given an initial estimate in the neighborhood of the final answer. Therefore, we estimate each camera's parameters before starting bundle adjustment.

The camera extrinsics (rotation and translation) for each image are initialized by performing RANSAC [6] on two sets of points: the ground-point and image-point half of each ground reference pair. First, the inverse of the post-transformation matrix is applied to the image points, to ensure correct centering ($M_{post}^{-1} \cdot p_{image}$). Second, the pre-transformation matrix is applied to the ground points, to ensure correct scaling ($M_{pre} \cdot p_{ground}$). Finally, RANSAC is preformed between these two sets of points, resulting in a transformation matrix for each image, that is then used as the first estimation in the bundle adjustment algorithm.

The camera intriniscs (i.e. the focal length) are directly initialized from the EXIF metadata fields recorded in the image file by the actual camera. If EXIF information is unavailable, we assume the image was taken with a 40° angle of view.

4.3 Optimization Using Bundle Adjustment

Once camera models have been given initial estimates, they are refined using an iterative nonlinear optimization process called *bundle adjustment* [4]. Given a number of parameters to adjust (known in bundle-adjustment terminology as *active states*), and an error metric based on those parameters, a bundle adjuster iteratively makes small updates to the parameters until the error metric falls below a threshold.

As discussed in previous sections, MapStitcher has two types of constraints: constraints that pull images towards their correct geography and constraints that place images to minimize seams at their overlap points. These two constraints are represented by two different types of error metrics to the bundle adjuster.

The representation of the geographic constraints are straightforward. The camera intrinsics and extrinsics are represented as active states. The user-supplied ground reference pairs are used to compute the error metric. MapStitcher computes the projection of the ground point into the image plane using the hypothesized camera parameters. The distance from the projected ground point to the user-selected image point is the error.

Image-stitching constraints are somewhat more complex to model. In this case, the stitcher does not have a known ground point—only a set of image points that, according to the feature matcher (Section 4.1), depict the same ground feature. We add a new active state for each group of feature match points; it represents the hypothesized point on the ground depicted by those features. The initial estimate of this ground point is the centroid of the projection of all the feature match points onto the ground, given the estimates of those images' camera models. In each iteration of the bundle adjuster, the hypothetical ground point is projected back into the image plane of each image using the updated camera models. The error metric is the sum (over each image) of the distances in image space from these projections to the corresponding feature match points.

For further technical details, we refer the reader to [1], which describes the application of the bundle adjustment algorithm in a similar context.

4.4 Grounding Images Iteratively

If the user originally supplies ground reference pairs for *every* image in the mosaic, the procedure described above will work in a single step. Each camera's parameters could be initially estimated based on its image's ground reference pairs, and all parameters could be optimized in a single bundle-adjustment operation. However, such a system would be difficult to use: it can be time-consuming to find ground reference pairs manually and many mosaics contain dozens or hundreds of images. To minimize user effort, MapStitcher creates synthetic ground reference pairs using

adjacent overlapping images that already have camera model estimates.

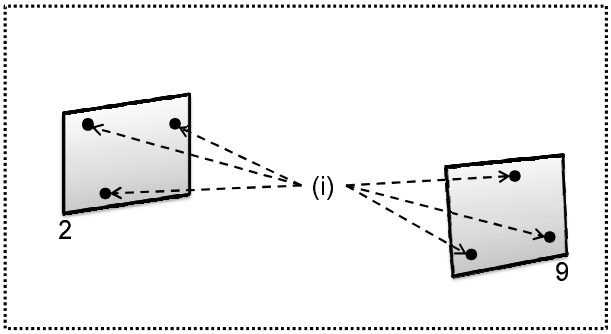
For example, imagine that our mosaic has images A and B. A has user-supplied ground reference pairs, but B does not. The feature matching algorithm tells us that pixel (A_x, A_y) in image A depicts the same feature as pixel (B_x, B_y) in image B. MapStitcher first "bootstraps" the mosaic using A's ground reference pairs to estimate A's camera parameters. It then uses those parameters to project (A_x, A_y) onto a ground point (A_{xg}, A_{yg}), and creates a synthetic ground reference pair for image B: (B_x, B_y) corresponds to (A_{xg}, A_{yg}). This technique can be used iteratively to propagate camera model estimates to an entire contiguous set of overlapping images. We call this successive propagation the *ripple algorithm*. Note that after each ripple, a *global* bundle adjustment is performed, as described in the previous section.

An example for a succession of ripple steps is shown in Figure 4.4. (For illustrative purposes, we depict only a small number of feature match points.) In the initial ripple, ground reference pairs (marked (i) on Figure 8(a)) are used to calculate the homographic transformations for image #2 and #9.

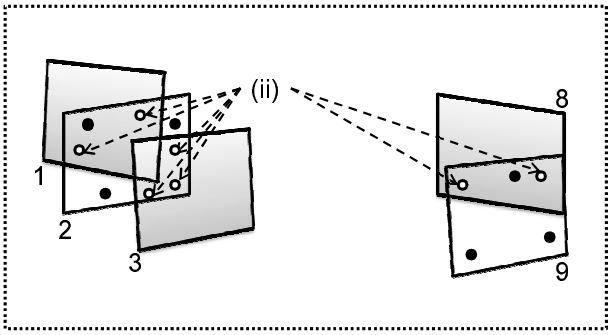
In the second ripple, feature match point pairs (marked (ii) on Figure 8(b)) are found that have one of their points on the known-model images: #2 and #9. These feature matches add images #1, #3 and #8 to the ripple. Note that although #1 and #3 overlap, the feature extraction and matching algorithm didn't find any feature match points between them in this case. The ground location of the feature match points are calculated using the homographic transformation obtained for image #2 and #9 in the initial ripple. After bundle adjustment, the ripple's three new images will also have their parameters for homographic transformation.

In the third ripple (Figure 8(c)), feature match points on images #3 and #8 add images #4, #5 and #7 to the ripple. Note that the two feature match points between the floating images #4 and #5, marked (iv), are feature match points without at least one image with a known position, and thus are not used in the RANSAC initialization of the third ripple.

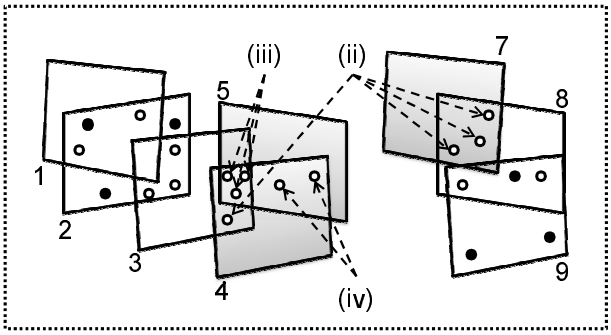
In the fourth ripple (Figure 8(d)), feature match points attach image #6 to both #5 and #7. Note that up until this ripple, there were two independent image groups: images #1–#5 were grounded based on ground reference pairs from image #2, and images #7–#9 were grounded based on ground reference pairs from image #9. The link provided by #6 joins these two groups, and the subsequent bundle adjustment jointly refines all 9 camera models *together* for the first time in search of a globally optimal solution. In addition, the feature match points marked (iv) between #4 and #5 can now be grounded (using the homographic projections from the previous ripple), which allows them to be used in the bundle adjustment. After the fourth ripple, all images in the cluster are grounded with homographic transformations, and the algorithm terminates.



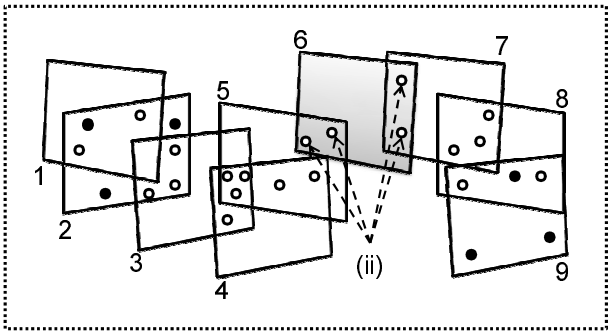
(a) Initial ripple; only ground reference pairs are used



(b) Second ripple; feature match points link some floating images to already grounded ones



(c) Third ripple; some feature match points link more than two images



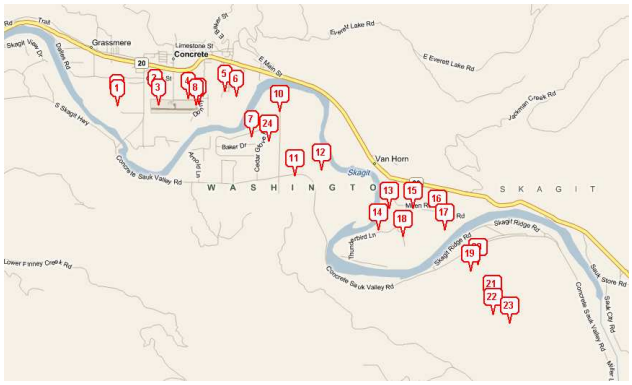
(d) Fourth and final ripple; a globally optimal solution is approached when independently estimated image groups join

Figure 8: A succession of ripples is used to estimate the position of all images, even though only a subset have user-specified ground reference pairs.

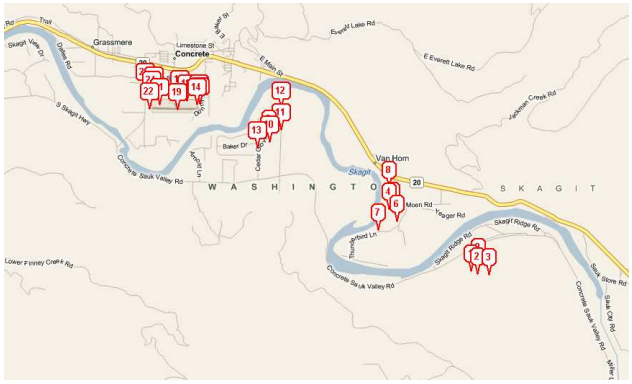
5. EVALUATION

MapStitcher is designed to produce a well-georeferenced aerial imagery layer stack with low human data-entry cost. To evaluate its design, we perform an experiment that compares a MapStitcher orthorectified image with two control methods, *no-stitch* and *stitch-first*. We measure each method on two criteria: cost of registration measured in number of manual ground reference pairs, and quality of registration measured in deviation of unreferenced points from ground truth. In these experiments, "ground truth" is defined by the lower-resolution Virtual Earth aerial photography of the subject region, and is affected by distortions in the Virtual Earth orthorectification pipeline.

(a) Ground reference points for *no-stitch* method.



(b) Ground reference points for *stitch-first* method.



(c) Ground reference points for MapStitcher method.

Figure 9: Locations of ground reference points.

5.1 Experiment Description

For this experiment, we use as input 60 source images we captured of the Skagit River Valley in the vicinity of the town of Concrete, WA. We used each of the three techniques to combine all source images to produce a single orthorectified, tiled composite image of 208 megapixels.

5.2 Measuring Cost

For the *no-stitch* method, we registered 257 points (mean 4.3 points per image; Figure 9(a)).

For the *stitch-first* method, we stitched the images with the fully automatic photo stitcher described in [2]. We georegistered the resulting composite image with 25 manually-entered ground reference pairs (Figure 9(b)).

Figure 10: The only images with manually entered ground reference pairs in our MapStitcher example.

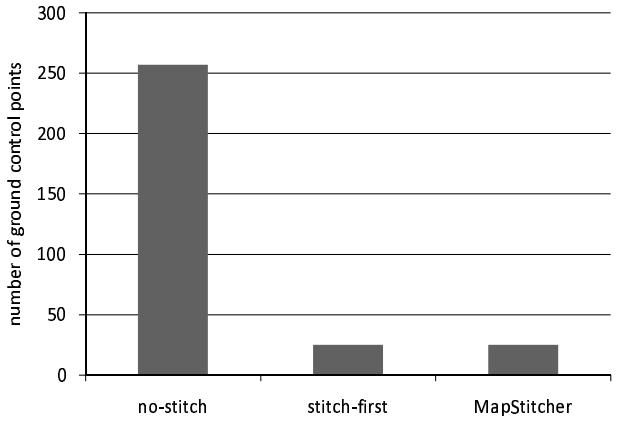


Figure 11: Number of manually entered ground reference pairs.

For the MapStitcher method, we registered 25 points spread out on 5 images (mean 0.4 points per image over the whole set; Figure 9(c)). We show the five images with manually entered ground reference pairs after transformation, on Figure 10, while Figure 3 shows all 60 images georegistered based on these five images.

Figure 11 shows the number of manual ground reference pairs for the three methods.

5.3 Measuring Quality

We manually selected 12 recognizable points in the scene, each from separate source images, none of which were used as manually-entered reference points in any of the methods. We measured the "ground truth" position of each point in the low-resolution Virtual Earth image. For each method, we computed the mean distance between where the method geolocates each point versus the point's ground truth position.

Figure 12 shows the mean and standard deviation of the registration errors for the three methods. The *no-stitch* method produces the best quality orthorectification, with $25.1\ m$ mean error and $15.8\ m$ standard deviation, but using 10.3 times as many manual points as the other methods. The reference *stitch-first* method results in a mean error of $354.1\ m$ (with a large $167.9\ m$ standard deviation), showing that it is difficult to recover geography as a discrete step if a mosaic is created using seamless-boundary

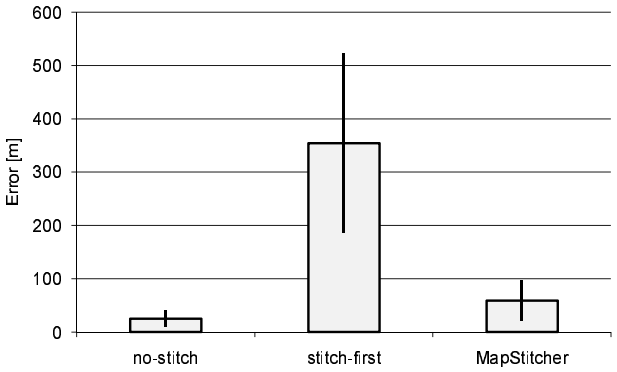


Figure 12: Mean and standard deviation of registration error.

constraints alone. Our method, which jointly optimizes image-to-ground and image-to-image alignment, results in a mean error that is 234% ($58.83\ m$) of the no-stitch method (with a standard deviation of $37.9\ m$), while needing only 9.7% of the manually entered ground reference pairs of the latter method. The increased error may be due to placing too much relative weight on image-to-image alignment—that is, in some cases, we may be sacrificing absolute positional accuracy for the sake of output that looks better.

6. FUTURE WORK

While our current system produces composite imagery whose georeferencing quality approaches that of the manual no-stitch method, it suffers from similar problems as that method: image boundaries remain clearly visible at some image boundaries. stitching techniques employ graphcut algorithms to reduce visible seams in the final composite [10], and gain compensation and multi-band blending is used to correct for unmodelled camera effects (e.g. vignetting) [2]. Our application would also benefit from these techniques. MapStitcher currently has no *a priori* information about the relative positions of any images, and thus must attempt to find feature matches between all image pairs. Adding a constraint that indicates potential image overlaps will simplify the problem of finding feature matches, as the number of candidate images to be considered will be reduced from $O(n^2)$ to a constant-sized neighborhood. This will significantly improve processing speed and reduce feature match outliers, and can be achieved using a low-cost (consumer-grade) GPS that is only loosely coupled to the image acquisition process.

7. CONCLUSION

MapStitcher produces orthorectified aerial imagery mosaics from images with poorly constrained geometry and only minimal manual labeling. The result is a system with low capital cost that produces high-quality image mosaics. We anticipate that access to such low-cost imaging will lead to a much wider grass-roots effort to produce aerial photography. We hope to facilitate community-supported efforts aimed, for example, at better coverage of non-urban areas, timely coverage of special events or natural disasters, or more frequent converage of fast-changing areas. Ultimately, if aerial imaging becomes as cheap and easy to produce as a blog, we may see aerial imagery with the same rich, decentralized diversity as the blogosphere.

8. REFERENCES

[1] M. Brown and D. G. Lowe. Unsupervised 3D object recognition and reconstruction in unordered datasets. In *3DIM*, pages 56–63. IEEE Computer Society, 2005.

[2] M. Brown and D. G. Lowe. Automatic panoramic image stitching using invariant features. *International Journal of Computer Vision*, 74(1):59–73, Aug. 2007.

[3] M. Brown, R. Szeliski, and S. Winder. Multi-image matching using multi-scale oriented patches. In *CVPR*, pages 510–517. IEEE Computer Society, 2005.

[4] B. A. DeWitt and P. R. Wolf. *Elements of Photogrammetry (with Applications in GIS)*. McGraw-Hill Higher Education, 2000.

[5] J. Elson, J. Howell, and J. R. Douceur. Mapcruncher: integrating the world's geographic information. *Operating Systems Review*, 41(2):50–59, 2007.

[6] M. A. Fischler and R. C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395, 1981.

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

[8] F. A. V. D. Heuvel. Exterior orientation using coplanar parallel lines. *Proceedings of the 10th Scandinavian Conference on Image Analysis*, pages 71–78, 1997.

[9] F. A. V. D. Heuvel. Estimation of interior orientation parameters from constraints on line measurements in a single image. *International Archives of Photogrammetry and Remote Sensing*, 32:81–88, 1999.

[10] V. Kwatra, A. Schödl, I. A. Essa, G. Turk, and A. F. Bobick. Graphcut textures: image and video synthesis using graph cuts. *ACM Trans. Graph*, 22(3):277–286, 2003.

[11] S. Mahamud, M. Hebert, Y. Omori, and J. Ponce. Provably-convergent iterative methods for projective structure from motion. In *CVPR*, pages 1018–1025. IEEE Computer Society, 2001.

[12] P. F. McLauchlan and A. Jaenicke. Image mosaicing using sequential bundle adjustment. *Image Vision Comput*, 20(9-10):751–759, 2002.

[13] L. Quan and Z.-D. Lan. Linear N-point camera pose determination. *IEEE Trans. Pattern Anal. Mach. Intell*, 21(8):774–780, 1999.

[14] B. Vandeportaele, C. Dehais, M. Cattoen, and P. Marthon. ORIENT-CAM, A camera that knows its orientation and some applications. In J. F. M. Trinidad, J. A. Carrasco-Ochoa, and J. Kittler, editors, *CIAPR*, volume 4225 of *Lecture Notes in Computer Science*, pages 267–276. Springer, 2006.