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

Robert T. Collins

**CMPSCI Technical Report 95-101**
December, 1995

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

Robert T. Collins

CMPSCI TR95-101

December 1995

* This work was funded by the RADIUS project under ARPA/Army TEC contract number DACA76-92-C-0041.

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

Robert T. Collins

Department of Computer Science
Lederle Graduate Research Center
Box 34610, University of Massachusetts
Amherst, MA. 01003-4610
rcollins@cs.umass.edu

**Keywords:** multi-image stereo, multi-image correspondence matching, scene reconstruction

## 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 based on the premise that areas of space where several viewing rays intersect are the likely locations of observed 3D scene features. It is shown that the intersections of viewing rays with a plane sweeping through space can be determined very efficiently, and a statistical model is developed to tell how likely it is that a given number of viewing rays will pass through an area of the plane by chance. The method is illustrated on a seven-image matching example from the aerial image domain.

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

# 1 Introduction

Due to advances in computer processing speed, coupled with larger and cheaper storage devices, computer vision methods that process multiple images are more feasible than ever. Indeed, scene reconstruction from a set of multiple views is currently of great interest in the field.

This paper considers the problem of **multi-image stereo reconstruction**, namely the recovery of static 3D scene structure from an unordered set of 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.

In Section 2, a set of conditions is presented that must hold for a matching algorithm to be called a "true multi-image" method. Briefly, we claim that a true multi-image matching technique should be applicable to any number of images $n \geq 2$, that the algorithmic complexity should be $O(n)$ in the number of images, and that the images should all be treated in an equal manner. Examples from the literature are presented to illustrate the meaning and motivation for each condition. The few positive examples that are found are examined for characteristics that they have in common.

In Section 3 we present a new approach to true multi-image matching that simultaneously determines 2D feature correspondences between multiple images and the positions of the ob-

served 3D features in the scene. The method can be visualized as sweeping a plane through space, while noting positions on it where many backprojected feature rays intersect.[1] A careful examination of the projective relationships between the images and the plane in space, and between different positions of the sweeping plane, shows that the feature mappings involved can be performed very efficiently. A statistical model is also developed to help decide how likely it is that the results of the matching procedure are correct.

Section 4 of this paper shows an illustrative example of the space-sweep method as applied to imagery from the RADIUS aerial image understanding project. The paper concludes with a brief summary and a discussion of extensions and improvements to the approach.

# 2 True Multi-Image Matching

## 2.1 What is a "True Multi-Image" Method?

This section presents, for the first time, a set of conditions that a stereo matching technique has to meet before it can be called a "true multi-image" method. By this we mean that the technique exploits the information present in a set of images in an effective and efficient manner. The use of the adjective "true" is not meant to denigrate methods that do not satisfy this set of conditions, but rather denotes that a method that does pass the test 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 special status as a "reference" image).

Condition 1 is almost a tautology, stating that a multi-image method should work for any number of images, not just two or three. Although the term "multiple" arguably does apply to

[1] Although we are tempted to call this a "plane-sweep" approach, since the geometric entity doing the sweeping is a plane, we defer to the computational geometry literature where the term plane-sweep denotes sweeping a line through a plane, and space-sweep refers to sweeping a plane through space [15].

the numbers 2 and 3, it also applies to the numbers 5, 10, and 20. 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. If one is serious about 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 subtle (and perhaps controversial). It states that the information content from each image must be treated as equally important. Note that this is **not** intended to mean that information from all images must be equally weighted in the final reconstruction; all images are not created equal, and 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 the images, as somehow being special, and deserving of 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 reconstruction even if it has been detected in all the other views, because the system won't know to even look for it in the other views.

We hope the reader agrees that the three conditions presented above seem well-motivated and reasonable. However, one is hard-pressed to find stereo reconstruction algorithms in the literature that meet all three conditions! We first present some negative examples from the computer vision literature - the aim is not a complete literature review, but rather an illustrative sampling of the range of different methods that have been proposed. We then show some of the

positive examples that were found, and examine them to extract characteristics they have in common.

## 2.2 Examples from the Literature

### 2.2.1 Multi-Baseline Stereo

Okutomi and Kanade developed a method called multi-baseline stereo for producing a dense depth map from multiple images taken by a coplanar set of cameras with parallel optical axes [14]. It is a generalization of two-image, correlation-based stereo where the correspondence of each pixel in the first image is found by sliding a correlation window over an epipolar line in the second image, and declaring a match at the pixel where a sum of squares difference (SSD) function is minimized. Okumomi and Kanade compute a SSD function with respect to inverse distance (rather than disparity) for all pairs of images, then combine them by adding to produce a sum of SSD (or SSSD) function. The minimum point of the SSSD function identifies the inverse distance to the point in the scene. They show convincingly that integrating information from multiple images reduces the inherent matching ambiguity that exists when only two images are used. Using all pairs of images makes this an $O(n^2)$ algorithm, however, and violates condition 2 of the true multi-image definition.

The basic multi-baseline system design was later transfered to hardware, resulting in a fast, real-time stereo machine [11, 22]. Rather than combining SSD functions from all pairs of views, these implementations combine SSD functions computed between a "base" view and all other views. A similar approach was developed in software by Tsai [21]. This processing strategy yields an $O(n)$ method rather than $O(n^2)$ (and the added efficiency is no doubt important for making a system that runs in real-time), however these implementations now violate condition 3, since one image is given special importance as a base or reference view. Any areas of the scene that are occluded in that image can not be reconstructed using this method.

### 2.2.2 Constrained Multiphoto Matching

Gruen and Baltsavias describe another correlation-based system for determining correspondences across multiple views [7], but the camera stations are allowed to be in general position. An intensity template extracted from a reference image is searched for along epipolar lines in a set of remaining views. To take into account wide differences in camera viewing angle, affine warping of the correlation template is performed for each image, and the the position of intensity templates in each image are constrained to move by fixed "step ratios" along an epipolar line in order to guarantee that all template positions correspond to a single point in space. Once again, however, condition 3 has been violated by choosing templates from a special reference image.

### 2.2.3 Plane + Parallax Approaches

An interesting result from projective geometry is that after compensating for the difference in appearance between two images of four points on a planar surface, other feature points on the plane have zero-disparity, and can thus be trivially matched. Points that do not lie in the plane have disparity vectors that lie along lines converging to a single point in the image, with lengths related to the distance of each point to the plane. Kumar et.al. describe a multi-image extension of the basic plane+parallax approach[12]. They compensate for the appearance of a known 3D surface between a reference view and each other view, then search for corresponding points along parallax lines, and compute 3D scene structure using the recovered parallax. Once again, a special reference view has been chosen, and the approach is basically that of repeatly applying a two-image method to pairs of images that containing the reference view.

### 2.2.4 Multilinear Constraints

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. Given a

point $\boldsymbol{p}$ in one image, its corresponding match $\boldsymbol{p}'$ in a second view is constrained to lie along an epipolar line that passes through a single point in that image called the epipole. The epipole is literally a picture of the focal point of the first camera as seen by the second, and the epipolar line is how the infinite viewing ray associated with $\boldsymbol{p}$ appears in the second image. The epipolar constraint is enormously powerful, since it reduces the search for a corresponding features from two dimensions down to one.

What is needed for simultaneous matching of features across multiple images is to generalize the two-image epipolar geometry to some multilinear relationship between the views. In this respect, results from the study of projective geometry at first appear to be promising. Shashua presents a "trilinear" constraint [19] in which points $\boldsymbol{p}$, $\boldsymbol{p}'$ and $\boldsymbol{p}''$ in three images can be the projections of a single 3D scene point if and only if an algebraic function vanishes, that is $f(\boldsymbol{p}, \boldsymbol{p}', \boldsymbol{p}'') = 0$. Hartley devised a similar constraint for lines in three views [9].

Can multilinear constraints be devised for any number of images? Unfortunately, the answer is no. A recent paper by Triggs [20] provides a framework for studying multilinear relationships between $m$ projective images by considering an abstract *joint image space* formed by concatenating the homogeneous coordinates of the set of images together. Within this framework, all possible 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 homogeneous coordinates of 3D space have only four components. Thus, hypothetical correspondence matching approaches based on multilinear relationships are strictly limited to four or fewer images. This violates condition 1 of the definition of a true multi-image method, namely that the method should generalize to any number of images. This result also calls into question whether any approach that operates purely in image space can ever be a true multi-image method.

## 2.3 Positive Examples

### 2.3.1 Object-Space Least Squares Matching

In contrast to the correlation and feature matching approaches outlined above, which can be considered strictly image-level approaches, most recent photogrammetric applications favor an object-space approach where correspondences between multiple images are determined by back-projecting image features into some surface in the world and performing correspondence matching in object space. These methods are primarily used to generate digital terrain maps (DTM), and correspondences correlations are determined via least-squares adjustment while simultaneously estimating the surface topography and radiometry of the terrain.

Helava presents a typical example of this kind of system [8], where a grid of ground elements or "groundels" in the scene is estimated along with the precise correspondence of intensity patches appearing in multiple images. An iterative least-squares procedure is used to simultaneously estimate groundel elevation and intensity parameters. The least-squares residuals are terms of the form $G_i - T_j(I_j, Z_i)$ where $G_i$ and $Z_i$ are the unknown grey-value and elevation of groundel $i$, and $T_j$ is the transformation function that determines how image intensities from the $j$-th image $I_j$ backproject onto the $i$-th groundel based on the current estimate of its elevation $Z_i$. The objective function to minimize is formed as the sum of squares of terms of this form, summed up over all images and all groundels. Similar systems are described in [3, 17].

Although this least-squares approach potentially involves solving for a huge number of parameters (DTM grid sizes of $500 \times 500$ are not uncommon), it does meet all three conditions for a true multi-image method. It generalizes to any number of images, the algorithm is linear in the number of images (although the run-time will typically be dominated by the number of groundels that have to be estimated), and most importantly, information from all of the images is treated on an equal footing.

### 2.3.2 Object-Centered Reconstruction via Image Energy Minimization

Fua and Leclerc describe an approach that combines reconstruction in object-space within a framework of image energy minimization [5]. Object-centered triangulated mesh representations of surfaces in the scene are directly reconstructed from multiple intensity images. An objective function is formed that contains energy terms based on image intensity information and object-level shape constraints. The surface shape is optimized iteratively by adding an adjustable regularization term and minimizing the total energy using a heuristic continuation method. Loosely speaking, the 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. This work also fits the definition of a multi-image method.

### 2.3.3 The Utility of 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. This does not mean that a matching technique that operates only in image space can not be a true multi-image method (however, see the comment about the limitations of multilinear matching constraints), but to date we know of none.

Unfortunately, the two object space approaches mentioned here involve setting up huge optimization problems with a large number of parameters. Initial estimates of scene structure are needed to reliably reach convergence. We present a much simpler and efficient approach in the next section that is suitable for matching point-like features across multiple images.

# 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 could be achieved by partitioning a volume of space into voxels, backprojecting each image point out as a ray through this volume, and recording how many rays pass through each voxel. Voxels with large numbers of viewing rays passing through them would be output as a set of likely 3D feature locations. Each detected 3D location could then be projected back into each of the images to trivially determine the locations of 2D image features in correspondence with it (and with each other).

The main drawback of this implementation would be its intensive use of storage space, needed to maintain a complete set of voxels, particularly when partitioning the area of interest very finely in order to achieve accurate localization of 3D features. In Section 3.1 we present a different approach where a single plane partitioned into cells is swept through the volume of space along a line perpendicular to the plane. At each position of the plane along the sweeping path, the number of viewing rays that intersect each cell are tallied, and any cell with sufficient numbers of intersections is output as the likely $(x, y, z)$ location of a 3D scene point. The plane then moves on.

Organizing the computation as a space-sweep algorithm not only saves a great deal of storage space, it also leads to a very efficient algorithm in terms of time. The operation of determining where viewing rays intersect the sweeping plane is crucial to the efficiency of the proposed algorithm. We show in Section 3.2 that ray intersections with the plane at any position along the sweep can be computed from the ray intersections at any other position of the plane along the sweep, by applying a simple dilation transformation. A second important issue that has to

be addressed is how many viewing rays need to intersect a cell in the sweeping plane before it is considered to be the likely location of a 3D scene feature. Section 3.3 develops a statistical model to help 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 [18]. They assume a uncalibrated, affine camera model, and first rectify each image using an affine plane+parallax approach. The affine subspace projecting to each image feature is explicitly constructed, and feature evidence is combined by intersecting these subspaces to localize the location of the observed 3D scene feature. Because evidence combination is performed via subspace intersections, only the correspondences and 3D structure of features detected in EVERY image are found – a severe limitation.

## 3.1 The Sweeping Plane

Our proposed method can be visualized as sweeping a plane through space along a line normal 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 $Z = z_i$. A bounded region of interest within the plane is partitioned into a grid of cells. Again without loss of generality, assume that the grid is aligned with the $X - Y$ axes of the scene, and that each cell in the grid is indexed by the location $(x, y)$ of its center. The volume of interest in space is bounded by the two planes $Z = z_{\min}$ and $Z = z_{\max}$. This volume is sampled by the sweeping plane at a discrete number of equally spaced $Z$-intervals within the limits $z_{\min}$ to $z_{\max}$.

At each position along the sweep, every cell on the sweeping plane $Z = z_i$ accumulates a count of the number of viewing rays passing through it. Each point feature in image $I_j$ determines a viewing ray that intersects some cell in the plane. This cell is determined by backprojecting the feature onto the plane $Z = z_i$, and incrementing the counts in cells whose centers fall within

some radius of the backprojected point position (selection of this radius is done automatically, and is described in Section 3.3). The backprojection function that maps feature points from the image plane onto the sweeping plane is a planar projective transformation that is determined by the perspective camera lens and pose parameters for $I_j$, and the position $z_i$ of the sweeping plane along its path, as developed fully in Section 3.2.

After accumulating counts from feature points in all of the images, cells containing counts that are "large enough" (in a sense to be determined in Section 3.3) are considered to be the locations of 3D scene features, and their locations $(x, y, z_i)$ in space are output. 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 $Z = z_i$.

## 3.2 Efficiency Considerations

The operation of determining where features in each image backproject onto each position $Z = z_i$ of the sweeping plane is crucial to the efficiency of the proposed algorithm. For a central projection camera model, the transformation that backprojects features from an image plane onto the sweeping plane is a nonlinear, 2D planar homography, representable as a $3 \times 3$ matrix $\boldsymbol{H}_i$ in homogeneous coordinates. In this section we will see 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 $\boldsymbol{H}_0$ represent the planar homography that maps image points onto the sweeping plane at some canonical position $Z = z_0$, and matrix $\boldsymbol{H}_i$ represent the homography mapping image points onto the plane $Z = z_i$ (refer to Figure 1). Since homographies are invertible and closed under composition, it follows that we can write a 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. This projection-backprojection homography is represented by the matrix $\boldsymbol{H_i H_0^{-1}}$. We will see that the homography $\boldsymbol{H_i H_0^{-1}}$ has a very simple structure indeed.

A change in notation makes it easier to discuss corresponding feature locations across different positions of the sweeping plane. Consider the (X,Y) location of a backprojected feature point to be indexed by the Z location of the plane it occurs in, written as $(x(Z), y(Z))$. We now have

$$\begin{bmatrix} x(z_i) \\ y(z_i) \\ 1 \end{bmatrix} \sim \boldsymbol{H_i H_0^{-1}} \begin{bmatrix} x(z_0) \\ y(z_0) \\ 1 \end{bmatrix} \tag{1}$$

where the symbol $\sim$ stands for equality up to a scale factor. Written in this way, it becomes convenient to view the different (X,Y) locations of an image feature backprojected onto the set of Z-planes as a curve or trajectory, parameterized by Z, within a stationary sweeping plane. The planar homography $\boldsymbol{H_i H_0^{-1}}$ can thus be viewed as a transformation acting on this stationary plane, mapping points $(x(z_0), y(z_0))$ into their new locations $(x(z_i), y(z_i))$.

To be precise about the actions of transformation $\boldsymbol{H_i H_0^{-1}}$, consider a standard central projection camera model, described by the following projection equation relating the homogeneous coordinates of 3D points in the scene to their corresponding 2D points in the image:

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \sim \boldsymbol{A} \begin{bmatrix} \boldsymbol{r}_1 & \boldsymbol{r}_2 & \boldsymbol{r}_3 & \boldsymbol{t} \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \tag{2}$$

where $\boldsymbol{A}$ is a $3 \times 3$ matrix describing the camera lens parameters and photo-processing effects such as enlargement and cropping, and the pose of the camera is represented as a $3 \times 4$ matrix composed of a translation vector $\boldsymbol{t}$ and an orthonormal rotation matrix with column vectors $\boldsymbol{r}_i$. For later reference, the 3D location of the camera focal point in scene coordinates is determined

as

$$(C_x, C_y, C_z) \;=\; (-\boldsymbol{r}_1 \cdot \boldsymbol{t},\; -\boldsymbol{r}_2 \cdot \boldsymbol{t},\; -\boldsymbol{r}_3 \cdot \boldsymbol{t})\,. \tag{3}$$

Restricting attention to scene points lying on the plane $Z = z_0$, we can immediately simplify the 3D-to-2D projection equation (2) into the invertible 2D-to-2D homography that maps sweeping plane coordinates $x(z_0)$ and $y(z_0)$ into image plane coordinates $u$ and $v$:

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \;\sim\; \boldsymbol{A}\,[\boldsymbol{r}_1 \;\; \boldsymbol{r}_2 \;\; z_0\boldsymbol{r}_3 + \boldsymbol{t}] \begin{bmatrix} x(z_0) \\ y(z_0) \\ 1 \end{bmatrix}. \tag{4}$$

Note that this equation represents the *forward* projection that maps features from the plane $Z = z_0$ onto the image plane. The backprojection labeled $\boldsymbol{H_0}$ in Figure 1 is found by inverting this $3 \times 3$ homography matrix.

An equation similar to (4) can also be written for the plane $Z = z_i$, yielding an equation for the homography labeled $\boldsymbol{H_i}$ in Figure 1. We can now determine the homography $\boldsymbol{H_i H_0^{-1}}$ describing the direct mapping between $(x(z_0), y(z_0))$ and $(x(z_i), y(z_i))$ as

$$\begin{aligned} \boldsymbol{H_i H_0^{-1}} \;&=\; [\boldsymbol{r}_1 \;\; \boldsymbol{r}_2 \;\; z_i\boldsymbol{r}_3 + \boldsymbol{t}]^{-1}\,(\boldsymbol{A}^{-1}\boldsymbol{A})\,[\boldsymbol{r}_1 \;\; \boldsymbol{r}_2 \;\; z_0\boldsymbol{r}_3 + \boldsymbol{t}] \\ &\sim\; [\boldsymbol{r}_2 \times (z_i\boldsymbol{r}_3 + \boldsymbol{t}) \;\; (z_i\boldsymbol{r}_3 + \boldsymbol{t}) \times \boldsymbol{r}_1 \;\; \boldsymbol{r}_1 \times \boldsymbol{r}_2]^T \;\; [\boldsymbol{r}_1 \;\; \boldsymbol{r}_2 \;\; z_0\boldsymbol{r}_3 + \boldsymbol{t}] \\ &=\; \begin{bmatrix} z_i + \boldsymbol{r}_3 \cdot \boldsymbol{t} & 0 & (z_i - z_0)\boldsymbol{r}_1 \cdot \boldsymbol{t} \\ 0 & z_i + \boldsymbol{r}_3 \cdot \boldsymbol{t} & (z_i - z_0)\boldsymbol{r}_2 \cdot \boldsymbol{t} \\ 0 & 0 & z_0 + \boldsymbol{r}_3 \cdot \boldsymbol{t} \end{bmatrix} \end{aligned} \tag{5}$$

where we have used the $3 \times 3$ matrix identity

$$[\boldsymbol{a}\;\boldsymbol{b}\;\boldsymbol{c}]^{-1} \;\sim\; [(\boldsymbol{b} \times \boldsymbol{c}) \;\; (\boldsymbol{c} \times \boldsymbol{a}) \;\; (\boldsymbol{a} \times \boldsymbol{b})]^T$$

as well as identities related to orthonormal vectors, e.g. $\boldsymbol{r}_1 \cdot \boldsymbol{r}_1 = 1$, $\boldsymbol{r}_1 \cdot \boldsymbol{r}_2 = 0$, $\boldsymbol{r}_1 = \boldsymbol{r}_2 \times \boldsymbol{r}_3$, and the vector triple product ([10], Appendix A), e.g. $(\boldsymbol{r}_2 \times \boldsymbol{t}) \cdot \boldsymbol{r}_1 = (\boldsymbol{r}_1 \times \boldsymbol{r}_2) \cdot \boldsymbol{t} = \boldsymbol{r}_3 \cdot \boldsymbol{t}$.

Finally, recalling the identity from Eq. (3), the equations governing the trajectory from point $(x(z_0), y(z_0))$ to $(x(z_i), y(z_i))$ can be written as

$$\left.\begin{array}{rcl} x(z_i) & = & \delta\, x(z_0) \;+\; (1-\delta) C_x \\ y(z_i) & = & \delta\, y(z_0) \;+\; (1-\delta) C_y \end{array}\right\} \text{ where } \delta = (z_i - C_z)/(z_0 - C_z) \qquad (6)$$

This is a special affine transform known as a *dilation* [13]; the important point is that it is linear in inhomogeneous coordinates. 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 mapping is an isotropic scaling about point $(C_x, C_y)$. All orientations and angles are preserved.

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 $Z = z_0$, where $z_0$ can be chosen as the Z-position midway between $z_{\text{min}}$ and $z_{\text{max}}$, for example. 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 (6), 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

Recall that our technique is based on backprojecting feature rays onto a planar array of cells sweeping through space. At each Z-location of the plane, every accumulator cell counts the number of viewing rays passing through it to determine the X,Y locations where several rays intersect. In a sense, each backprojected feature "votes" for whether or not that cell is the location of a 3D scene feature. The more votes a cell accumulates, the more likely it is that an observable 3D scene feature is present in that location. In this section we develop a simple statistical model that describes how this likelihood increases as a function of the number of votes

in the cell. We can expect a certain number of viewing rays to pass through an accumulator cell purely by chance; our model allows us to quantify this expectation.

An approximate statistical model of clutter is developed that tells how likely a set of viewing rays could coincide by chance. The model will be used to choose a threshold on the number of votes needed before an accumulator cell will be considered to reliably contain a 3D scene feature. The term "clutter" as used here refers not only to spurious features among the images, but also to sets of correctly extracted features that just don't correspond to each other. Loosely speaking, any particular 3D point in the scene induces a correspondence involving only one pixel from each image, and for the purpose of deciding whether there is a scene feature located at that point in space, all other image pixels are clutter.

In order to determine the expected number of votes (viewing rays) a cell in the sweeping plane receives, two values must be estimated: how many points from each image are mapped to the sweeping plane, and how many cells each point votes for. Computation of the first quantity is simplified by assuming that extracted point features are roughly uniformly distributed in each image. This is manifestly untrue, since features in the image exhibit a regularity that arises from the underlying scene structure. In fact, point-like features such as edgels are obviously not uniformly distributed, but instead tend to spatially organize into chains. Nonetheless, these features will be uniform enough for our purposes as long as any $k \times k$ block of pixels in the image contains roughly the same number of features as any other $k \times k$ block. Under this assumption, the density of point features in image $i$ is

$$E_i = \frac{\text{number of features in image } i}{\text{number of pixels in image } i} \tag{7}$$

which represents the expected number of point features ($E_i << 1$) extracted per pixel in image $i$. This is multiplied by the number of pixels $O_i$ that have viewing rays that pass through any cell in the sweeping plane. $O_i$ is computed by projecting the boundary of the accumulator cell

array onto the virtual plane containing the boundary of image $i$, and computing the number of pixels in the intersection of the two polygons. The expected number of features that image $i$ projects into the sweeping plane is this number of pixels times the expected number of features per pixel, namely $E_i * O_i$.

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 specified by the Jacobian of the projective transformation from image $i$ onto the sweeping plane. We make a second simplifying assumption that the Jacobian of this projective mapping is roughly constant. This is equivalent to assuming that the camera projection equations are approximately affine over the portion of the scene that is of interest. Let the four corners $\{\boldsymbol{c}_k, k = 1, ..., 4\}$ of the rectangular boundary of image $i$ map to the sweeping plane as a quadrilateral with vertices $\boldsymbol{a}_k = \{\boldsymbol{H}^{-1}\boldsymbol{c}_k, k = 1, ..., 4\}$. The approximate Jacobian for the mapping from image $i$ onto the sweeping plane is computed as

$$J_i = \frac{\text{number of pixels in } \{\boldsymbol{c}_1, \boldsymbol{c}_2, \boldsymbol{c}_3, \boldsymbol{c}_4\}}{\text{number of cells in } \{\boldsymbol{a}_1, \boldsymbol{a}_2, \boldsymbol{a}_3, \boldsymbol{a}_4\}} \tag{8}$$

The expected number of votes that image $i$ contributes to the sweeping plane is estimated as the number of features mapped to the plane, times the number of cells that each feature votes for, that is

$$\text{votes from image } i \;\; = \;\; E_i * O_i * J_i \;\; . \tag{9}$$

The probability $\theta_i$ that any cell in the sweeping plane will get a vote from image $i$ is therefore

$$\theta_i = \frac{E_i * O_i * J_i}{\text{number of cells in the sweeping plane}} \quad . \tag{10}$$

For each cell, we model the process of receiving a vote from image $i$ as a Bernoulli random variable with probability of success (receiving a vote) equal to $\theta_i$. Let $x_i = \{0, 1\}$ be a Bernoulli random variable representing the number of votes a cell receives from the $i$-th image. Then the probability distribution of $x_i$ is [4]

$$P(x_i; \theta_i) = \theta_i^{x_i} (1 - \theta_i)^{1-x_i} = \begin{cases} (1 - \theta_i) & ; x_i = 0 \\ \theta_i & ; x_i = 1 \end{cases} \quad . \tag{11}$$

The total number of votes in any sweeping plane cell is simply the sum of the votes it receives from all images, namely $V = \sum_{i=1}^{n} x_i$. The distribution of $V$ is clearly that of a sum of $n$ Bernoulli random variables with probabilities of success $\theta_1, \ldots, \theta_n$. The range of $V$ is $0, 1, \ldots, n$. The expected value of $V$ is $E(V) = E(\sum x_i) = \sum E(x_i) = \sum \theta_i$. We are unaware of a closed-form function representing the distribution of a sum of Bernoulli random variables, but it is easily computed by the following pseudo-code fragment:

```
; Compute the distribution function D[k], k = 0,1,...,n for the sum
; of n Bernoulli random variables with probabilities of success θ_i.
Let D[k] = 0, k = 0,1,...,n
For each bitstring B between 0 and 2^n - 1
    Set x_i, i = 1,...,n equal to the i-th bit of B (0 or 1)
    Compute the event probability Q = ∏ P(x_i, θ_i).
    Let k = ∑ x_i (number of 1 bits in bitstring B).
    Increment D[k] by Q.
```

The probability distribution function $D[k]$ tells, for any possibly number of votes $k = 0, 1, \ldots, n$ in a cell, what the probability is that $k$ votes could have arisen by chance. In other words, $D[k]$ specifies how likely is it that $k$ backprojected feature rays could have passed through that cell due purely to clutter or accidental alignments.

Once the clutter distribution function $D[k]$ is known, a solid foundation exists for evaluating decision rules that determine which sweeping plane cells are likely to contain a scene feature based on the evidence provided by backprojected image feature rays. A simple decision rule is to compare the number of votes $V$ in each cell against a global threshold $T$, and declaring that cell 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 wish the matching results to be. For example, if a value $T$ is chosen for which $F[T] = 0.10$, this implies that approximately 10% of the computed matches may be false positives due purely to clutter. Although it will always be the case that some percentage of the matches will be false positives ($F[T]$ descends to the value 0 only for $T > n$, which would reject every accumulator cell), the threshold value $T$ can be chosen to exert some control over the percentage of them.

# 4 Experimental Example

This section presents an in-depth example of the space-sweep approach to multi-image matching using aerial imagery from the RADIUS (Research and Development for Image Understanding Systems) project [6]. 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). Included with each image is accurate knowledge of the absolute camera position and orientation, as measured with respect to a local cartesian coordinate system with its Z-axis pointing up in the scene, parallel to gravity.

The point features used are edgels detected by the Canny edge operator [2]. This operator classifies pixels in a grey-level image to produce a binary edge image where a pixel is set to 1 if it is located on an intensity discontinuity, and 0 otherwise. Figure 4 shows a binary edge image extracted from one of the views. Note the significant amount of clutter due to trees in the scene, and a row of parked cars in front of one of the buildings.

The input data to the space-sweep matching system is the set of canny edge images plus the camera lens and pose information. The goal is to determine edgel correspondences across all seven images, as well as the 3D positions of significant surface and intensity discontinuities in the scene. Structural features of particular interest are the building rooftops and the network of walkways between the buildings. Reconstruction was carried out in a volume of space that is fully visible in all the images. The X, Y and Z dimensions of this volume are $136 \times 130 \times 30$ meters. A horizontal plane of containing an array of cells is swept through this volume along the Z-axis. Each cell is 1/3 meter square, a size chosen to roughly match the resolution of the highest resolution image. The sweeping plane pauses to sample the space of viewing ray intersections at 100 equally-spaced locations along the sweeping path, yielding approximately a 1/3-meter resolution in the vertical direction as well.

Figure 5 shows three sample plane locations along the sweeping path (specifically, locations number 28, 44 and 61 out of 100). These three levels were 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, in order to detect significant 3D feature locations.

The approximate statistical model of clutter presented in Section 3.3 needs to be validated with respect to the data, since it was based on two simplifying assumptions, namely that edgels in the each image are distributed fairly uniformly over the image, and that the Jacobian of the projective transformation from each image to the sweeping plane is roughly constant. We performed two simple tests of the clutter model. The first was to compare the number of ray intersections recorded at each $Z$-position of the sweeping plane with the expected number of votes estimated by summing up the term in Equation 9 over all images. This comparison is plotted in Figure 6a, where the dotted curve shows the actual number of votes cast, and the

solid line the number estimated by our statistical model. They are in fairly good agreement, with the largest absolute error being on the order of 2110 votes difference out of 97032, yielding a maximum relative error of roughly 2.2%. The average relative error over all 100 sweeping plane positions is around 1.7%.

The second test of the clutter model is to compare the theoretical clutter probability distribution $D[k], k = 0, 1, ..., 7$ against the empirical distributions of feature votes collected in each of the 100 sweeping plane positions. Recall that the clutter distribution $D[k]$ 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 well-known chi-square statistic [16] is used to measure how similar these two discrete distributions are for each $Z$-position of the sweeping plane; the results are plotted in Figure 6b. 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. The ground-level peak in the plot is a bit more spread out than the peak for roof-level features, because the ground actually slopes gently in the scene.

Recall that once the clutter distribution $D[k]$ 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 its 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:

| 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 |

(12)

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.

We arbitrarily choose a desired confidence level of 99% for recovered 3D scene features. That is, we are willing to tolerate only 1% false positives due to clutter. Based on this choice and the above table, the optimal choice of a 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. Each image displays the (x,y) locations of cells that are classified as scene features within a range of Z locations determined by the two peaks in Figure 6b. Specifically, the range of sweeping plane locations for ground features was chosen as 23-35 (a vertical extent of 3.6 meters) and for roof features the range was 59-63 (1.2 meters). Positions of ground features span a larger vertical range due to the slope of the terrain. It can be seen that feature locations extracted using a threshold of 5 trace out the major rooftop and walkway boundaries quite well, but there are a noticable number of false positives scattered around the image. A threshold of 6 shows significantly less clutter, but far fewer structural features as well. Choosing an optimal threshold is a balancing act; ultimately, the proper tradeoff between structure and clutter needs to determined by the application.

# 5 Summary and Extensions

This paper defines the notion of a "true multi-image" matching technique, in order 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. Several multi-image matching techniques that only operate in image-space were found not to pass this set of conditions. Techniques that can be considered to be true multi-image methods all explicitly reconstruct scene structure in object space while determining correspondences in image space. Object space seems to be the conduit through which successful multi-image methods combine information from each image.

A new space-sweep approach to true multi-image matching is presented that simultaneously determines 2D feature correspondences between multiple images together with the 3D positions of feature points in the scene. The method is based on the premise that areas of space where several viewing rays intersect are the likely locations of observed 3D scene features. It was shown that the intersections of viewing rays with a plane sweeping through space could be determined very efficiently. A statistical model of feature clutter was developed to tell how likely it is that a given number of viewing rays would 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. This approach was illustrated using a seven-image matching example from the aerial image domain.

Several extensions to this basic approach are being considered. One that is currently underway is the development of a more sophisticated model of clutter that adapts to the spatial distribution of feature points in each image. Rather than characterize the distribution of features in each image as a single uniform density, each image will be partitioned into small, local patches, with separate uniform densities estimated for each patch. Thus, areas containing many features will have a high feature density, while areas with few will have a low density. The expected benefit is the development of a scene feature detection threshold $T$ that can automatically adjust to be higher in textured areas, and lower in very homogeneous regions.

The example in Section 4 dealt with matching and reconstruction using Canny edges across multiple images. Each edgel was simply treated as a point, however, with only an $(X, Y)$ image

location to identify it. Gradient edgels have considerably more geometric structure, however. When matching edgel features, the orientations of potentially corresponding features should be taken into account. For example, when accumulating feature votes in a sweeping plane cell, only edgels with compatible orientations should be added together. Each backprojected edgel can be considered to be a tiny line segment in the sweeping plane – this line segment and the proper camera focal point form a plane. Only edgels whose plane normals are all roughly perpendicular to some 3D orientation vector should be allowed to vote for the occurance of a 3D scene edgel within the sweeping plane cell. 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] N.Ayache. *Artificial Vision for Mobile Robots.* MIT Press, Cambridge, MA, 1991.

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

[3] H.Ebner, C.Heipke and M.Holm, "Global Image Matching and Surface Reconstruction in Object Space using Aerial Images," *Integrating Photogrammetric Techniques with Scene Analysis and Machine Vision,* SPIE Vol. 1944, Orlando, 1993, pp. 44–57.

[4] J.Freund and R.Walpole, *Mathematical Statistics,* Prentice-Hall Inc, Englewood Cliffs, NJ, 1980.

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

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

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

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

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

[10] B.Horn, *Robot Vision,* MIT Press, Cambridge, MA, 1986.

[11] T.Kanade, "Development of a Video-Rate Stereo Machine," *Proc. Arpa Image Understanding Workshop,* Monterey, CA, Nov 1993, pp.549–557.

[12] R.Kumar, P.Anandan and K.Hanna, "Shape Recovery from Multiple Views: A Parallax Based Approach," *Proc. Arpa Image Understanding Workshop,* Monterey, CA, Nov 1993, pp.947–955.

[13] H.Levy, *Projective and Related Geometries,* MacMillan Co., NY, 1964.

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

[15] F.Preparate and M.Shamos, *Computational Geometry,* Springer-Verlag, New York, 1985.

[16] W.Press, S.Teukolsky, W.Vetterling, and B.Flannery, *Numerical Recipes in C,* Second Edition, Cambridge University Press, 1992.

[17] T.Schenk and C.Toth, "Use of Object Space Matching for Feature Extraction in Multiple Aerial Images," *Integrating Photogrammetric Techniques with Scene Analysis and Machine Vision,* SPIE Vol. 1944, Orlando, 1993, pp. 58–67.

[18] 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.

[19] A.Shashua, "Trilinearity in Visual Recognition by Alignment," *Proc. European Conference on Computer Vision,* LNCS-Series Vol. 800, Springer-Verlag, 1994, pp. 479–484.

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

[21] R.Tsai, "Multiframe Image Point Matching and 3D Surface Reconstruction," *IEEE Pattern Analysis and Machine Intelligence,* Vol. 5(2), 1983, pp. 159–698.

[22] J.Webb, "Implementation and Performance of Fast Parallel Multi-Baseline Stereo Vision," *Proc. Arpa Image Understanding Workshop,* Washington, DC, April 1993, pp.1005–1010.

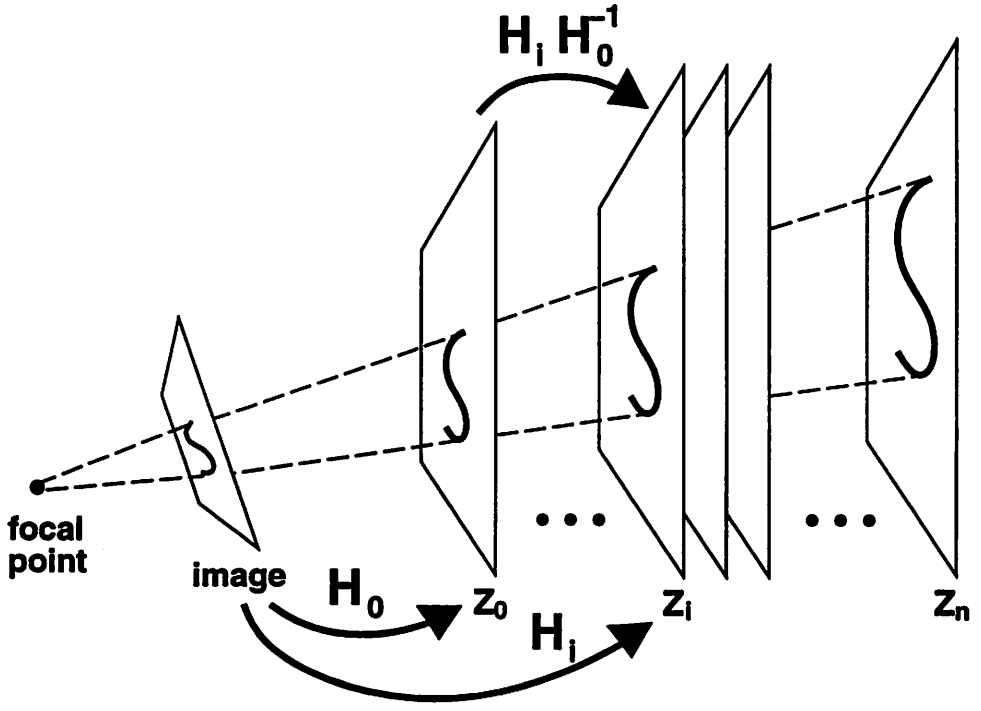


Figure 1: Projective transformations $\boldsymbol{H_0}$ and $\boldsymbol{H_i}$ between one image and two $Z$–positions of a plane sweeping through space can be replaced with the direct transformation $\boldsymbol{H_i H_0^{-1}}$.

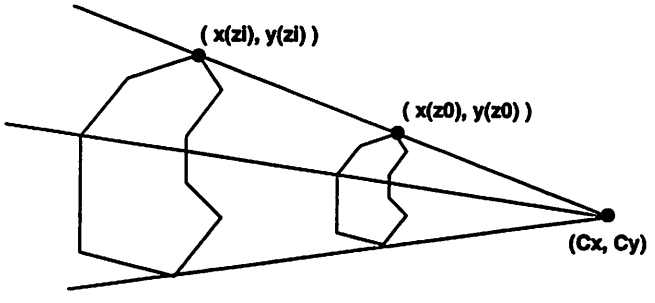


Figure 2: Transformation $\boldsymbol{H_i H_0^{-1}}$ is a dilation that maps points along trajectories defined by straight lines passing through the fixed point $(C_x, C_y)$.

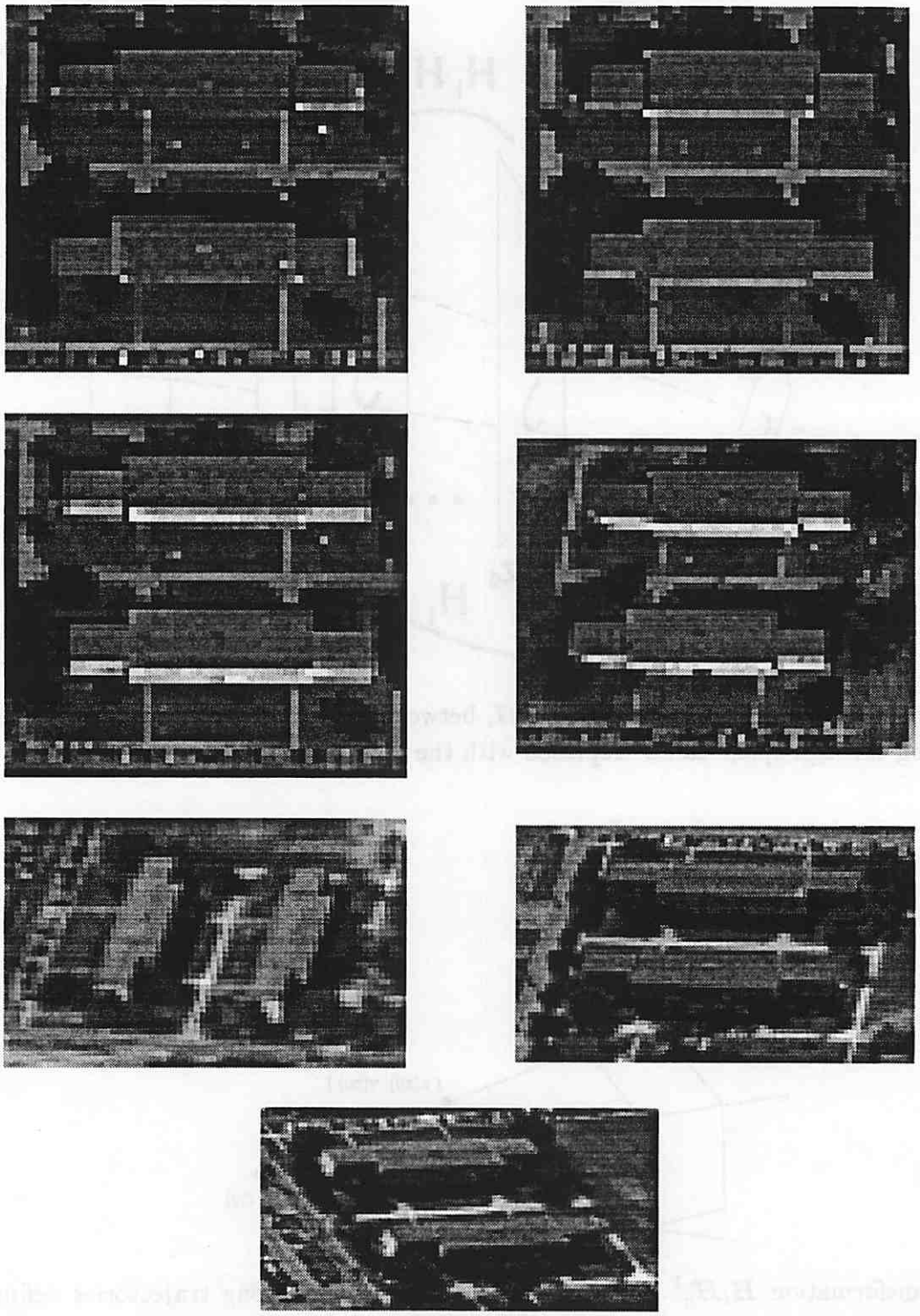

Figure 3: Seven aerial subimages of two buildings at Fort Hood, Texas.

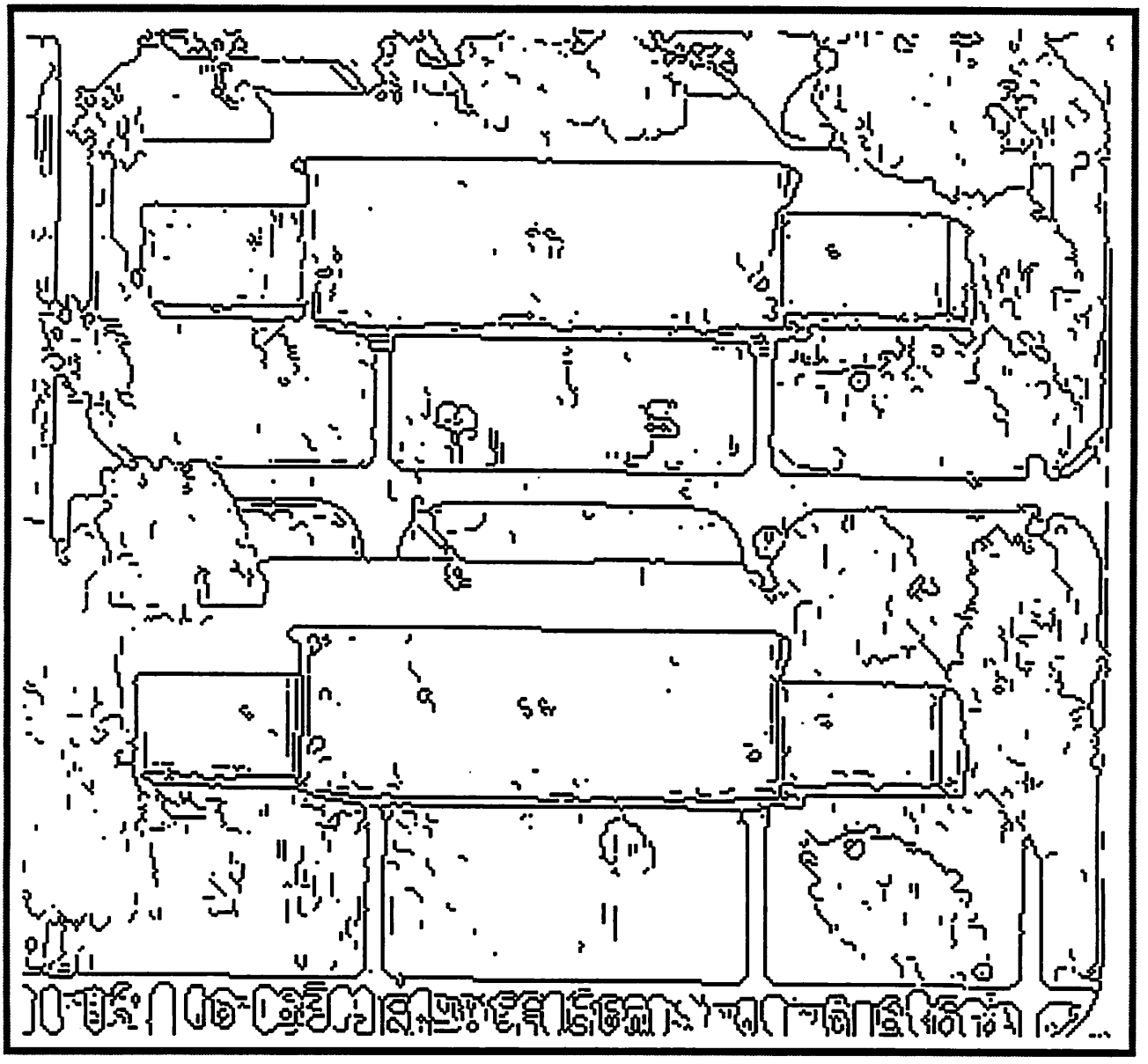

Figure 4: Canny edges extracted from the upper left hand image in Figure 3.

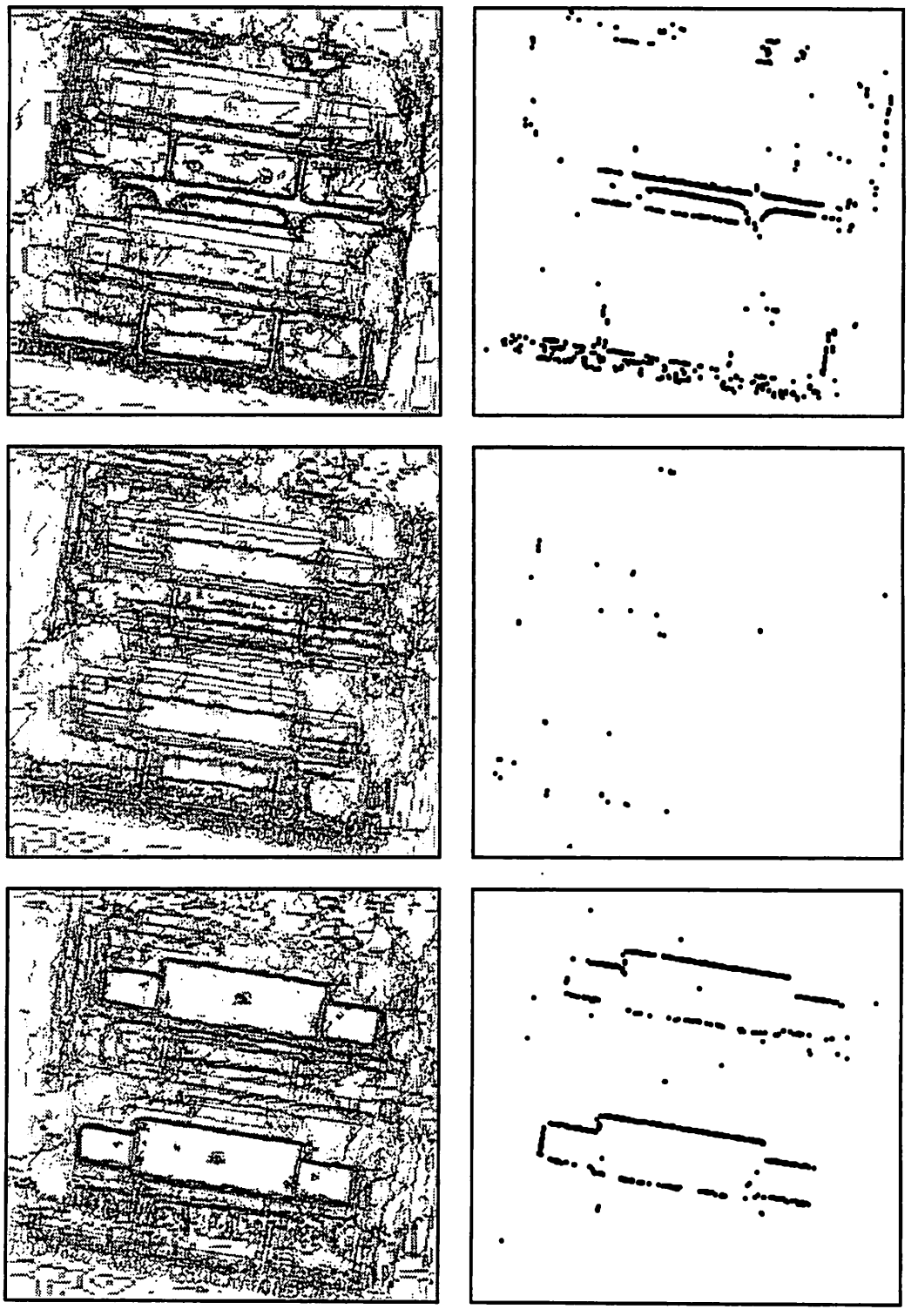

Figure 5: Three sample $Z$-positions of the sweeping plane coinciding with (top) ground-level features, (middle) no structure, and (bottom) roof features. Left shows votes in the sweeping plane, encoded by 0 = pure white and 7 = pure black. Right is the results of feature classification using a threshold value of 5.

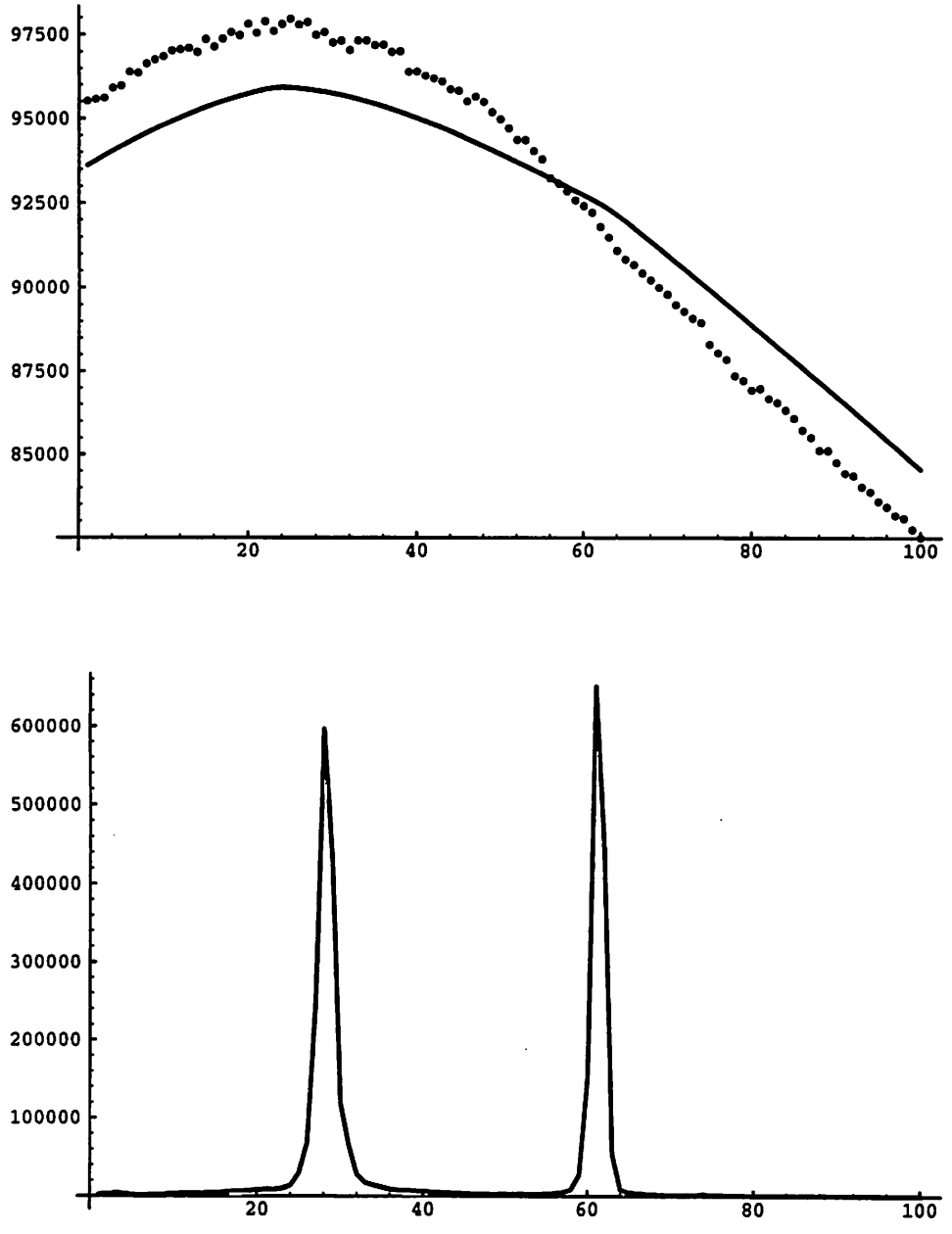


Figure 6: Two validations of the statistical clutter model. Top: expected number of votes (solid curve) versus actual number of votes (dotted curve) at each $Z$-position of the sweeping plane. Bottom: plot of chi-square test values comparing theoretical and empirical clutter distributions at each sweeping plane position (see text).

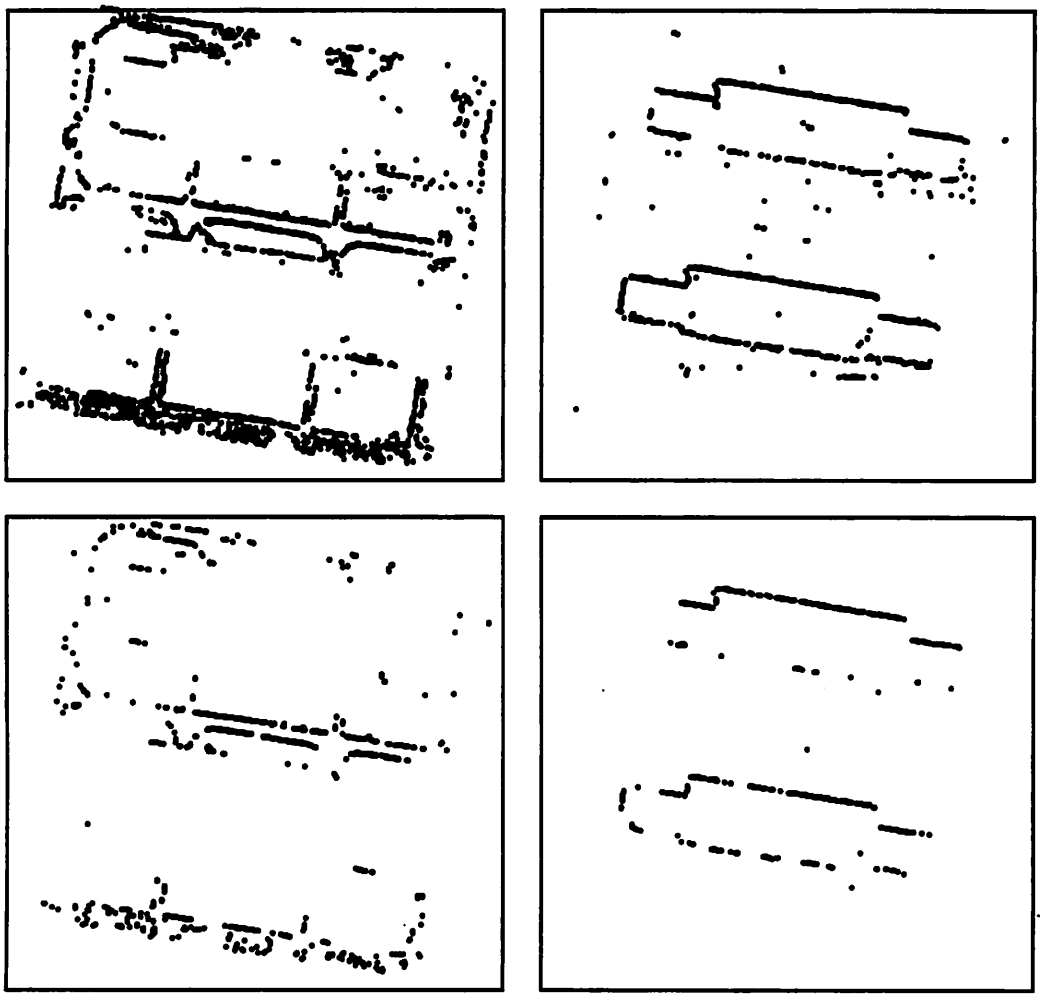

Figure 7: $XY$ locations of detected scene features for a range of $Z$-values containing ground features (top) and roof features (bottom). Results from two different threshold values of 5 (top) and 6 (bottom) are compared.