Structure and motion in urban environments using upright panoramas

Jonathan Ventura · **Tobias Höllerer**

Received: date / Accepted: date

Abstract Image-based modeling of urban environments is a key component of enabling outdoor, vision-based augmented reality applications. The images used for modeling may come from offline efforts, or online user contributions. Panoramas have been used extensively in mapping cities, and can be captured quickly by an end-user with a mobile phone. In this paper, we describe and evaluate a reconstruction pipeline for upright panoramas taken in an urban environment. We first describe how panoramas can be aligned to a common vertical orientation using vertical vanishing point detection, which we show to be robust for a range of inputs. The orientation sensors in modern cameras can also be used to correct the vertical orientation. Secondly, we introduce a pose estimation algorithm which uses knowledge of a common vertical orientation as a simplifying constraint. This procedure is shown to reduce pose estimation error in comparison to the state of the art. Finally, we evaluate our reconstruction pipeline with several real-world examples.

Keywords structure and motion · urban environments · panoramas

1 Introduction

In this paper we consider the problem of image-based modeling with the constraint that all cameras have zero pitch and roll. We call this the 'upright constraint.' This constraint applies in two interesting scenarios for the creation of urban models for virtual environments. Firstly, panoramic captures of urban environments are typically made using a camera mounted above a moving platform, which ensures that the images are roughly upright most of the time. Secondly, modern smartphones include orientation sensors for determining the pitch and roll of the camera, which can then be removed from the image. Images captured by an outdoor augmented reality user, for example, can be rotated to upright using these sensors. We show in this paper that in either case, the upright constraint can be used to improve the robustness of structure from motion with panoramic imagery.

J. Ventura and T. Höllerer
Department of Computer Science
University of California
Santa Barbara, CA 93106-5110
Tel.: +1-805-893-2614
Fax: +1-805-893-8553
E-mail: jventura@cs.ucsb.edu

We show in Section 4 how images that are approximately upright can be aligned to a common vertical orientation using vanishing point detection. We provide an evaluation of vanishing point detection on spherical panoramas, which have distorted line segments. Our evaluation shows that our method is typically accurate to one degree in spite of this distortion.

We also developed a novel algorithm for absolute pose estimation with the upright constraint (Section 5). In comparison to the state of the art, our algorithm produces less estimation error in the face of image noise, and is simpler to compute. This makes our method an attractive option for model-based tracking in outdoor augmented reality.

The image alignment and structure from motion procedures we describe can be combined into a reconstruction pipeline for urban environment modeling from panoramas (Section 6). We prepared several test sequences to evaluate this pipeline, both using a camera on a tripod and using a handheld smartphone. We show that our methods lead to reconstructions with little drift for small datasets, even without bundle adjustment (Section 7).

Algorithm 1 gives an overview description of the main steps in our approach to reconstructing urban environments from panoramas.

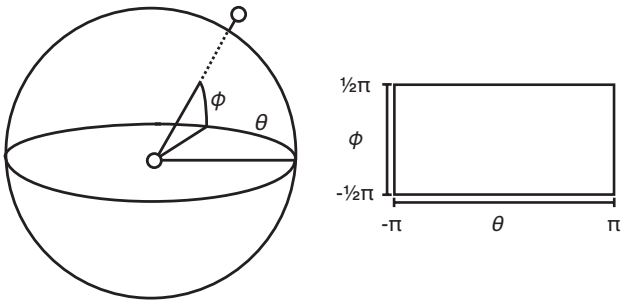
Algorithm 1 Structure and motion from panoramas

Rectify vertical orientation of each panorama (Section 4)
Extract and match feature descriptors between panoramas
Determine relative pose of the first two panoramas (Section 5.1)
Determine absolute pose of the remaining panoramas (Section 5.2)

2 Related work

Vanishing point detection is a well studied problem, and several robust methods have been proposed. Rother's work provides a good overview of approaches to the problem, which are generally based on either the Hough transform, RANSAC, or exhaustive search [16]. In the context of image-based modeling, Antone and Teller used vanishing point detection on hemi-spherical images and matched between capture points to completely determine relative orientations [1]. In our pipeline, we independently correct the orientation of each panorama to a common vertical, and then use structure from motion methods to resolve the unknown yaw between cameras. This has the advantage that images can be pre-processed in parallel to bring them to a common orientation. The vertical and horizontal vanishing point can also be used to determine a homography which rectifies the image of a building facade. This simplifies building recognition and the camera pose estimation [15,2]. However, in the case of significant occlusions, it may not be possible to reliably extract both vanishing points. The procedure described by Gallagher also used vertical vanishing point detection to correct for image roll, but did not compensate for both pitch and roll as is done here [4].

Much previous work has focused on urban modeling with images taken from a moving platform [14,12,19] or from community photo collections [17]. Here, we examine how the reconstruction problem is made simpler and more robust by using the upright constraint on camera poses. Previously, knowledge of two orientation angles has been used to constrain relative pose estimation [3]. This special case leads to a simplified essential matrix which can be estimated using three points correspondences at minimum, as opposed to the standard five point algorithm [13]. Constrained absolute pose estimation has also been studied previously, leading to a minimal solution using two point correspondences [9]. In this work we demonstrate a novel linear estimation algorithm for constrained absolute pose, which permits an overdetermined solution. The overdetermined solution is important when dealing with noisy measurements. Our evaluations show our algorithm to be more robust to image noise than current methods.



(a) (b)

Fig. 1: (a) Illustration of spherical projection. (b) Mapping from angles to the 2D image.

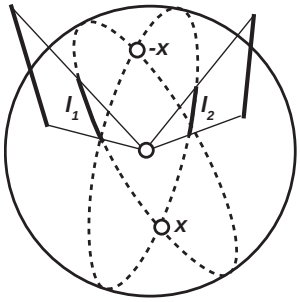


Fig. 2: Lines in 3D space project to arcs on the sphere. We find their intersection point by extending the arcs to great circles.

3 Panoramic projection

Standard computer vision makes use of perspective projection which maps a point in 3D space $\mathbf{X} = [X\ Y\ Z\ 1]^T$ to image coordinates (x,y) by projecting points onto the plane $Z = 1$:

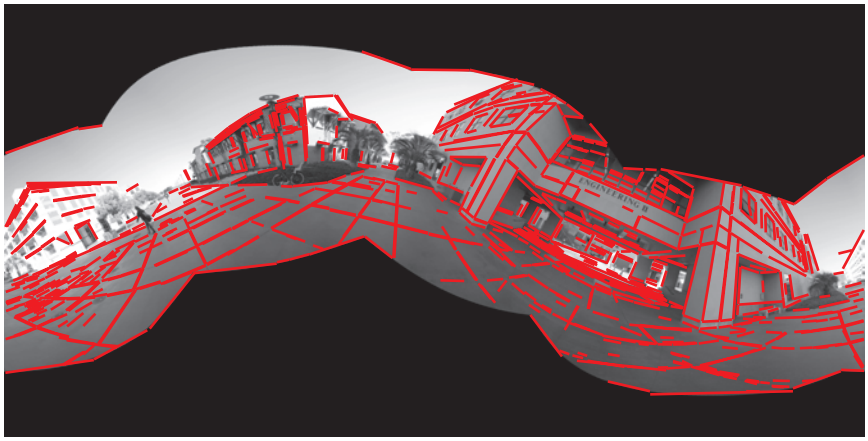
$$x = \frac{X}{Z} \tag{1}$$

$$y = \frac{Y}{Z} \tag{2}$$

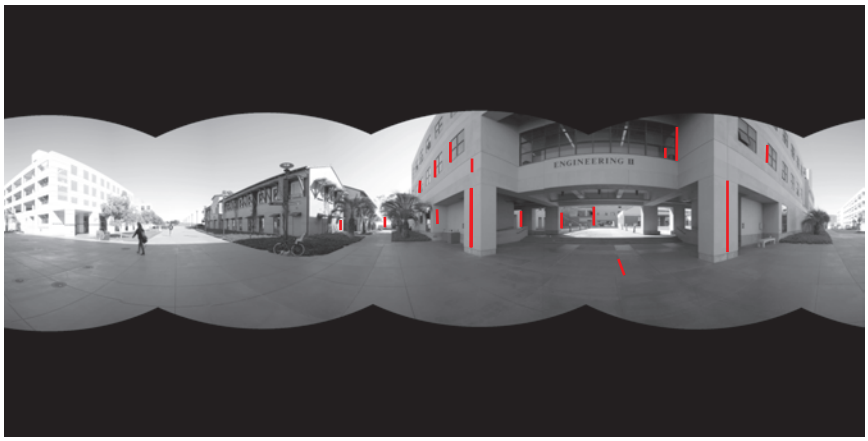
Perspective projection is not suitable for panoramic images, because of the singularity at $Z = 0$, and the fact that $\mathbf{X}$ and $-\mathbf{X}$ map to the same point. In this paper we consider the equirectangular or spherical projection, where we project points onto the unit sphere (Figure 1). The image is parameterized by the two angles θ and ϕ describing the ray from the origin to the point:

$$\theta = \tan^{-1}\left(\frac{X}{Z}\right) \tag{3}$$

$$\phi = \sin^{-1}\left(\frac{Y}{||\mathbf{X}||}\right) \tag{4}$$



(a)



(b)

Fig. 3: (a) A panorama which has been synthetically rotated away from upright, with detected line segments in red. (b) The panorama after rotation correction, with inliers to the vertical vanishing point detection process.

Alternatively the cylindrical projection can be used, which limits the vertical field of view but lessens compression of the image. Given coordinates (θ, ϕ) in the panoramic image, we can invert the projection to acquire the point where the sampled ray intersects the unit sphere.

We note that line segments in 3D space project to arcs on the unit sphere (Figure 2). Given two endpoints $\mathbf{x}_1$ and $\mathbf{x}_2$ in homogeneous coordinates, the infinite line connecting them can be found using the cross product.

$$\mathbf{l} = \mathbf{x}_1 \times \mathbf{x}_2 \tag{5}$$

This line represents a great circle on the sphere. The intersection point of two lines can be found also by using the cross product.

$$\mathbf{x} = \mathbf{l}_1 \times \mathbf{l}_2 \tag{6}$$

On the sphere, the second intersection point lies at $-\mathbf{x}$. For parallel lines in 3D space, the intersection point maps to the vanishing point in the image (which may lie at infinity).

The angular distance on the sphere between a point and a line is given by:

$$d(\mathbf{l}, \mathbf{x}) = \sin^{-1}(\mathbf{l} \cdot \mathbf{x}) \tag{7}$$

assuming that $\mathbf{l}$ and $\mathbf{x}$ have unit length.

4 Rectification

We observe that with either perspective, spherical or cylindrical projection, vertically straight edges in 3D space are projected to vertical lines in image space. In other words, line segments where X and Z are constant project to imaged

lines where x or θ is constant. Thus we can bring any suitable image to a common upright orientation by rotating the projected lines to be vertically straight. Note that this is a 3D rotation which corrects for both pitch and roll. This rotation is equivalent to aligning the vertical axis of the camera with the vertical axis of the scene.

Using the spherical or cylindrical projection, non-vertical lines will appear curved. However, the distortion increases as the camera is rotated away from vertical. We make the assumption that the camera's orientation is upright enough that we can still detect straight lines to be rectified.

We first detect lines using the method of Kosecka and Zhang [8] which extracts Canny edges thresholded by length. Lines which are too short are discarded as noise. To obtain candidates for vertical segments, we filter the lines by their angle in the image, keeping only those within 45 degrees of vertical. We assume that the line segments in the image represent projections of straight lines in 3D space.

We use a RANSAC procedure on line pairs to detect the vertical vanishing point in the image. Given a pair of lines $\mathbf{l}_1$ and $\mathbf{l}_2$, we determine their intersection $\mathbf{x} = \mathbf{l}_1 \times \mathbf{l}_2$. Other lines are classified as inliers or outliers by thresholding $|\mathbf{l} \cdot \mathbf{x}| \leq \tau$ (where $\mathbf{l}$ and $\mathbf{x}$ are normalized). In our experiments we used a threshold of $\tau = \sin(2^\circ)$ corresponding to an angular error of two degrees (see Equation 7). Figure 3 gives an example of the process.

Once the vertical vanishing point has been determined, we calculate the rotation which brings the vertical vanishing point to the top of the sphere (see Figure 2). The line detection and RANSAC procedure can optionally be iterated for more accuracy. Since the distortion of vertical lines is reduced after each iteration, by re-running the detection step we can extract more and longer vertical lines to be used in vanishing point detection.

We evaluated this method with a dataset of upright panoramas, which were taken using a digital SLR camera on a level tripod. We then generated synthetically rotated panoramas by applying a random rotation about the Y-axis followed by a rotation about a vector in the X-Z plane and ran our straightening algorithm to correct them (see Figure 3). Using four different panoramas, we tested a rotation of 10, 20 and 30 degrees with 25 trials each. Figure 4 shows the results of our experiment. On average, the algorithm corrects the panorama to within one degree of the original pose in one iteration.

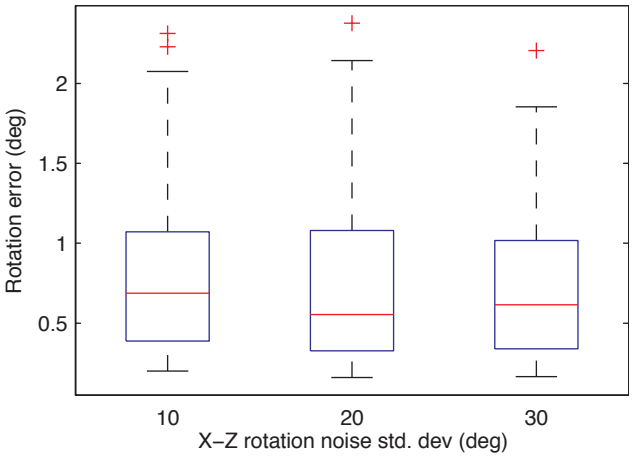


Fig. 4: Remaining error in vertical orientation after the vanishing point detection and alignment process. We tested the accuracy of vanishing point alignment by synthetically introducing rotation to an upright panorama and then using our rectification technique to return the panorama to upright. The box plots shows the error in the rectified result over 100 trials for each amount of off-vertical rotation synthetically added.

5 Upright Pose Estimation

The previous section presented a method to bring panoramas to a common vertical orientation based on vertical vanishing point detection. Alternatively, a measurement of the vertical orientation could be provided by sensors such as an accelerometer. In either case, we can reduce the rotation between panoramas to a single degree of freedom, the rotation about the vertical axis (the Y-axis). In this section we discuss methods to determine the rotation and translation between upright panoramas based on the projections of mutually observed points in 3D space (see Figure 5).

We examine the pose estimation problem under the constraint that there is only rotation about the Y-axis between cameras. Consider two cameras, $P = [I \mid 0]$ and $P' = [R \mid \mathbf{t}]$, where world points $\mathbf{X} = [X\ Y\ Z\ 1]^T$ are projected into the two images by $\mathbf{x} = P\mathbf{X}$ and $\mathbf{x}' = P'\mathbf{X}$. According to our upright motion assumption, the rotation can be expressed in terms of a single parameter θ:

$$R(\theta) = \begin{bmatrix} \cos(\theta) & 0 & \sin(\theta) \\ 0 & 1 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) \end{bmatrix} \tag{8}$$

In this section we review previous approaches to relative and absolute pose estimation using the upright constraint. In section 5.2 we present our novel absolute pose estimation method.

5.1 Relative pose

Fraundorfer, Tankskanen and Pollefeys have analyzed this special case of essential matrix estimation and tested the linear five point estimation algorithm described below, as well as polynomial four point and three point algorithms which

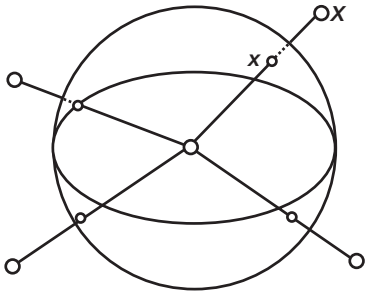


Fig. 5: Using correspondences between known 3D points and their projections on the sphere, the pose of the panorama can be determined.

incorporate nonlinear constraints on the essential matrix [3]. These are analogous to the familiar eight point, seven point, and five point algorithms for general essential matrix estimation. Their evaluation shows that the linear five point and the three point algorithm are both more robust to image measurement noise than the standard five point algorithm in the case of motion with only one unknown orientation angle.

We review the linear relative pose algorithm here. For relative pose estimation, epipolar constraints on point correspondences can be used to estimate the essential matrix E. The essential matrix represents the epipolar constraints $\mathbf{x}'^T E\mathbf{x} = 0$. Using the upright assumption, $E = R[\mathbf{t}]_\times$ has the form:

$$E = \begin{bmatrix} -t_y\sin(\theta) & t_x\sin(\theta) - t_z\cos(\theta) & t_y\cos(\theta) \\ t_z & 0 & -t_x \\ -t_y\cos(\theta) & t_x\cos(\theta) + t_z\sin(\theta) & -t_y\sin(\theta) \end{bmatrix} \tag{9}$$

From this we can derive the following relations on the essential matrix:

$$E_{22} = 0 \tag{10}$$
$$E_{11} = E_{33} \tag{11}$$
$$E_{13} = -E_{31} \tag{12}$$

This leaves five unknowns in the essential matrix. Re-arranging the epipolar constraints $\mathbf{x}'^T E\mathbf{x} = 0$ results in a linear system of rank four (since the essential matrix is only determined up to scale). This system can be solved using singular value decomposition (SVD), and a proper essential matrix can be extracted by constraining the singular values of E such that $s_1 = s_2 = 1$ and s_3=0.

5.2 Absolute pose

Absolute pose estimation can also be constrained using the upright assumption. Kukelova, Bujnak and Pajdla presented a second-degree polynomial solution to the problem in the minimal case of two correspondences [9]. Here, we present a linear solution requiring three correspondences. Our algorithm, although less exact in the minimal case, permits a solution to an overdetermined system. Solving the overdetermined system may reduce error in the estimate using multiple noisy correspondences.

We introduce here an accurate linear solution to the absolute pose problem using knowledge of two orientation angles. Given image observations $\mathbf{x}$ of world points $\mathbf{X}$ we wish to solve for the camera's extrinsic parameters $P = [R \mid \mathbf{t}]$ subject to the projection equations. This is also known as the Perspective-n-Point or PnP problem. The projective relations are as follows:

$$\frac{x}{w} = \frac{R_1\mathbf{X} + t_x}{R_3\mathbf{X} + t_z} \tag{13}$$
$$\frac{y}{w} = \frac{R_2\mathbf{X} + t_y}{R_3\mathbf{X} + t_z} \tag{14}$$

Using the single-parameter form of rotation given in Equation 8, the camera matrix has the form

$$P = \begin{bmatrix} \cos(\theta) & 0 & \sin(\theta) & t_x \\ 0 & 1 & 0 & t_y \\ -\sin(\theta) & 0 & \cos(\theta) & t_z \end{bmatrix}. \tag{15}$$

From this we identify seven linear relations, leaving five unknowns in P.

$$P_{12} = P_{21} = P_{23} = P_{32} = 0 \tag{16}$$
$$P_{22} = 1 \tag{17}$$
$$P_{11} = P_{33} \tag{18}$$
$$P_{13} = -P_{31} \tag{19}$$

By re-arranging the projection equations, each 2D-3D correspondence gives two linear equations:

$$\begin{bmatrix} -wX + Zx & Zy \\ -wZ - Xx & -Xy \\ -wW & 0 \\ 0 & -Wy \\ 0 & -wW \\ Wx & Wy \end{bmatrix}^T \begin{bmatrix} P_{11} \\ P_{13} \\ P_{14} \\ P_{22} \\ P_{24} \\ P_{34} \end{bmatrix} = 0. \tag{20}$$

Note that we use the homogeneous coordinate w in the equations, rather than assuming $w = 1$. This is because in the case of panoramas with complete horizontal field of view, it is possible to have image coordinates where w is near zero.

We use P_{22} as the free scale parameter, so the system has six unknowns, but has rank five. This means we need three correspondences for absolute pose estimation. (Actually only five equations are needed in the minimal case, so the y-coordinate can be disregarded from one correspondence). The standard direct linear transform (DLT) algorithm for general absolute pose estimation requires six correspondences [6].

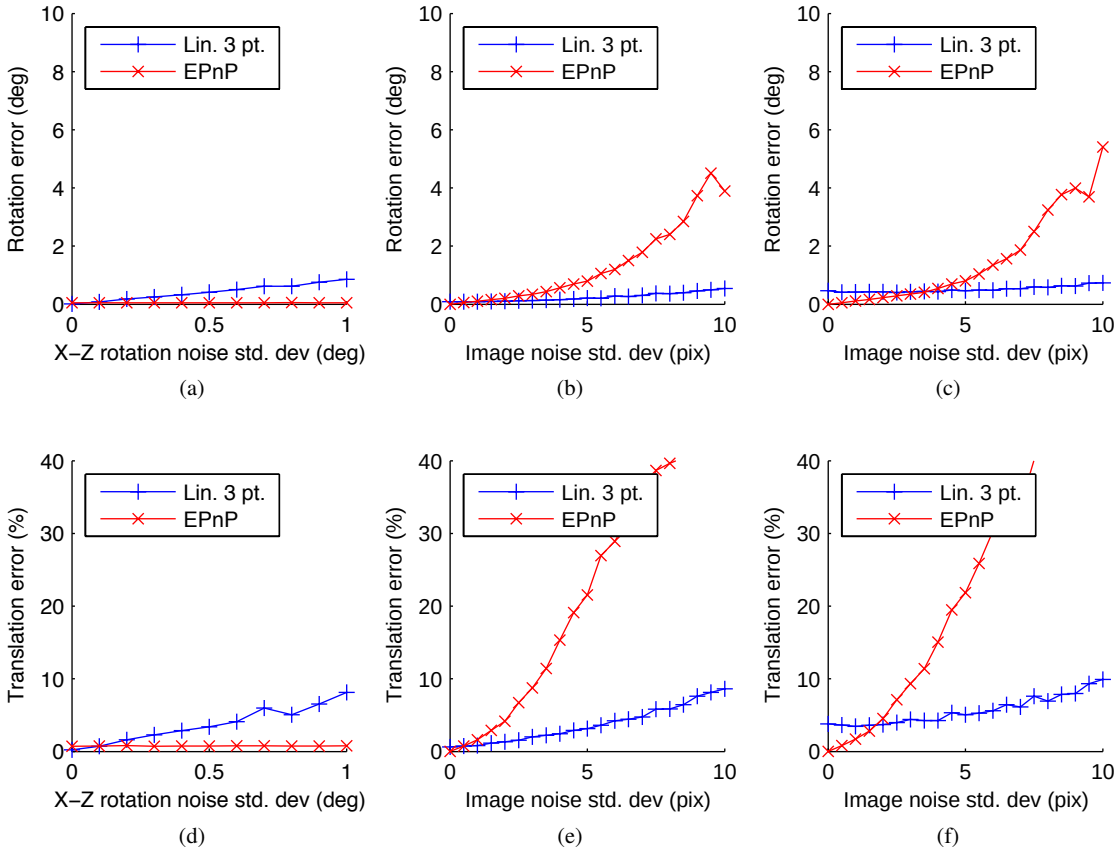


Fig. 6: *Left column*: Rotation and translation error in absolute pose estimation as the camera is tilted away from vertical, with image observation noise of $\sigma_{\text{im}} = 0.5$ pixels. *Middle and right column*: Rotation and translation error in absolute pose estimation as image observation noise is increased, with a random rotation away from vertical of $\sigma_{XZ} = 0.1$ degrees (middle column) and $\sigma_{XZ} = 0.5$ degrees (right column). Each data point represents the average of three hundred trials. One hundred 2D-3D correspondences were used for each trial.

This system of equations can be solved using the singular value decomposition (SVD). The resulting solution is normalized by dividing by P_{22}. In the case of noisy measurements, this solution may not produce a true rotation matrix which satisfies the quadratic constraint:

$$P_{11}^2 + P_{13}^2 = 1. \tag{21}$$

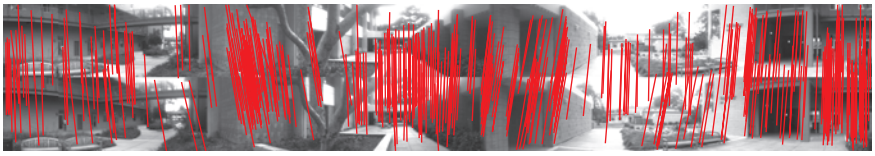
The two solutions satisfying this constraint are found by normalizing P_{11} and P_{13} by $\pm\sqrt{P_{11}^2 + P_{13}^2}$.

The general DLT algorithm is traditionally not used for absolute pose estimation, because it is not very robust to noise. Instead, most methods estimate the distance from the camera to the 3D points in the camera's reference frame [5]. Then, a standard algorithm is used to determine the rotation and translation between the camera's coordinate system and the world coordinate system [7]. However, because of the upright constraint, we have less unknowns in our system, and thus can be more robust to noise, even when using the simple linear formulation.

5.2.1 Synthetic evaluation

We evaluated our algorithm in comparison to a recent non-iterative method called EP*n*P which is accurate and fast for general absolute pose estimation [10]. We generated synthetic test cases to evaluate the two methods. Our experimental setup replicated that of Lepetit et al. [10]. We sampled 3D points with $X \in [-2,2]$, $Y \in [-2,2]$ and $Z \in [4,8]$. The image observations were generated using a focal length of $f = 800$ and the principal point at (320,240). Gaussian noise of standard deviation σ_{im} pixels was then added to the image coordinates.

The camera matrix was chosen as follows. Each component of the translation vector was uniformly sampled from $[-1,1]$. The rotation matrix models the the case of upright motion contaminated by some small rotation noise which



(a)

(b)

Fig. 7: (a) Two images from the Scene 1 sequence. Red lines indicate inlier correspondences from the absolute pose estimation procedure. (b) Images and inlier correspondences from the handheld sequence.

tilts the up vector away from vertical. The rotation was computed as $R = R_{XZ}R_Y$ where R_Y is a random rotation about the Y axis with $\theta_Y \in [-\frac{\pi}{4}, \frac{\pi}{4}]$ and R_{XZ} is a random rotation about the vector $[1\ 0\ 1]^T$ with $\theta_{XZ} \sim N(0, \sigma_{XZ})$.

We ran two tests to evaluate the behavior of our linear algorithm and the EPnP algorithm. For each test we ran 300 trials for each parameter setting. The rotation error was calculated as the angular magnitude of the minimal rotation between the estimated result and the correct result. The translation error was calculated as the percentage error between the estimated result and the correct result, given by $||\mathbf{t} - \mathbf{t}_{ans}||/||\mathbf{t}_{ans}||$.

For our first test, we evaluated robustness to non-upright movement by increasing the off-vertical rotational noise σ_{XZ} from 0 to 1 degree. For these tests we used an image observation error of $\sigma_{\text{im}} = 0.5$ pixels. Figure 6 plots the estimation error of the two algorithms. Clearly our algorithm performs best when the amount of rotation away from vertical is minimal.

For our second test, we evaluated robustness to increasing image measurement noise assuming that the camera pose is roughly upright. We used a rotational noise of $\sigma_{XZ} = 0.1$ and $\sigma_{XZ} = 0.5$ degrees and increased the image observation noise σ_{im} from 0 to 10 pixels. Figure 6 plots the results. Our linear algorithm is shown to be more robust and stable than EPnP across a range of image noise. The average error is roughly constant for our algorithm, while the error of EPnP grows rapidly as image noise increases. This shows that the upright assumption can increase robustness to image noise when estimating absolute pose, even when the assumption is not perfectly true.

6 Reconstruction pipeline

We assume here that we have a sequence of panoramas such that neighboring images have significant visual overlap.

6.1 Image pre-processing

First, we ensure that all panoramas are upright by using our vanishing point alignment technique described in Section 4. Even when we have an orientation sensor, this step can compensate for noise in the sensor by verifying that vertical lines are straight.

6.2 Pose estimation

Once we have straightened all panoramas in the sequence, we perform point-based structure and motion analysis to determine camera poses and a point cloud. We detect points using the SIFT detector [11]. Because we have removed all in-plane rotation from the images, we use the upright SIFT descriptor for feature matching. The upright descriptor has been shown to be more discriminative when matching images without in-plane rotation [2]. We extract descriptors directly from the spherical panoramas. Although the panorama has some non-linear distortion due to the spherical projection, this distortion does not affect feature matching when the baseline between panoramas is small as in our datasets.

We perform relative pose estimation between the first and second panoramas in the sequence using the linear five-

point essential matrix estimation algorithm [3]. The essential matrix can be decomposed into four possible poses, giving two rotations and translation vectors. The ambiguity in rotation is resolved by checking that the up vector stays upright. The ambiguity in translation is traditionally resolved by checking the cheirality of each solution, i.e. the number of triangulated points which are in front of the both cameras [6]. However, in the case of panoramas, points can equally be observed with a positive or negative Z value. Instead, we use a more general form of cheirality by checking that the observed ray is in the same direction as the triangulated point [20]. More formally, given an observed ray $\mathbf{x}$ and the ray to the triangulated point $\hat{\mathbf{x}}$, we desire that $\mathbf{x} \cdot \hat{\mathbf{x}} > 0$. The solution with the higher number of points in agreement is selected.

After relative pose estimation, we triangulate 3D points using the first image pair and the direct linear transform (DLT) triangulation method. Then, for each remaining panorama we use our absolute pose estimation algorithm described in Section 5.2 to determine the pose from observed points, and then re-triangulate points incorporating the new image. In previous work, the relative pose between each image was estimated, and then the scale for each pair was determined by aligning triangulated points [18]. However, we found that for small datasets, the upright assumption constrains the estimation enough that we can directly solve for the pose and scale without significant drift.

7 Evaluation

We evaluated our reconstruction pipeline on real image sequences to test the usefulness of the upright constraint. Here we present results on panoramas captured using several different methods. We tested the reconstruction pipeline both with and without the vanishing point alignment step, to evaluate its benefit when the vertical orientation is known to varying degrees. An top-down view of the reconstructions is presented in Figure 9, with recovered camera poses (in red) and 3D triangulated points (in blue).

7.1 Tripod panorama capture

Our first evaluation uses sequences of panoramas taken precisely using a tripod and a pan-tilt-unit (PTU). The panoramas were captured by mounting a Point Grey Dragonfly2 camera to the PTU and panning in one degree increments. The camera had about a forty-five degree field of view and was calibrated beforehand. The tripod was placed on the ground and checked with a bubble level. The panoramas were taken in a line by moving the tripod along a measuring tape affixed to the ground, so that each image has equal spacing (about .5 meters) from the previous. We used panoramic

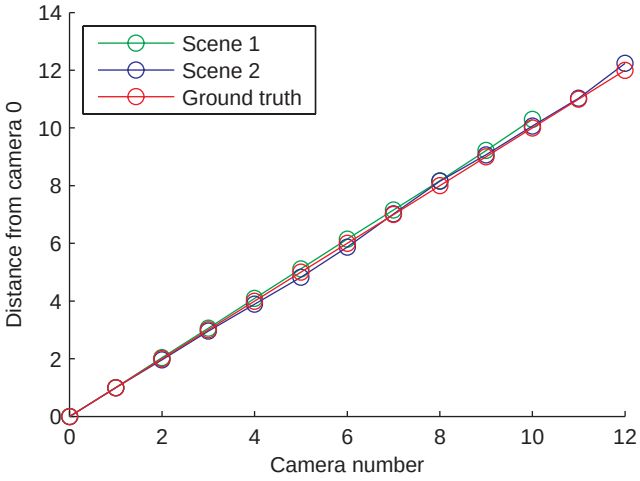


Fig. 8: Plot of the distance to the first camera for both scenes. Ground truth was established by physically measuring the distance on the ground between capture points.

sequences captured in two building courtyards on our campus[1].

We ran our image-space leveling algorithm on each spherical panorama, and then ran the two sequences through the structure-and-motion pipeline described in Section 6. Figure 7 shows inlier correspondences to the absolute pose estimation process across two images, and Figure 9 shows the reconstructions. Note that the camera moves in a straight path as expected.

To check the validity of the structure and motion output, and to determine the drift of the solution, we plotted the distance of each camera from the first in each sequence. For each sequence the translation in the initial pair was assumed to have unit length. This means that the distance from the first camera should increase linearly with the camera number, since the cameras were evenly spaced. Figure 8 plots the distances for the two sequences compared to ground truth. The reconstruction in both cases does not exhibit significant drift.

7.2 Handheld panoramas from a smartphone

We also tested the system using panoramas created using a handheld smartphone. We used an iPhone 4 to create panoramas using the 360 Panorama software[2]. This software uses the accelerometer and gyroscopes in the device to track rotation and establish orientation with respect to the ground. It also visually detects loop closures to reduce drift. The loop closure process, however, can adversely affect the vertical orientation estimate.

[1] Panoramas available from: `http://tracking.mat.ucsb.edu/`
[2] From Occipital Inc.: `http://www.occipital.com/360/`

To create each panorama, we held the phone with two hands roughly upright, and slowly spun once in a full circle with the phone held as tightly as possible to the axis of rotation. Using this method, we captured two panoramic sequences from outdoor scenes in a neighborhood with houses and trees, moving forward a few steps (about 1.5 meters) between captures. Figure 7 shows the last two images from one sequence and the inlier correspondences found by RANSAC absolute pose estimation. Figure 9 shows the reconstruction results for another sequence of six handheld panoramas. Because of the error in vertical orientation caused by the loop closure problem, we found that the reconstruction benefited from use of the vanishing point alignment technique, despite the use of vertical orientation sensors during capture.

7.3 Google Street View

We also tested our algorithm on a subset of panoramas from the Google Street View Research Dataset. These panoramas are captured from a moving vehicle using a panoramic camera which produces a spherical panorama with a very large vertical field of view. We used a set of thirty panoramas taken as the vehicle moved in a straight line down a city street, with about 1.5 meter spacing between panoramas. Although the camera is roughly upright, there can be variation in the vertical direction due to the orientation of the street and the movement of the vehicle.

In this case, we notice significant improvement to the reconstruction when using vertical vanishing point detection to align the panoramas beforehand (see the third row of Figure 9). The vertical vanishing point alignment decreased noise in point triangulation and led to a straighter camera pose path. In this reconstruction we noticed some drift in the reconstruction, which suggests that the system would benefit from the use of bundle adjustment and loop closure when a larger number of panoramas are reconstructed.

8 Conclusion

We have presented here methods for rotationally aligning images based on the vertical vanishing point, and reconstructing a sequence of images using the constraint that they have only rotation about the vertical axis between them. Using synthetic and real-world image sequences, we showed how our methods are robust to a variety of noise and improve upon the state of the art.

We believe these methods would be very useful in an outdoor augmented reality system using a smartphone. For example, in an unknown environment, a single person could capture several panoramas which are combined to produce a complete visual model of the environment. This model can then be used immediately for visual tracking and scene annotation, and can be stored and later combined with other reconstructions to improve outdoor AR experiences. We have shown that image-based modeling from user-contributed panoramas is improved by use of orientation sensors and image processing techniques. Our absolute pose estimation method is robust and accurate while being simple to compute, and so would be useful for visual tracking on a mobile device with orientation sensors. Our evaluation shows that when using orientation sensors, some amount of filtering and correction is needed to maintain an accurate estimate of vertical orientation for pose estimation purposes. In the future we would like to investigate the application of our methods to mobile localization and real-time tracking for outdoor augmented reality applications.

We also aim to investigate dense reconstruction using upright panoramas. For example, previously a piecewise planar reconstruction of urban environments has been produced using street-level panoramic sequences [12]. The upright constraint and knowledge of the vertical vanishing point in all images might be used to simplify the reconstruction and make it more efficient.

Acknowledgments

Thanks to Chris Coffin and Sehwan Kim for preparing the tripod panorama datasets, and to Google, Inc. for providing the Street View datasets. This work was partially supported by NSF CAREER grant IIS-0747520.

References

1. Antone, M., Teller, S.: Scalable extrinsic calibration of omnidirectional image networks. Int. J. Comput. Vision **49**, 143–174 (2002)
2. Baatz, G., Köser, K., Chen, D., Grzeszczuk, R., Pollefeys, M.: Handling urban location recognition as a 2d homothetic problem. In: K. Daniilidis, P. Maragos, N. Paragios (eds.) Computer Vision – ECCV 2010, *Lecture Notes in Computer Science*, vol. 6316, pp. 266–279. Springer Berlin / Heidelberg (2010)
3. Fraundorfer, F., Tanskanen, P., Pollefeys, M.: A minimal case solution to the calibrated relative pose problem for the case of two known orientation angles. In: Proceedings of the 11th European conference on Computer vision: Part IV, ECCV'10, pp. 269–282. Springer-Verlag, Berlin, Heidelberg (2010)
4. Gallagher, A.C.: Using vanishing points to correct camera rotation in images. In: Proceedings of the 2nd Canadian conference on Computer and Robot Vision, CRV '05, pp. 460–467. IEEE Computer Society, Washington, DC, USA (2005)
5. Haralick, R.M., Lee, C.N., Ottenberg, K., Nölle, M.: Review and analysis of solutions of the three point perspective pose estimation problem. Int. J. Comput. Vision **13**, 331–356 (1994)
6. Hartley, R.I., Zisserman, A.: Multiple View Geometry in Computer Vision, second edn. Cambridge University Press, ISBN: 0521540518 (2004)
7. Horn, B.K.P., Hilden, H., Negahdaripour, S.: Closed-form solution of absolute orientation using orthonormal matrices. Journal of the Optical Society of America **5**(7), 1127–1135 (1988)

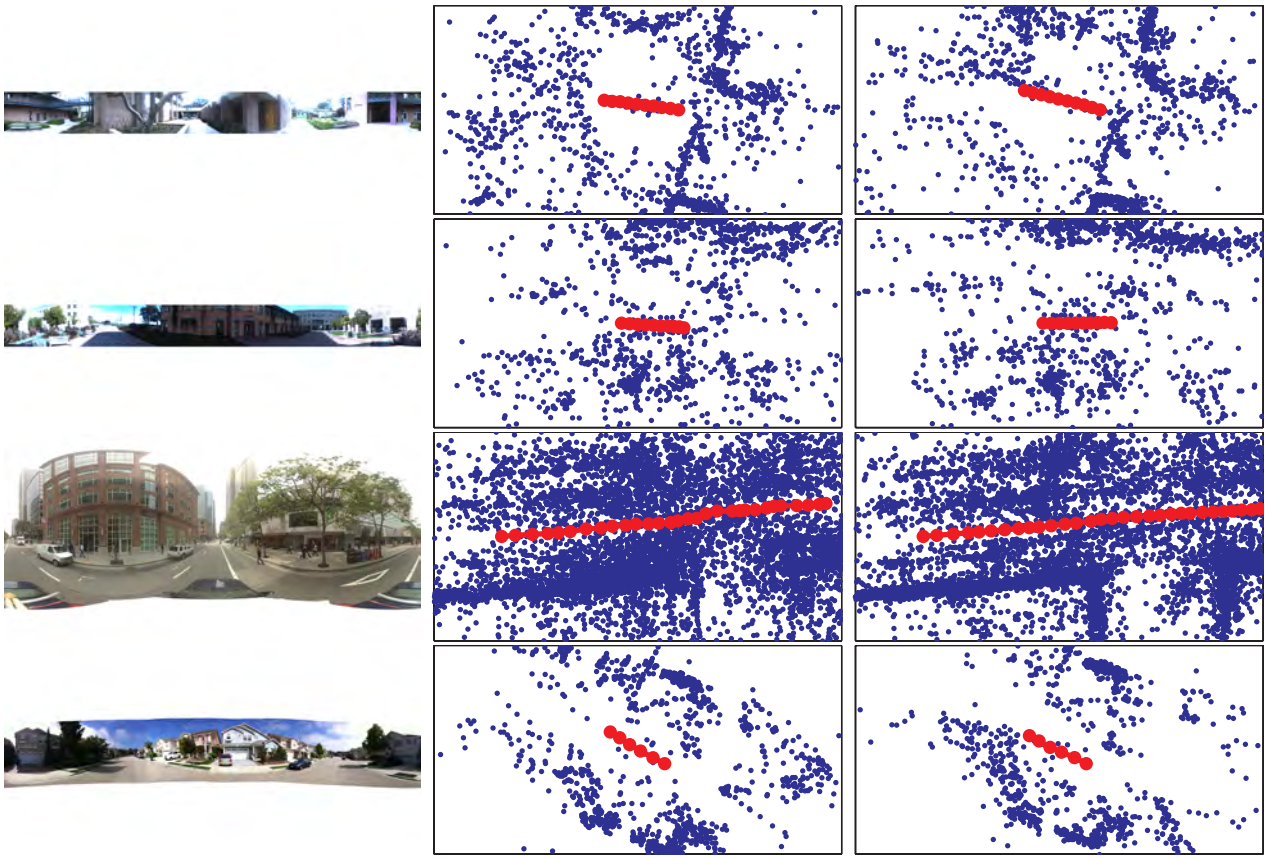


Fig. 9: Top-down view of reconstructions. Each row presents a separate reconstruction from a panoramic sequence, with a sample panorama shown in the left column. The middle column shows the reconstruction without vanishing point alignment, and the right column shows the result with vanishing point alignment. Camera locations are shown as red circles and triangulated points as blue dots. *From top to bottom:* Scenes 1 and 2 from our campus; Google Street View dataset; handheld smartphone panorama sequence.

8. Kosecka, J., Zhang, W.: Video compass. In: Proceedings of the 7th European Conference on Computer Vision-Part IV, ECCV '02, pp. 476–490. Springer-Verlag, London, UK, UK (2002)
9. Kukelova, Z., Bujnak, M., Pajdla, T.: Closed-form solutions to minimal absolute pose problems with known vertical direction. In: Computer Vision – ACCV 2010, *Lecture Notes in Computer Science*, vol. 6493, pp. 216–229 (2011). DOI 10.1007/978-3-642-19309-5_17
10. Lepetit, V., Moreno-Noguer, F., Fua, P.: Epnp: An accurate o(n) solution to the pnp problem. Int. J. Comput. Vision **81**, 155–166 (2009)
11. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision **60**, 91–110 (2004)
12. Micusik, B., Kosecka, J.: Piecewise planar city 3d modeling from street view panoramic sequences. Computer Vision and Pattern Recognition, IEEE Computer Society Conference on **0**, 2906–2912 (2009)
13. Nistér, D.: An efficient solution to the five-point relative pose problem. IEEE Trans. Pattern Anal. Mach. Intell. **26**, 756–777 (2004)
14. Pollefeys, M., Nistér, D., Frahm, J.M., Akbarzadeh, A., Mordohai, P., Clipp, B., Engels, C., Gallup, D., Kim, S.J., Merrell, P., Salmi, C., Sinha, S., Talton, B., Wang, L., Yang, Q., Stewénius, H., Yang, R., Welch, G., Towles, H.: Detailed real-time urban 3d reconstruction from video. Int. J. Comput. Vision **78**, 143–167 (2008)
15. Robertson, D., Cipolla, R.: An image-based system for urban navigation. In: British Machine Vision Conference, pp. 819–828 (2004)
16. Rother, C.: A new approach to vanishing point detection in architectural environments. Image and Vision Computing **20**(9-10), 647 – 655 (2002)
17. Snavely, N., Seitz, S.M., Szeliski, R.: Photo tourism: exploring photo collections in 3d. In: ACM SIGGRAPH 2006 Papers, SIGGRAPH '06, pp. 835–846. ACM, New York, NY, USA (2006)
18. Tardif, J.P., Pavlidis, Y., Daniilidis, K.: Monocular visual odometry in urban environments using an omnidirectional camera. In: IROS'08, pp. 2531–2538 (2008)
19. Torii, A., Havlena, M., Pajdla, T.: From google street view to 3d city models. In: Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on (2009)
20. Werner, T., Pajdla, T.: Cheirality in epipolar geometry. Computer Vision, IEEE International Conference on **1**, 548 (2001). DOI http://doi.ieeecomputersociety.org/10.1109/ICCV.2001.10062