A New Scale Arbitration Algorithm for Image Sequences Applied to Cylindrical Photographs

Mark A. Livingston, Irwin E. Sobel
Mobile and Media Systems Laboratory
HP Laboratories Palo Alto
HPL-2001-226 (R.1)
May 19th , 2003*

camera calibration, colorimetry

We chose cylindrical cameras for our work in reconstructing a scene that surrounds the camera positions. While previous algorithms for calibration exist, we found that improvements were needed to give reasonable results for our data, especially in the determination of the relative scale of translations between the cylinders.

One approach to determining camera motion is from corresponding elements in the images, for example points or lines. One class of solutions is based on finding a transformation that yields coplanarity of certain vectors. This type of solution is not limited to any particular form of camera. This algorithm adapts to omnidirectional imagers with minor modifications in the constraint matrices. But the algorithm can only determine five degrees of freedom. A scale arbitration algorithm must determine the sixth degree of freedom, the relative lengths of the translation.

While robust algorithms have been proposed for determining the relative scale, we found that standard robust statistics were insufficient for registering our cylindrical camera locations. Thus we introduce a new scale arbitration algorithm that takes into account the confidence with which a point has been triangulated using a previous camera pair. It then decides how to use the point to determine scaling for future camera pairs. This issue arises in standard rectilinear images as well, and our confidence metric adapts to any imager and other solution methods.

We also address consistent illumination of cylindrical images and an ambiguity that can arise in the rotation and translation direction computations.

* Internal Accession Date Only
Approved for External Publication

A New Scale Arbitration Algorithm for Image Sequences Applied to Cylindrical Photographs

Mark A. Livingston and Irwin E. Sobel

Hewlett-Packard Laboratories, Palo Alto, California

Abstract

We chose cylindrical cameras for our work in reconstructing a scene that surrounds the camera positions. While previous algorithms for calibration exist, we found that improvements were needed to give reasonable results for our data, especially in the determination of the relative scale of translations between the cylinders.

One approach to determining camera motion is from corresponding elements in the images, for example points or lines. One class of solutions is based on finding a transformation that yields coplanarity of certain vectors. This type of solution is not limited to any particular form of camera. This algorithm adapts to omnidirectional imagers with minor modifications in the constraint matrices. But the algorithm can only determine five degrees of freedom. A scale arbitration algorithm must determine the sixth degree of freedom, the relative lengths of the translation.

While robust algorithms have been proposed for determining the relative scale, we found that standard robust statistics were insufficient for registering our cylindrical camera locations. Thus we introduce a new scale arbitration algorithm that takes into account the confidence with which a point has been triangulated using a previous camera pair. It then decides how to use the point to determine scaling for future camera pairs. This issue arises in standard rectilinear images as well, and our confidence metric adapts to any imager and other solution methods.

We also address consistent illumination of cylindrical images and an ambiguity that can arise in the rotation and translation direction computations.

1 Introduction

Computing the relative motion of a camera between the acquisition of two or more images is one of the classic problems in computer vision [2, 4, 7, 18]. Such algorithms are often the first stage in reconstructing a dense model of the scene.

Many scene reconstruction algorithms rely on camera calibration to yield sufficient information for precise reconstruction of 3D geometry. One such approach is the technique of *voxel coloring* [20]. This technique has been extended to handle arbitrary camera placement [3] and to encompass surrounding geometry [21]. However, the fundamental requirement of calibrated cameras remains constant through the variations on the algorithm.

One approach to determining camera motion is from corresponding elements in the images, for example points or lines. There are algorithms to estimate the relative motion between two images from correspondence data, but these algorithms are limited to recovering five degrees of freedom [4, 7]. In order to register the pairwise (if two-view geometry descriptors are used) relative motions during acquisition of a sequence of images into a single coordinate system, the scale of each translation must be determined. One method of measuring the relative scale is to triangulate the points in the reference image pair and the points in the new image pair (presumably one image is shared between the pairs), then scale the new calculation of the points into the previous one. Proceeding along the sequence in this manner, the initial estimate will encompass six degrees of freedom of each of a set of cameras (or a single camera at multiple time steps) within a single coordinate system. Once a common coordinate frame is determined, an optimization procedure can refine this estimate into an optimal arrangement under some error metric. However, the search space can be of very high dimension if many images are used, and thus the quality of the initial estimate is very important to the success of the optimization procedure.

It requires only two points shared in three images to determine a scale factor for the relative translation of all three images, but commonly multiple points are used [13]. We found that standard robust statistics were insufficient for registering the relative scale of our data, a set of cylindrical images. Thus we introduce a new scale arbitration algorithm that accounts for the likelihood that a point has been accurately triangulated by a previous camera pair. It then decides how to use the point to determine scaling for future camera pairs. This issue arises in standard rectilinear images as well, and our algorithm can be used for any imager.

We also address other issues in the processing pipeline we use to prepare cylindrical images for dense reconstruction.

1.1 Overview of our algorithm

Assume for a moment that we have correctly identified (noisy) correspondences across three images, and that we are ready to determine a scale factor that will register the third camera location into the coordinate system defined by the first camera location, with the second camera location set to some standard distance, and that we measured the third camera location relative to the second to within the scale of the translation. We register the two transformations with the following algorithm.

1. Triangulate the points seen in the first two images to form a reference point set. Assign a confidence metric to all points.
2. Triangulate the points in the second and third images, including at least two points that were in the reference point set. Assign a confidence metric to all points.
3. Find the two points with the highest product of the confidence metrics from the two triangulations.
4. Register the second point set using these two points to set the scale.
5. For points in both the new and reference sets, update the estimated positions of the triangulated points in

the reference set if (and only if) the confidence metric is greater in the new set.

This algorithm is applied for each new image, with the reference point set growing as new points are added from new images.

The basis for our algorithm is the simple observation that triangulation from two rays works best when the two rays are nearly perpendicular. Thus our confidence metric for a point reflects the angle at which the two rays (nearly) intersected. We choose the absolute value of the sine between the two rays, but obtain similar results using one minus the cosine. To reflect the performance of triangulation in both sets, we use the product of the two confidence scores (where a missing point has zero confidence). We have not found a threshold for confidence in either set or in the product to be necessary to achieve good performance. The two points with the greatest confidence should provide the best estimate of the relative scaling between the two points sets, and thus between the translations measured for the two image transformations. We can then update the reference point set for processing of the next image in the sequence by replacing the point locations of those point triangulated with greater confidence in the new point set.

1.2 Organization of the manuscript

Section 2 summarizes the geometric imaging model we use for our cylindrical camera and the geometric and photometric calibration process. We summarize previous solutions for the scaling problem and introduce our new scaling algorithm in Section 3. We conclude with discussion of our technique and its application to rectilinear images and other scaling algorithms in Section 5.

2 Imaging Model and Calibration

The imaging takes place within a camera-centric coordinate system (Figure 1). For each camera location, we must also measure the position and orientation of the camera-centric system with respect to the global coordinate system. Pixel coordinates (u, v) are calculated from the world-space point $[\ x_w\ \ y_w\ \ z_w\ ]^T$ with the following set of equations based on the geometry in Figures 1, 2, and 3.

$$\begin{aligned}
\begin{bmatrix} x \\ y \\ z \end{bmatrix} &= [R \mid -Rt] \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix} \\
v_n &= \frac{f \cdot z}{d_y \cdot \left(\sqrt{x^2 + y^2} - E_r \right)} \\
v &= C_v - v_n \cdot (1 + \kappa v_n^2) \cdot N_v \\
u_n &= \frac{\arctan\left(\frac{y}{x}\right)}{2\pi} \\
u &= N_u - (u_n + \sigma * v_n) * N_u
\end{aligned}$$

The vertical component uses the standard perspective projection equations with focal length f, image center C_v, image height in pixels N_v, and image height in millimeters d_y. One difference from Tsai's camera model [22] is that the distance is expressed as distance in the xy-plane, and the height is the z coordinate. We use a single-parameter (denoted κ) radial lens distortion model, but since the imaging device is a linear CCD, the "radial" distance is really only the vertical distance from the center. Our vertical field of view was approximately 78° and every pixel is in the horizontal center of the field of view; thus we find one radial term to be a sufficient model for distortion.

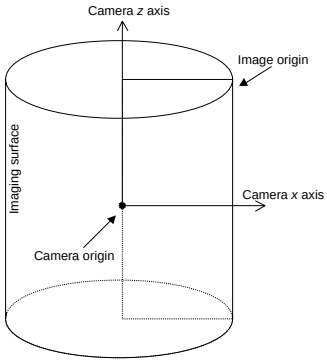

Figure 1: Diagram of the camera coordinate systems. The camera y axis is not shown, but is derived by the right-hand rule. The image coordinate origin is at the top of the cylinder and has a y coordinate of 0.0.

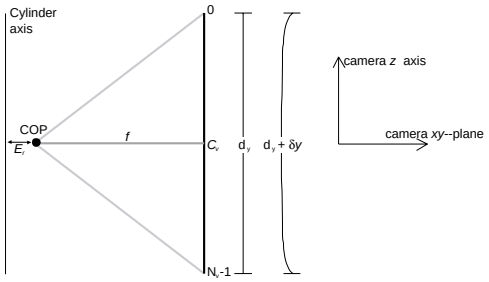

Figure 2: The typical perspective projection situation for cameras applied to the vertical component of the cylindrical imaging system. We use a single-parameter "radial" distortion model, which is restricted to vertical distance since the CCD is linear. The pixels are numbered from 0 at the top to $N_v - 1$ at the bottom.

The rotating head of the camera enables the linear CCD to acquire the same type of image at every step around the cylinder. Thus basic trigonometry determines the horizontal image coordinate using the number of pixels N_u around the cylinder. Cylinder skew σ is non-zero when the linear CCD is not parallel to the cylinder axis, but rather leans along the direction of the (positive or negative) horizontal tangent to the cylinder.

Eccentricity E_r accounts for the fact that the camera lens may not be mounted on the camera and tripod assembly such that the center of projection lies on the cylinder axis. This is shown in Figures 2 and 3 with the center of projection offset from the axis in the imaging direction ($E_r > 0$), but the offset could occur in the opposite direction ($E_r < 0$).

We did not include a parameter to account for an angular difference between the direction of the rays and the cylinder normal [11]. We found this parameter to be unnecessary with our images. In fact, we find the skew parameter to be negligible as well.

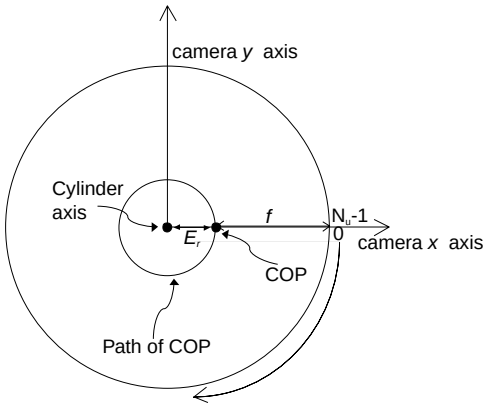

Figure 3: The imaging geometry in the horizontal direction of the cylinder. The quantities labeled in *slanted* typeface indicate intrinsic parameters of the camera that must be determined during calibration. Other quantities can be stated from manufacturing specifications. The pixels are numbered from 0 to $N_u - 1$, with the camera x axis intersecting the border of the first and last pixels.

2.1 Calibration procedure

Once we understand the imaging geometry, we can now set about the task of determining the parameters of the model. We require three camera constants as inputs:

- the number of pixels horizontally, N_u
- the number of pixels vertically, N_v
- the height of the imaging array in millimeters, d_y

We have a set of five intrinsic parameters to generate as output. These are assumed to be the same for all images.

- the focal length f
- vertical image center pixel C_v
- eccentricity E_r
- radial distortion parameter κ
- cylinder skew parameter σ

For each image, the calibration algorithm must determine six extrinsic parameters (three rotation DOFs and three translation DOFs). The initial estimates for the intrinsic parameters are derived from manufacturer's specifications. We use the essential matrix method as adapted to cylindrical images by Kang and Szeliski [13] to compute an initial estimate for the extrinsic parameters of the second and succeeding cylinders; the first cylinder is assigned an orientation coincident with the world orientation and at the world origin.

2.1.1 Adapted Eight-Point Algorithm

The calibration algorithm we use takes as input 2D coordinates of corresponding points in some number of the input images. We specified our correspondences manually, but an automatic tracking algorithm could be used as well [13].

The Eight-Point Algorithm [14] computes the *essential matrix* which maps points in one image to epipolar lines in another image, on which the corresponding point must lie. The analogous situation for cylindrical imaging surfaces is that the coordinates should lie on an epipolar curve [11, 16]. But the Eight-Point Algorithm expresses only the coplanarity of three vectors: the two rays that correspond to the two pixels (emanating from the respective centers of projection) and the ray that points from one center of projection to the other. Kang and Szeliski [13] give the new constraint equation for cylindrical images (which we convert to the imaging geometry we use).

$$\left[u', v', w'\right] \mathbf{E} \begin{bmatrix} u \\ v \\ w \end{bmatrix} = \vec{0}$$

where

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = \begin{bmatrix} f\cos(2\pi u_n) \\ -f\sin(2\pi u_n) \\ v_n \end{bmatrix}$$

and similarly for the corresponding (primed) point. They also caution that the vector $[u, v, w]^T$ should be normalized to reduce the sensitivity of the constraint matrix, similar to Hartley's transformation for the original Eight-Point Algorithm [8]. We solve the resulting linear system for the elements of $\mathbf{E}$. This procedure is wrapped inside a RANSAC [5] to identify outliers in the correspondence data. This assumes that we will have sufficient inliers for a minimal solution and that those points will constitute at least one sample set in the RANSAC procedure.

2.1.2 Decomposing E

There are four possible ways to decompose $\mathbf{E}$ into a rotation and translation that would yield the given essential matrix [7]. These can be determined using the singular value decomposition (SVD) of the essential matrix, $\mathbf{E} = U\Sigma V^T$. The direction of translation is given to within a sign by the left singular vector in the column of U associated with the smallest singular value (which should be zero, since the essential matrix is rank-2). Independent of the correct sign, the rotation is given by either UWV^T or UW^TV^T, where

$$W = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

Only one of these four possibilities will reconstruct the points in front of both cameras. We found that this decomposition did not successfully identify any solution as reasonable for some matrices and SVDs. The problem comes from the fact that the essential matrix has one unique singular value, and thus a family of SVDs. Using the Numerical Recipes [19] SVD computation gave us matrices that had U and V matrices with determinants of -1. Zucchelli and Christensen [23] parameterize the matrix W with the signs of the determinants:

$$W = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & \det(V)\cdot\det(U) \end{bmatrix}$$

But this method did not work for all our data. Independently, we had already adapted the algorithm by forcing the signs of the determinants to be $+1$ by negating the entire matrix U or V (or both) when the respective determinants are -1. This is equivalent to the original formulation if the determinants of U and V have the same sign, and equivalent to negating W when the signs of the two determinants differ (as the method of Zucchelli and Christensen does). This yields a reasonable decomposition for all our image pairs.

The remaining step, determining the scale of the translation, will be discussed in Section 3.

2.1.3 Parameter refinement

Starting with the initial estimate, we refine all extrinsic and intrinsic parameters using the coplanarity constraint described above. The cost function for each image of a point seen in a pair of images is the distance from the epipolar plane.

$$\text{Cost}(i,j,k) = (T_k - T_j) \cdot (R_j^T \text{Ray}(u_{ij}, v_{ij}) \times R_k^T \text{Ray}(u_{ik}, v_{ik}))$$

$$\Sigma_{i,j,k} = w_i \cdot (\text{Cost}(i,j,k))^2 \quad (1)$$

This sum is taken over all points i and all cylinders j and k that see point i, subject to $j < k$. Thus each unique point-cylinder-cylinder tuple contributes only once to the error computation. We have also introduced a weighting factor, w_i. Each distance is squared and weighted by the angle between the baseline used for triangulation and the ray emanating from the first cylinder.

$$w_i = 1 - \cos\phi$$

The "first" is chosen as the cylinder with the lower identification number of the two, and the ID numbers are assigned in the order in which the cylinders were acquired. Triangulation is well-known in computer vision to be numerically sensitive. This weighting function decreases the importance of correspondences for which the two lines of sight are nearly parallel to the baseline used for triangulation, which is a configuration that is numerically unstable. This is the observation that sparked our new scaling determination algorithm.

We use Powell's multi-dimensional optimization method with an exhaustive 1D search in the inner loop. This algorithm does require numerous function evaluations and thus is computationally expensive, but it is unlikely to get caught in a local minimum that is not the global minimum.

2.2 Photometric correction

One significant assumption of many scene reconstruction algorithms is that all surfaces are diffuse. The rotating-head camera requires approximately 90 seconds to capture a panorama with about 30 degrees of overlap. We thus had two conflicting goals: to shoot when few non-static objects (people and strong shadows) were in the environment and to shoot in constant illumination. We chose to acquire image in the early morning and provide a software correction for illumination change and to acquire multiple panoramas from a single location in order to paint out any non-static elements.

We used a light meter, we found that the light changed almost linearly near sunrise (Figure 4). We adjusted the exposure value between photographs and use the following linear brightness correction function within a single panorama.

$$\text{mask}_i(\lambda) = 1 + \lambda \cdot \frac{(N_u - 1 - i)\%N_u}{N_u}$$

where % indicates the modulus operator. The correction is performed on each color channel and clamped to the appropriate range. The variable i indicates the horizontal pixel coordinate and ranges from 0 to $N_u - 1$. The parameter λ was experimentally determined to be 0.87 for our data.

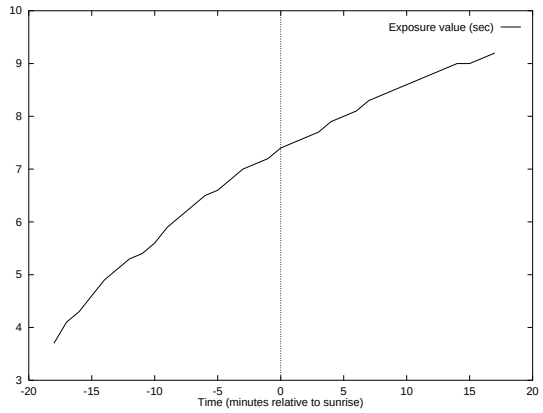

Figure 4: Light meter readings during the time of the panorama acquisition. The horizontal axis indicates time relative to sunrise. We started shooting just before sunrise and took 30 minutes to acquire the ten clean cylinders.

Lens vignetting results in inconsistent colors for surfaces near the camera positions. To correct vignetting, a cosine-fourth function is a common model for the light fall-off, although we and others [1] have found it is not always a sufficient model. We use a tangent-squared correction instead.

$$\text{mask}_j(\gamma) = 1 + \left(\frac{1}{\gamma} - 1\right) \cdot \tan^2\left(\frac{2j}{N_v} - 1\right)$$

Again, each channel is corrected separately and clamped. The variable j indicates the vertical pixel coordinate and ranges from 0 to $N_v - 1$. The parameter γ was experimentally determined to be 0.61 for our data.

3 Scale Arbitration Algorithm

As noted earlier, only two points across three images are required to estimate the scale factor that registers the pairwise translations between camera locations. So the simplest algorithm is to find two such points, triangulate them from the first two and last two images, then use the distance between them in each reconstruction space to determine the relative scale of the two spaces. These reconstruction spaces include the camera locations. First, we summarize previous algorithms to improve upon this basic algorithm, then we examine the problem we faced in our cylindrical image set, and finally introduce our new algorithm.

3.1 Previous Scale Determinations

One can use absolute measurements to determine the true scale of a reconstructed data set, such as using a motion platform to control the precise translation distance between image locations [15]. One could also use known 3D points or a prior scene model [10] to arrive at a Euclidean reconstruction (true scale).

Since relative scale is all that is necessary, one can fix the scale of the second and succeeding translations by using the estimated 3D locations from the first pair of images. This is essentially switching to a 3D-from-2D calibration method for the third and later images [2, 18]. In a similar manner, one can find a scale factor that minimizes the distance between corresponding points in 3D or in image space [6, 17], for the

best fit half of the points [13], or by using a robust statistic such as a least median error metric [12]. A trade-off between the distance of the baseline and the number of shared correspondences can be established between two frames [17], but requires making assumptions about the camera motion.

3.2 Issues in Scale Arbitration

The problem in determining scale is that the algorithm must rely on the accuracy of the points that actually determine the scale factor (i.e. but not outliers). We found that with the 3D points somewhat sparse in regions of the volume surrounding the cameras, there were many more outliers than inliers when triangulating the points. This problem is compounded when the two view rays are nearly parallel, which results in an ill-conditioned triangulation problem. For an omnidirectional imager, this situation can arise for any translation. For standard rectilinear imagers, this is most likely to arise when the motion has a strong component perpendicular to the image plane and the points are concentrated in the center of the image. In either situation, the points are difficult to accurately locate in 3D. The error can grow large quite quickly when the two rays are nearly parallel, and thus it becomes difficult to bound the error created by comparing the distances to determine a relative scale. It is this difficulty we seek to overcome.

3.3 New Scaling Algorithm

We introduce a confidence metric into the scaling algorithm that indicates the certainty with which a point was triangulated and indicates which points in a set we can use to determine the scale with the most confidence. The confidence metric is simply the absolute value of the sine between the two view rays. This vanishes when the view rays are parallel and reaches 1.0 when the view rays are perpendicular, which is the best situation for triangulation from two rays [7].

We integrate this metric into the scaling algorithm to produce the following scale arbitration algorithm.

1. Compute an initial reference point set using the first pair of images. Compute the confidence metric for all points.

2. For each successive cylinder

   (a) Triangulate the points seen in the new image pair, which consists of one previous image and one new image.

   (b) Compute the confidence metric for the new triangulation of the points.

   (c) Find the two points with the highest product of the confidence metrics from the two triangulations. These two points must have a positive confidence.

   (d) Determine the scale of the second camera translation using these two points.

   (e) For each point in the new point set

      i. If the point does not exist in the reference set, insert it.

      ii. If the point does exist and has higher confidence in the new set is greater, update the point location and its confidence metric.

When we multiply two confidence values together, the range is still [0.0,1.0]. It will be 1.0 if we have complete confidence in both triangulations of the point and degrades to zero as one or both triangulations are computed with more and more parallel view rays. When we update the computed positions and the confidence metric, we do not use the product of the old and new metrics, just the greater of the old and new. This completes the processing for the current frame, and we proceed to the next frame.

In processing omnidirectional sequences, we can hope to find most points occurring in every image. If this is the case (as it is in our data), we can compute every transformation relative to the first image, rather than computing each transformation relative to the most recent image. This opens the possibility of extending the scale arbitration algorithm. If we record the confidence values for the two points that were used for determining the scale, we can go back to each image and see if we can improve on that confidence metric with points that were triangulated (again) after the image was originally processed. In fact, this can be applied to any image sequence in which points are tracked through many images. The longer a point stays in the image, the more chances it has to be triangulated accurately from a varying camera pose. In the case of sequential processing where each frame is processed relative to the last, updating the scale of a translation several frames back in the sequence will cause a ripple effect of translations up to the current frame.

As a final post-process we optimize the scale factors for the second and succeeding images using the cost function (Equation 1) that we use in the final stage of optimizing the full extrinsic parameter set. This allows a bit of refinement using the fact that many points are seen in more than three cylinders, so the scale factors can be refined between all cylinders at once, rather than the pairwise manner in which they were initially estimated. We use the Levenberg-Marquardt optimization procedure [19]. We can guide the optimization process by using the confidence values to derive an estimated variance for each parameter.

4 Results

We acquired ten panoramas (Figure 5). The images have approximately 2500 pixels around the cylinder and 884 pixels vertically. We manually identified edges that appeared in the overlap region for each cylinder and measured the distance between the edges to the nearest whole pixel. This number gave us N_u for each cylinder, and we cropped the images at the measured length. For certain cylinders, there were objects that were imaged that were either moving at the time of the imaging, or were not present for other cylinders. We acquired multiple cylinders from a single camera location in order to have enough data to "paint out" these objects; they would violate an important assumption made in the reconstruction process: that the scene is static.

Table 1 compares the three algorithms to the ground truth. In some sense, this is an impossible comparison, since the direction of translation is not perfectly computed. However, it does provide a useful view of the performance of the various algorithms. The results are shown only for the second and succeeding cylinders. The first cylinder is placed at the origin, and the first cylinder is assigned a distance. We used the true distance for the first cylinder (from the tape measurements), but could also have used 1.0.

The decomposition of the essential matrix uses a unit length vector to describe the translation, so the scale factors shown here represent the actual or estimated distance

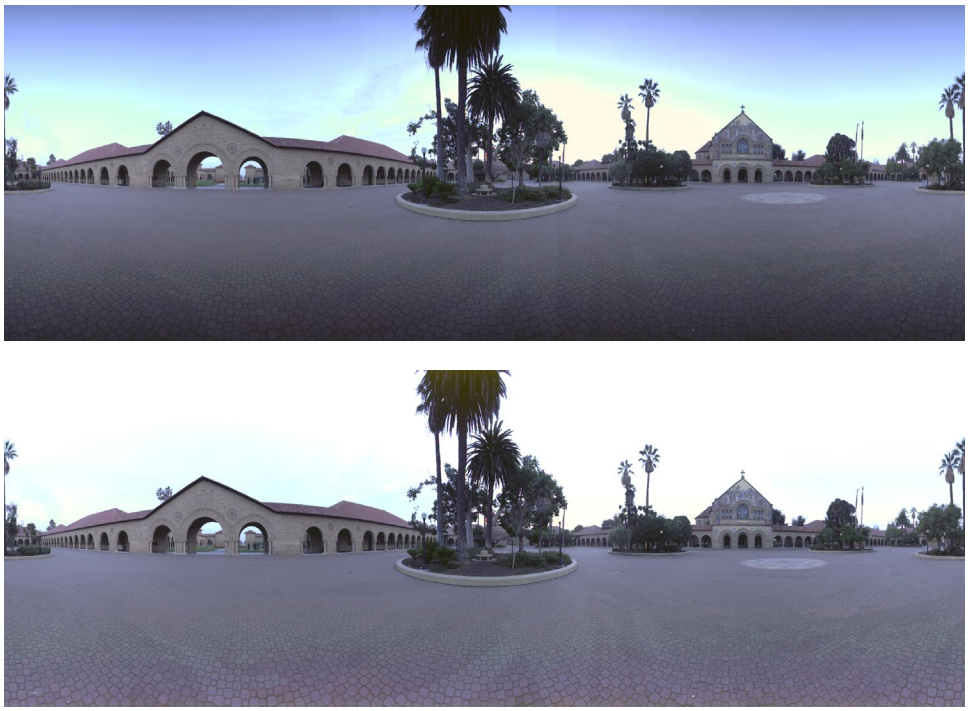

Figure 5: A cylinder from our data set. Top: before photometric correction. Note the vignetting at the top and bottom of the image, and the vertical seam just right of the center. This is the rotated image to align the first pixel with the world x axis, so the linear illumination change creates a seam at the pixel which was originally the first pixel. Bottom: after photometric correction (and still rotated).

of the translation. The table lists the true cylinder distances, the estimated distance from the robust algorithm that uses 50% of the points, the confidence metric from our algorithm, the estimated distance from our algorithm, the relative error in the estimated scale (computed as the estimated length divided by the true length), and the result from an optimization pass that uses our algorithm as input and optimizes the distances for the second and succeeding images. The ground truth was acquired with a tape measure. We give only the scale (distance) of translation, not the translation, since our algorithm is computing only the scale of translation. The direction of translation and the rotation are identically computed for all methods with the adapted Eight-point Algorithm.

The table shows that the previously described robust algorithm failed badly on certain cylinders. This occurs because the assumption made in the robust statistics is that a certain percentage of the data (in this case, 50%) is within a tolerance of the correct answer (in this case, point location). This assumption is likely to hold when many points are uniformly sampled in the imaged volume. The imaging geometry of the cylinders and the lack of reasonable tracking targets available in certain regions of the scene (e.g. in the highly repetitive brick structure and in the sky) prevented this assumption from being valid on our data. Thus our new algorithm, which attempted to find the best two (i.e. minimum number of necessary) points, outperformed the previous algorithm.

This difference is shown graphically in Figure 6, in which the xy positions of the camera locations are plotted. Note that the robust algorithm placed most of the cameras very near the origin, although the shape of the point set is similar (in the geometric sense) to the shape of the point set that depicts the new algorithm before optimization.

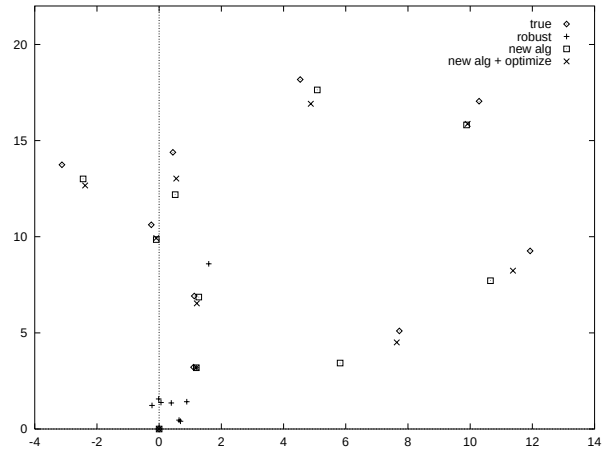

Figure 6: Plot comparing the xy positions of the camera locations. The robust algorithm yields a geometrically similar layout, but most points are clearly poorly scaled and left near the origin. The new algorithm and the optimization of its result much more closely match the true arrangement.

The relative length is consistently below 1.0 for the new algorithm; that is, the new algorithm underestimates the distances. The error after the optimization pass correlates well, however, with the confidence metric, except for cylinder #5. This indicates that on the whole, our algorithm has done a good job of characterizing which points are the most reliable for determining scale. The optimization procedure yields a

Cyl#	Correct scale	Robust statistics	Conf	New Algorithm	Relative Length	Optimized	Relative Length
2	7.007	8.742	0.670	6.981	0.996	6.648	0.949
3	10.625	1.565	0.542	9.854	0.927	9.930	0.935
4	14.091	1.245	0.101	13.233	0.939	12.886	0.914
5	14.395	1.381	0.746	12.198	0.847	13.035	0.905
6	18.736	1.410	0.145	18.354	0.980	17.602	0.939
7	19.912	1.675	0.231	18.648	0.937	18.713	0.940
8	15.101	0.789	0.166	13.152	0.871	14.044	0.930
9	9.256	0.797	0.712	6.756	0.730	8.868	0.958

Table 1: Results comparison of the algorithms. For each cylinder, we give the correct scale, the result from the previous robust algorithm, the confidence metric from our new algorithm, the estimated scale from our new algorithm, the relative error (estimated scale divided by true scale) and the result of an optimization using our algorithm's output for input. We give the scale (in meters), not the direction of translation, since that is all the new algorithm computes. Except for cylinder #5, the confidence correlates well with the relative error.

mixed result; it tends to smooth the relative error distribution rather than simply pull the less-accurately estimated cylinders towards the correct scale. Since each cylinder's transformation is computed relative to the first cylinder, there is no cascading effect from a previously scaled cylinder. (Cylinder #4's low confidence has no effect on cylinder #5's lower accuracy.) An early implementation which also allowed the scale factor of the first cylinder to be modified had marginally better success. We are still investigating these phenomena.

5 Discussion

The success of this method comes from not assuming a percentage of the data will be reliable. We have demonstrated an improvement by not relying on anything more than the minimum number of necessary reliable points to exist in the data, and providing a method to identify those reliable points.

The assumption made in this method is that correspondences truly represent a single 3D point. If correspondences errors occur in points that are near the initial camera position and thus have significant disparity over the first few frames, this method will still label those points as reliable for determining scale. Such an error could be detected by checking the consistency of the optical flow against the measured camera velocity, but this would still be unlikely to lead to satisfactory performance.

While it is true that many algorithms would have difficulty with this scenario, our algorithm as stated thus far would in fact be more prone to error since it relies on only two points. This would suggest that a weighted least squares method (perhaps using this metric in combination with metrics regarding the confidence in assigning the correspondence) would outperform this method. In fact, we implemented a weighted least-squares algorithm using just our confidence metric as the weights. The results are shown in Table 2. On this data set (with no outliers in the correspondence data), both methods should and do perform well. We argue that this demonstrates the usefulness of the metric, and that it should be adapted to a robust method that uses all points according to a confidence metric, such as weighted least-squares. We also compare against a method in which the scale is recomputed using only those points which are within three standard deviations of the mean of the estimate computed from all points. this two-pass method should be more robust, and it does improve significantly the estimate for the scale of the final cylinder, with mixed results on the other cylinders. Note that since we are determining the sixth degree of freedom, we expect that outliers will have already been removed in the determination of the first five degrees of freedom; however, the scale can be prepared for outlier removal to fail in the earlier processing.

The certainty for triangulation could be characterized by the size of the intersection of two solid angles which represent the certainty of the pixel locations. One could use the volume or the largest diameter of this intersection as a measure of certainty. We find that we get good performance from simply using the sine of the angle. Whether a similar measure such as the volume of the intersection would more precisely indicate confidence and lead to better scale estimates remains to be seen.

This confidence metric can be applied to non-cylindrical imaging systems. It does not rely on the imaging geometry; like the adapted Eight-Point Algorithm, it uses only vector operations which can be defined for any imaging system. This metric could in fact be incorporated into the other algorithms for determining the motion of the second and succeeding images from a sequence. For example, in the methods where the complete projection matrix is determined from the 3D triangulated points as computed from the first two images [2, 18], a weighted least squares solution could be implemented using this confidence metric as the weight. Similarly, an image-plane distance metric to minimize reprojection error of points [6, 17] could be converted into a weighted metric. Further, in such a method, the confidence in triangulation could be converted into a variance on the point location, and the distance error could in fact be considered to be zero if its projection falls within that variance.

We have thus far assumed the variance on all the correspondence coordinates is identical; however, if it is not, then this information can be used to derive more specific confidence measures. We could also take into account the distance from the cameras the point is. The angular error is fixed, best it is multiplied by the moment arm from the camera to the point. This would help minimize the angular error in reprojection [9].

The novel feature of this algorithm is in its method of selecting the best points to use for determining the scale. By doing this, it avoids making any assumption about the percentage of accurately triangulated points. This enabled the new algorithm to produce a much more accurate initial estimate for the relative scale of translations between camera locations for our image sequence. The risk it takes is that the outliers in the correspondence will not have been identified during the computation of the essential matrix. Given

Cyl#	Correct scale	Conf	New Algorithm	Rel. Length	WLS	Rel. Length	Two-pass	Rel. Length
2	7.007	0.670	6.981	0.996	6.934	0.990	6.983	0.997
3	10.625	0.542	9.854	0.927	9.990	0.940	9.826	0.925
4	14.091	0.101	13.233	0.939	13.041	0.925	13.256	0.941
5	14.395	0.746	12.198	0.847	12.207	0.848	11.907	0.827
6	18.736	0.145	18.354	0.980	17.722	0.946	18.453	0.985
7	19.912	0.231	18.648	0.937	19.502	0.979	19.095	0.959
8	15.101	0.166	13.152	0.871	13.681	0.906	13.803	0.914
9	9.256	0.712	6.756	0.730	4.909	0.530	7.010	0.757

Table 2: Comparison of new method using only two points versus new metric used as the weight in a weighted least-squares algorithm, and in a two-pass algorithm that computes the mean and standard deviation of the estimated scale using all (weighted) points, then removes those points more than three standard deviations from the mean and recomputes the estimate with the remaining points. Since there are no outliers in the data, we expect and see few differences in the results.

our success in solving this part of the problem, we feel confident that our method will correctly identify the scale factor necessary to register the data.

References

[1] Aggarwal, M., and Ahuja, N. A new imaging model. In *Proceedings of the Eighth International Conference on Computer Vision (ICCV-2001)* (July 2001), vol. 1, pp. 82–89.

[2] Beardsley, P. A., Torr, P. H. S., and Zisserman, A. 3D model acquisition from extended image sequences. In *Proceedings of the 4^{th} European Conference on Computer Vision, LNCS 1065, Cambridge* (1996), pp. 683–695.

[3] Culbertson, W. B., Malzbender, T., and Slabaugh, G. G. Generalized voxel coloring. In *Vision Algorithms: Theory and Practice* (Sept. 1999), pp. 67–74.

[4] Faugeras, O. *Three-Dimensional Computer Vision: A Geometric Viewpoint.* MIT Press, 1993.

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

[6] Fitzgibbon, A. W., and Zisserman, A. Automatic camera recovery for closed or open image sequences. In *Proceedings of European Conference on Computer Vision (ECCV 1998)* (June 1998), pp. 311–326.

[7] Hartley, R., and Zisserman, A. *Multiple View Geometry in Computer Vision.* Cambridge University Press, Cambridge, England, 2000.

[8] Hartley, R. I. In defence of the 8-point algorithm. In *Fifth International Conference on Computer Vision* (June 1995), IEEE, pp. 1064–1070.

[9] Hartley, R. I., and Sturm, P. Triangulation. In *Proceedings of Conference on Computer Analysis of Images and Patterns* (1995).

[10] Hsu, S., Samarasekera, S., Kumar, R., and Sawhney, H. S. Pose estimation, model refinement, and enhanced visualization using video. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)* (June 2000), vol. 1, pp. 488–495.

[11] Huang, F., Wei, S. K., and Klette, R. Geometrical fundamentals of polycentric panoramas. In *Proceedings of the Eighth International Conference on Computer Vision (ICCV-2001)* (July 2001), vol. 1, pp. 560–565.

[12] Kang, S. B. Catadioptric self-Calibration. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)* (June 2000), vol. 1, pp. 201–207.

[13] Kang, S. B., and Szeliski, R. 3-D scene data recovery using omnidirectional multibaseline stereo. Tech. Rep. 95/6, Digital Equipment Corporation Cambridge Research Lab, Oct. 1995.

[14] Longuet-Higgins, H. C. A computer algorithm for reconstructing a scene from two projections. *Nature 293* (Sept. 1981), 133–135.

[15] McMillan, L. Personal communication, Nov. 1999.

[16] McMillan, L., and Bishop, G. Plenoptic modeling: An image-based rendering system. In *SIGGRAPH 95 Conference Proceedings* (Aug. 1995), R. Cook, Ed., Annual Conference Series, ACM SIGGRAPH, Addison Wesley, pp. 39–46. held in Los Angeles, California, 06-11 August 1995.

[17] Nistér, D. Reconstruction from uncalibrated sequences with a hierarchy of trifocal tensors. In *European Conference on Computer Vision (ECCV 2000)* (June 2000), pp. 649–663.

[18] Pollefeys, M., Koch, R., and Gool, L. V. Self-calibration and metric reconstruction in spite of varying and unknown internal camera parameters. In *Proceedings of the International Conference on Computer Vision (ICCV '98)* (Jan. 1998), pp. 90–95.

[19] Press, W. H., Teukolsky, S. A., Vetterling, W. T., and Flannery, B. P. *Numerical Recipes in C: The Art of Scientific Computing*, 2nd ed. Cambridge University Press, Cambridge, England, 1988.

[20] Seitz, S. M., and Dyer, C. R. Photorealistic scene reconstruction by voxel coloring. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '97)* (June 1997), IEEE, IEEE Computer Society Press, pp. 1067–1073.

[21] Slabaugh, G. G., Malzbender, T., and Culbertson, W. B. Volumetric warping for voxel coloring on an infinite domain. In *Second Workshop on Structure from Multiple Images of Large Scale Environments (SMILE)* (July 2000), pp. 41–50.

[22] Tsai, R. Y. A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses. *IEEE Journal of Robotics and Automation RA-3*, 4 (Aug. 1987), 323–344.

[23] Zucchelli, M., and Christensen, H. L. A comparison of stereo based and flow based structure from parallax. In *Eighth International Symposium on Intelligent Robotic Systems (SIRS2000)* (July 2000).