A Space-Sweep Approach to True Multi-Image Matching *

Robert T. Collins
Department of Computer Science
University of Massachusetts
Amherst, MA. 01003-4610
Email: rcollins@cs.umass.edu
URL: http://vis–www.cs.umass.edu/~rcollins/home.html

Abstract

The problem of determining feature correspondences across multiple views is considered. The term "true multi-image" matching is introduced to describe techniques that make full and efficient use of the geometric relationships between multiple images and the scene. A true multi-image technique must generalize to any number of images, be of linear algorithmic complexity in the number of images, and use all the images in an equal manner. A new space-sweep approach to true multi-image matching is presented that simultaneously determines 2D feature correspondences and the 3D positions of feature points in the scene. The method is illustrated on a seven-image matching example from the aerial image domain.

1 Introduction

This paper considers the problem of **multi-image stereo reconstruction**, namely the recovery of static 3D scene structure from multiple, overlapping images taken by perspective cameras with known extrinsic (pose) and intrinsic (lens) parameters. The dominant paradigm is to first determine corresponding 2D image features across the views, followed by triangulation to obtain a precise estimate of 3D feature location and shape. The first step, solving for matching features across multiple views, is by far the most difficult. Unlike motion sequences, which exhibit a rich set of constraints that lead to efficient matching techniques based on tracking, determining feature correspondences from a set of widely-spaced views is a challenging problem. However, even disparate views contain underlying geometric relationships that constrain which 2D image features might be the projections of the same 3D feature in the world. The purpose of this paper is to explore what it means to make full and efficient use of the geometric relationships between multiple images and the scene.

2 True Multi-Image Matching

This paper presents, for the first time, a set of conditions that a stereo matching technique should meet to be called a "true multi-image" method. By this we mean that the technique truly operates in a multi-image manner, and is not just a repeated application of two- or three-camera techniques.

Definition: A *true multi-image* matching technique satisfies the following conditions:

1. the method generalizes to any number of images greater than 2,
2. the algorithmic complexity is $O(n)$ in the number of images, and
3. all images are treated equally (i.e. no image is given preferential treatment).

Condition 1 is almost a tautology, stating that a multi-image method should work for any number of images, not just two or three. An algorithm for processing three images is not a "multi-image" method, but rather a trinocular one. Condition 2 speaks directly to the issue of efficiency. To enable processing large numbers of images, the method used should be linear in the number of images. This condition precludes approaches that process all pairs of images, then fuse the results. Such an approach is not a multi-image method, but rather a repeated application of a binocular technique.

Condition 3 is the most important – it states that the information content from each image must be treated equally. Note that this is **not** intended to mean that information from all images must be equally weighted; some may be from better viewing positions, of higher resolution, or more in focus. Instead, condition 3 is meant to preclude singling out one image, or a subset of images, to receive a different algorithmic treatment than all the others. A common example is the selection of one image as a "reference" image. Features in that image are extracted, and then the other images in the dataset are searched for correspondence matches, typically using epipolar constraints between the reference image and each other image in turn. Although a popular approach, there is an inherent flaw in this style of processing – if an important feature is missing in the reference image due to misdetection or occlusion, it will not be present in the 3D reconstruc-

*Funded by the RADIUS project under ARPA/Army TEC contract number DACA76-92-C-0041.

tion even if it has been detected in all the other views, because the system won't know to look for it.

Although the conditions presented above are well-motivated and reasonable, there are hardly any stereo matching algorithms in the literature that meet all three. For example, Okutomi and Kanade describe a **multi-baseline stereo** method for producing a dense depth map from multiple images by performing two-image stereo matching on all pairs of images and combining the results [10]. Although they show convincingly that integrating information from multiple images is effective in reducing matching ambiguity, using all pairs of images makes this an $O(n^2)$ algorithm that violates condition 2 of the true multi-image definition. The basic multi-baseline system design was later transfered to hardware, and the control strategy changed to combining two-image stereo results between a "base" view and all other views [8]. This yields an $O(n)$ method rather than $O(n^2)$, however the implementation now violates condition 3, since one image is given special importance as a reference view. Any areas of the scene that are occluded in that image can not be reconstructed using this method.

Gruen and Baltsavias describe a **constrained multiphoto matching** system where intensity templates extracted from one reference image are affine-warped and correlated along epipolar lines in each other image [5]. Kumar et.al. present a multi-image **plane+parallax matching** approach where they compensate for the appearance of a known 3D surface between a reference view and each other view, then search for corresponding points along lines of residual parallax [9]. In both cases, special reference views have been chosen, and the algorithms essentially just apply a two-image matching technique repeatedly to pairs of images containing the reference view.

The reason why so many approaches attempt to solve the multi-image matching problem by splitting the set into pairs of images that are processed binocularly is because matching constraints based on the epipolar geometry of two views are so powerful and well-known. What is needed for simultaneous matching of features across multiple images is to generalize two-image epipolar relations to some **multilinear relation** between the views. For example, Shashua presents a "trilinear" constraint [12] where points in three images can be the projections of a single 3D scene point if and only if an algebraic function vanishes. Hartley devised a similar constraint for lines in three views [7]. A recent paper by Triggs [13] provides a framework in which all projective multilinear relationships can be enumerated: the binocular epipolar relationship, Shashua's trilinear relationship for points, Hartley's trilinear relationship for lines, and a quadrilinear relation for points in four views. The number of views is limited to four since the projective coordinates of 3D space have only four components. This violates condition 1 of the definition of a true multi-image method, and calls into question whether any approach that operates purely in image space can be a true multi-image method.

In contrast to the strictly image-space approaches above, some photogrammetric object-space approaches **do** fit the definition of a true multi-image method. Helava presents a typical example of **object-space least-squares matching** where correspondences between multiple images are determined by backprojecting image features onto some surface in the world and performing the correspondence matching in object space [6]. Another example is the work of Fua and Leclerc, who describe an approach for object reconstruction via **image energy minimization**, where 3D surface mesh representations are directly reconstructed from multiple intensity images [3]. Loosely speaking, triangular surface elements are adjusted so that their projected appearance in all the images is as similar as possible to the observed image intensities, while still maintaining a consistent shape in object-space.

One thing that the true multi-image matching/reconstruction methods above have in common is the explicit reconstruction of a surface or features in object space, simultaneous with the determination of image correspondences. In this way, object-space becomes the medium by which information from multiple images is combined in an even-handed manner. Unfortunately, the two object space approaches mentioned here involve setting up huge optimization problems with a large number of parameters, and initial estimates of scene structure are needed to reliably reach convergence. We present a much more efficient approach in the next section.

3 An Efficient Space-Sweep Approach

This section presents a true multi-image matching algorithm that simultaneously determines the image correspondences and 3D scene locations of point-like features (e.g. corners, edgels) across multiple views. The method is based on the premise that areas of space where several image feature viewing rays (nearly) intersect are likely to be the 3D locations of observed scene features. A naive implementation of this idea would partition a volume of space into voxels, backproject each image point out as a ray through this volume, and record how many rays pass through each voxel. The main drawback of this implementation would be its intensive use of storage space, particularly when partitioning the area of interest very finely to achieve accurate localization of 3D features.

3.1 The Space-Sweep Method

We propose to organize the computation as a space-sweep algorithm. A single plane partitioned into cells is swept through the volume of space along a line perpendicular to the plane. Without loss of generality, assume the plane is swept along the Z-axis of the scene, so that the plane equation at any particular point along the sweep has the form $Z = z_i$ (see Figure 1). At each position of the plane along the sweeping path, the number of viewing rays that intersect each cell are tallied. This is done by backprojecting point features from each image onto the sweeping plane (in a manner described in Section 3.2), and incrementing cells whose centers fall within some radius of the backprojected point position (as described in Section 3.3).

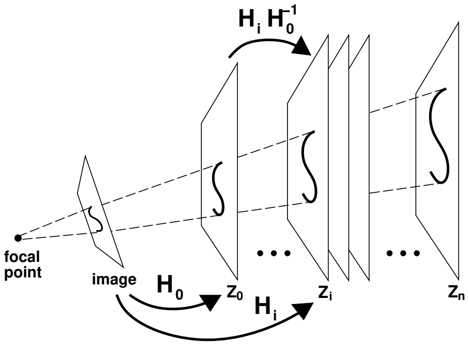

Figure 1: Illustration of the space-sweep method. Features from each image are backprojected onto successive positions $Z = z_i$ of a plane sweeping through space.

After accumulating counts from feature points in all of the images, cells containing counts that are "large enough" (Section 3.3) are hypothesized as the locations of 3D scene features. The plane then continues its sweep to the next Z location, all cell counts are reset to zero, and the procedure repeats. For any feature location (x, y, z_i) output by this procedure, the set of corresponding 2D point features across multiple images is trivially determined as consisting of those features that backproject to cell (x, y) within the plane $Z = z_i$.

Two implementation issues are addressed in the remainder of this section, namely how to efficiently determine where viewing rays intersect the sweeping plane, and how to decide whether a given number of ray intersections is statistically meaningful, or could instead have occurred by chance. We note in passing a method developed by Seitz and Dyer that, while substantially different from the approach here, is based on the same basic premise of determining positions in space where several viewing rays intersect [11]. However, because feature evidence is combined by geometric intersection of rays, only the correspondences and 3D structure of features detected in EVERY image are found – a severe limitation.

3.2 Efficient Backprojection

Recall that features in each image are backprojected onto each position $Z = z_i$ of the sweeping plane. For a perspective camera model, the transformation that backprojects features from an image onto the plane $Z = z_i$ is a nonlinear planar homography represented by the 3×3 matrix:

$$\boldsymbol{H}_i \;=\; \boldsymbol{A}\; [\boldsymbol{r}_1 \;\; \boldsymbol{r}_2 \;\; z_i\boldsymbol{r}_3 + \boldsymbol{t}],$$

where $\boldsymbol{A}$ is the 3×3 matrix describing the camera lens parameters, and the camera pose is composed of a translation vector $\boldsymbol{t}$ and an orthonormal rotation matrix with column vectors $\boldsymbol{r}_i$. This section shows that it is more efficient to compute feature locations in the plane $Z = z_i$ by modifying their locations in some other plane $Z = z_0$ to take into account a change in Z value, than it is to apply the homography $\boldsymbol{H}_i$ to the original image plane features.

Let matrix $\boldsymbol{H_0}$ be the homography that maps image points onto the sweeping plane at some canonical position $Z = z_0$. Since homographies are invertible and closed under composition, it follows that the homography that maps features between the plane $Z = z_0$ and $Z = z_i$ directly, by first (forward) projecting them from the z_0-plane onto the image, then backprojecting them to the z_i-plane, can be written as $\boldsymbol{H}_i\boldsymbol{H}_0^{-1}$ (refer to Figure 1).

It can be shown that the homography $\boldsymbol{H}_i\boldsymbol{H}_0^{-1}$ has a very simple structure [2]. In fact, if (x_0, y_0) and (x_i, y_i) are corresponding backprojected locations of a feature point onto the two positions of the sweeping plane, then

$$\begin{aligned} x_i &= \delta\, x_0 \;+\; (1-\delta)C_x \\ y_i &= \delta\, y_0 \;+\; (1-\delta)C_y \end{aligned} \qquad (1)$$

where $\delta = (z_i - C_z)/(z_0 - C_z)$ and $(C_x, C_y, C_z) = (-\boldsymbol{r_1} \cdot \boldsymbol{t}, -\boldsymbol{r_2} \cdot \boldsymbol{t}, -\boldsymbol{r_3} \cdot \boldsymbol{t})$ is the location of the camera focal point in 3D scene coordinates. A transformation of this form is known as a *dilation.*[1] The trajectories of all points are straight lines passing through the fixed point (C_x, C_y), which is the perpendicular projection of the camera focal point onto the sweeping plane (see Figure 2). The effect of the dilation is an isotropic scaling about point (C_x, C_y). All orientations and angles are preserved.

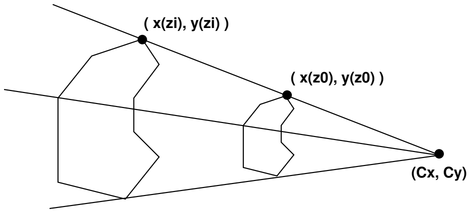

Figure 2: Transformation $H_i H_0^{-1}$ is a dilation that maps points along trajectories defined by straight lines passing through the fixed point (C_x, C_y).

Our strategy for efficient feature mapping onto different positions of the sweeping plane is to first perform a single projective transformation of feature points from each image $I_j, j = 1, ..., n$ onto some canonical plane $Z = z_0$. These backprojected point positions are not discretized into cells, but instead are represented as full precision (X,Y) point locations. For any sweeping plane position $Z = z_i$, each of these (X,Y) locations is mapped into the array of cells within that plane using formula (1), taking care to use the correct camera center $(C_x, C_y, C_z)_j$ for the features from image I_j.

3.3 A Statistical Model of Clutter

This section sketches an approximate statistical model of clutter that tells how likely it is for a set of viewing rays to coincide by chance (more details are given

[1] This is unrelated to the morphological dilation operator.

in [2]). Determining the expected number of votes each cell in the sweeping plane receives is simplified considerably by assuming that extracted point features are roughly uniformly distributed in each image. This is manifestly untrue, of course, since image features exhibit a regularity that arises from the underlying scene structure. Nonetheless, they will be uniform enough for the purpose of this discussion as long as a $k \times k$ block of pixels in the image contains roughly the same number of features as any other $k \times k$ block. Under this assumption, let the density of point features in image i be $E_i << 1$ (computed empirically). The expected number of features that image i projects into the sweeping plane is then this expected number of features per pixel E_i times the number of pixels O_i that have viewing rays passing through some cell in the sweeping plane.

Recall that each point feature in image i is allowed to vote for a set of cells surrounding the intersection of its viewing ray with the sweeping plane. Votes are given to the set of cells roughly contained in the region subtended by a pixel-shaped cone of viewing rays emanating from the point feature in image i. Pixels from images farther away from the sweeping plane thus contribute votes to more cells than pixels from images that are closer. This mechanism automatically accounts for the fact that scene feature locations are localized more finely by close-up images than by images taken from far away.

The number of cells in the sweeping plane that a pixel in image i votes for is thus specified by the Jacobian J_i of the projective transformation from image i onto the sweeping plane. We make a second simplifying assumption that this Jacobian is roughly constant, which is equivalent to assuming that the camera projection equations are approximately affine over the volume of interest in the scene. The total expected number of votes that image i contributes to the sweeping plane is thus estimated as the number of features mapped to the plane, times the number of cells that each feature votes for, that is $E_i * O_i * J_i$. Dividing this by the number of accumulator cells in the sweeping plane yields the probability θ_i that any cell in the sweeping plane will get a vote from image i.

For each accumulator cell, the process of receiving a vote from image i is modeled as a Bernoulli random variable with probability of success (receiving a vote) equal to θ_i. The total number of votes V in any sweeping plane cell is simply the sum of the votes it receives from all n images. Thus V is a sum of n Bernoulli random variables with probabilities of success $\theta_1, \ldots, \theta_n$. Its probability distribution function $D[V]$ tells, for any possible number of votes $V = 0, 1, ..., n$, what the probability is that V votes could have arisen by chance. In other words, $D[V]$ specifies how likely is it that V backprojected feature rays could have passed through that cell due purely to accidental alignments.

Once the clutter distribution function $D[V]$ is known, a solid foundation exists for evaluating decision rules that determine which sweeping plane cells are likely to contain scene features based on the evidence provided by backprojected image feature rays. A simple decision rule compares the number of votes V in each cell against a global threshold T, declaring that location to contain a feature when $V \geq T$. For each potential threshold $T \in \{1, ..., n\}$, the false positive rate $F[T]$ of this decision rule is easily computed as $F[T] = \sum_{i=j}^{n} D[i]$. A threshold T can then be chosen based on how certain we want the results to be.

4 Experimental Example

This section presents an in-depth example of the space-sweep algorithm for multi-image matching using aerial imagery from the RADIUS project [4]. Seven images of Fort Hood, Texas were cropped to enclose two buildings and the terrain immediately surrounding them. The images exhibit a range of views and resolutions (see Figure 3). The point features used are edgels detected by the Canny edge operator [1]. Figure 4 shows a binary edge image extracted from one of the views. Structural features of particular interest are the building rooftops and the network of walkways between the buildings. Note the significant amount of clutter due to trees in the scene, and a row of parked cars at the bottom.

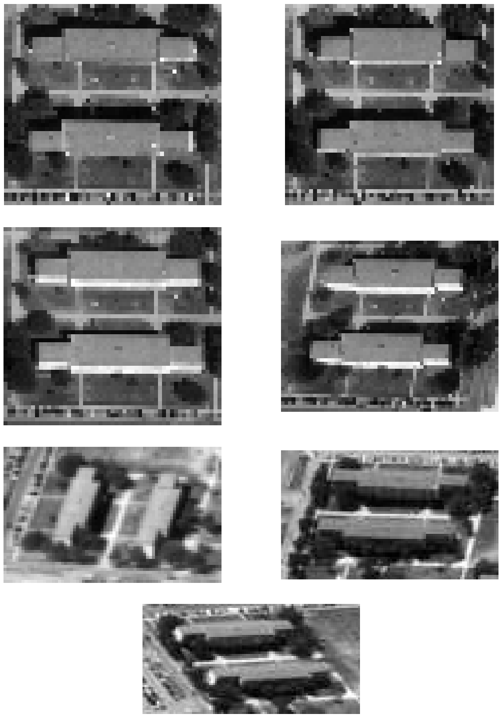

Figure 3: Seven aerial subimages of two buildings.

Reconstruction was carried out in a volume of space with dimensions 136 × 130 × 30 meters. A horizontal sweeping plane was swept through this volume along the Z-axis. Each accumulator cell on the plane was 1/3 meter square, a size chosen to roughly match the resolution of the highest resolution image. Viewing ray intersections were sampled at 100 equally-spaced locations along the sweeping path, yielding approx-

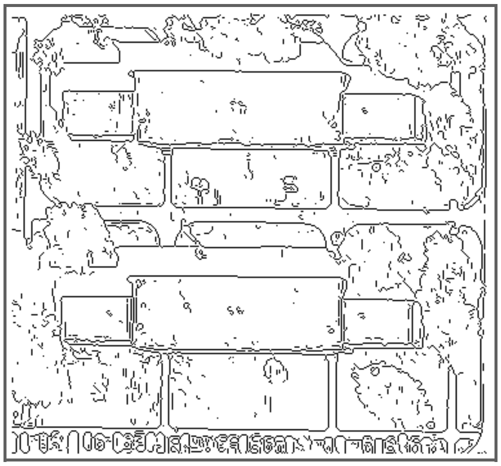

Figure 4: Canny edges extracted from one image.

imately a 1/3-meter resolution in the vertical direction as well. Figure 5 shows three sample plane locations along the sweeping path, chosen to illustrate the state of the sweeping plane when it is coincident with ground-level features (a), roof-level features (c) and when there is no significant scene structure (b). Also shown are the results of thresholding the sweeping plane at these levels, displaying only those cells with five or more viewing rays passing through them.

The approximate statistical model of clutter presented in Section 3.3 needs to be validated, since it is based on two simplifying assumptions, namely that edgels in the each image are distributed uniformly, and that the Jacobian of the backprojection from each image to the sweeping plane is roughly constant. This was done by comparing the theoretical clutter probability distribution $D[V], V = 0, 1, ..., 7$ against the empirical distributions of feature votes collected in each of the 100 sweeping plane positions. Recall that the clutter distribution $D[V]$ tells how many ray intersections are likely to pass through each accumulator cell purely by chance. This theoretical distribution should match the empirical distribution well for sweeping plane positions where there is no significant 3D scene structure. The chi-square statistic was used to measure how similar these two discrete distributions are for each Z-position of the sweeping plane; the results are plotted in Figure 6. Lower values mean good agreement between the two distributions, higher values mean they are not very similar. Two prominant, sharp peaks can be seen, implying that the dominant 3D structure of this scene lies in two well-defined horizontal planes, in this case ground-level features and building rooftops. More importantly, the plot is very flat for Z-levels that contain no significant scene structure, showing that the theoretical clutter model is actually a very good approximation to the actual clutter distribution.

Recall that once the clutter distribution $D[V]$ is computed for any Z-position of the sweeping plane, a vote threshold $T = 1, ..., n$ for classifying which cells contain 3D scene features can be chosen taking into account the expected false positive rate $F[T]$. The false positive rates computed for this dataset are very consistent across all Z positions of the sweeping plane. A representative sample is:

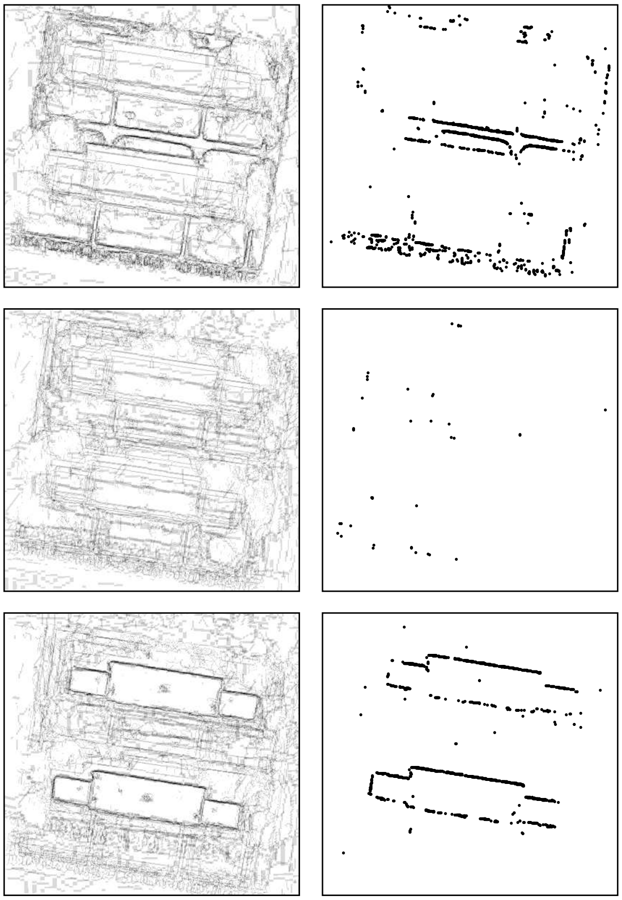

Figure 5: Three sample Z-positions of the sweeping plane. Left shows votes in the sweeping plane; right the results of feature classification using a threshold value of 5.

T	1	2	3	4	5	6	7
100 F[T]	88.4	59.0	27.3	8.3	1.6	0.17	0.01

This table displays for any given choice of threshold T, what the percentage of false positives would be if cells with votes of T or higher are classified as the locations of 3D scene features.

A desired confidence level of 99% was chosen for recovered 3D scene features, implying that we are willing to tolerate only 1% false positives due to clutter. Based on this choice and the above table, the optimal threshold should be between 5 and 6, but closer to the former. Figure 7 graphically compares extracted 3D ground features and roof features using these two different threshold values. Choosing an optimal threshold is a balancing act; ultimately, the proper tradeoff between structure and clutter needs to determined by the application.

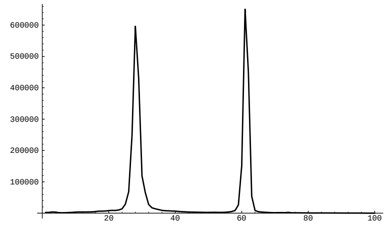

Figure 6: Comparison of theoretical and empirical clutter distributions at each sweeping plane position (see text).

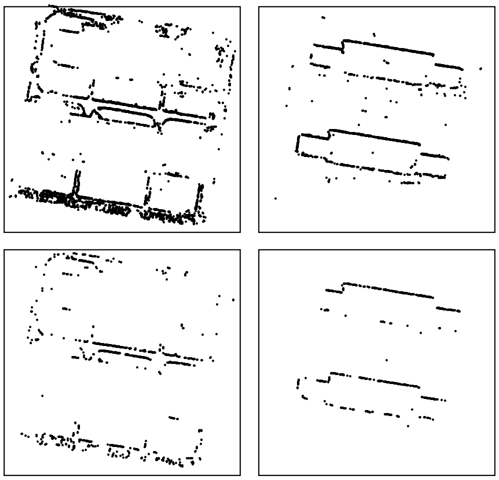

Figure 7: XY locations of detected scene features for a range of Z-values containing ground features (left) and roof features (right). Results from two different threshold values of 5 (top) and 6 (bottom) are compared.

5 Summary and Extensions

This paper defines the term "true multi-image" matching to formalize what it means to make full and efficient use of the geometric relationships between multiple images and the scene. Three conditions are placed on a true multi-image method: it should generalize to any number of images, the algorithmic complexity should be linear in the number of images, and every image should be treated on an equal footing, with no one image singled out for special treatment as a reference view. A new space-sweep algorithm for true multi-image matching is presented that simultaneously determines 2D feature correspondences between multiple images and the 3D positions of feature points in the scene. It is shown that the intersections of viewing rays with a plane sweeping through space can be determined very efficiently. A statistical model of feature clutter is developed to tell how likely it is that a given number of viewing rays pass through some area of the sweeping plane by chance, thus enabling a principled choice of threshold to be chosen for determining whether or not a 3D feature is present. The approach is illustrated using a seven-image matching example from the aerial image domain. Several extensions to this basic approach are being considered. One is the development of a more sophisticated model of clutter that adapts to the spatial distribution of feature points in each image. The second extension is to consider the gradient orientations of potentially corresponding edgel features; when accumulating feature votes in a sweeping plane cell, only edgels with compatible orientations should be added together. With the introduction of orientation information, detected 3D edgels could begin to be linked together in the scene to form 3D chains, leading to the detection and fitting of symbolic 3D curves.

References

[1] J.Canny, "A Computational Approach to Edge Detection," *IEEE Pattern Analysis and Machine Intelligence,* Vol. 8(6), 1986, pp. 679–698.

[2] R.Collins, "A Space-Sweep Approach to True Multi-Image Matching," Technical Report 95-101, Computer Science Department, UMass, December 1995.

[3] P.Fua and Y.Leclerc, "Object-centered Surface Reconstruction: Combining Multi-Image Stereo and Shading," *IJCV,* Vol. 16(1), 1995, pp. 35–56.

[4] D.Gerson, "RADIUS : The Government Viewpoint," *Proc. ARPA Image Understanding Workshop,* San Diego, CA, January 1992, pp. 173–175.

[5] A.Gruen and E.Baltsavias, "Geometrically Constrained Multiphoto Matching," *Photogrammetric Engineering and Remote Sensing,* Vol. 54(5), 1988, pp. 633–641.

[6] U.Helava, "Object-Space Least-Squares Correlation," *Photogrammetric Engineering and Remote Sensing,* Vol. 54(6), 1988, pp. 711–714.

[7] R.Hartley, "Lines and Points in Three Views – an Integrated Approach," *Proc. ARPA Image Understanding Workshop,* Monterey, CA, 1994, pp. 1009–1016.

[8] T.Kanade, "Development of a Video-Rate Stereo Machine," *Proc. Arpa Image Understanding Workshop,* Monterey, CA, Nov 1994, pp.549–557.

[9] R.Kumar, P.Anandan and K.Hanna, "Shape Recovery from Multiple Views: A Parallax Based Approach," *Arpa IUW,* Monterey, CA, Nov 1994, pp.947–955.

[10] M.Okutomi and T.Kanade, "A Multiple-Baseline Stereo," *IEEE Pattern Analysis and Machine Intelligence,* Vol. 15(4), April 1993, pp. 353–363.

[11] S.Seitz and C.Dyer, "Complete Scene Structure from Four Point Correspondences," *Proc. International Conference on Computer Vision,* Cambridge, MA, June 1995, pp. 330–337.

[12] A.Shashua, "Trilinearity in Visual Recognition by Alignment," *Proc. European Conference on Computer Vision,* Springer-Verlag, 1994, pp. 479–484.

[13] B.Triggs, "Matching Constraints and the Joint Image," *Proc. International Conference on Computer Vision,* Cambridge, MA, June 1995, pp. 338–343.