Terrain reconstruction from oblique views*

Howard Schultz
Department of Computer Science
University of Massachusetts, Amherst

Abstract

When a disparity map is computed from widely separated images the perspective distortion may result in a large number of false matches and poor reconstruction accuracy. This paper describes three image matching algorithms designed specifically to process images taken from widely varying viewpoints. They include a new match score, and modifications to standard subpixel and hierarchical matching techniques. The algorithms are incorporated into a stereo analysis package and the system is tested by processing a sequence of simulated images with base-to-height ratios that varied between 0.25 and 2.25, and a single pair of high altitude images with a base-to-height ratio of 0.63. Analysis of the simulated data showed that when these algorithms are implemented the reconstruction accuracy remains independent of the base-to-height ratio.

1 Introduction

For applications such as unmanned ground vehicles, stealth navigation, RADIUS, and sensor fusion, terrain maps must be reconstructed from images gathered from distant reconnaissance sources. These images present unique problems for terrain reconstruction systems because of their oblique viewing geometry and the associated large base-to-height ratios. In this paper, algorithms designed specifically to produce accurate elevation maps from image pairs with a large base-to-height ratio are discussed. It is assumed that the camera parameters and poses are known, and the discussion is focused on developing methods for computing a disparity map. The discussion is further limited to terrain in which the characteristic disparity of an object is roughly proportional to its horizontal dimensions (i.e., objects that cover a large area have large disparities and objects that cover a small area have small disparities), the surfaces are highly textured, and most surface features are visible in both images. In addition, attention is given mainly to retrieval accuracy; at this time computation speed is not considered.

When image matching is used to compute a disparity map, a dilemma occurs when the size of the correlation mask is selected. To increase robustness to random noise, the mask should be as large as possible, and to minimize the effects of projective distortion, the mask should be as small as possible [Mostafavi, 1978]. To help develop algorithms that balance these competing factors we take advantage of the fact that pixels near the mask center are less affected by projective distortion. The weighted cross-correlation match score (described in Section 2) and the subpixel image matching technique (described in Section 3) are designed specifically to place more emphasis on the pixels near the center of the correlation mask. In addition, a hierarchical matching scheme is discussed Section 4 that iteratively corrects for perspective distortion.

2 Weighted correlation mask

Starting with two views of an object (labeled R and L), the goal of image matching is to find pixel pairs (one in the R image and one in the L image) that view the same spot on the object. In the matching process a series of match scores $\rho(i, j, \delta i, \delta j), \delta i = \delta i_{min} \cdots \delta i_{max}$ and $\delta j = \delta j_{min} \cdots \delta j_{max}$ is computed between a window of pixels (correlation mask) centered at pixel $(i, j)_R$ in the R image and a similar mask centered at $(i + \delta i, j + \delta j)_L$ in the L image. By convention, disparities $(\delta i, \delta j)$ are defined relative to a fixed position in the R image, and a variable position in the L image. Next, the optimal disparity

*Sponsored by a grant from the Office of Naval Research (N00014-89-J-3229).

$(\delta i^*, \delta j^*)$ corresponding to the best match is selected. If $(\delta i^*, \delta j^*)$ satisfies the condition that $(i,j)_R$ and $(i+\delta i^*, j+\delta j^*)_L$ view the same spot on the object, the best match is said to be correct; otherwise, a false match has occurred. False matches occur when noise or distortion lowers the match score for the correct disparity and/or raises the match score for an incorrect disparity in such a way as to cause the wrong disparity to be selected.

One of the most robust and commonly used match scores is the cross-correlation coefficient $\rho(i,j,\delta i,\delta j)$ between a rectangular mask centered at $(i,j)_R$ and a similar mask centered at $(i+\delta i, j+\delta j)_L$. By definition $\rho(i,j,\delta i,\delta j)$ is given by

$$\rho(i,j,\delta i,\delta j) = \frac{Cov\left[I_R(i,j), I_L(i+\delta i, j+\delta j)\right]}{\sqrt{Var\left[I_R(i,j)\right] Var\left[I_L(i+\delta i, j+\delta j)\right]}} \tag{1}$$

where $I_R(i,j)$ and $I_L(i,j)$ are pixel intensities, $Cov\left[I_R(i,j), I_L(i+\delta i, j+\delta j)\right]$ is the covariance between masks, and $Var\left[I_R(i,j)\right]$ and $Var\left[I_L(i+\delta i, j+\delta j)\right]$ are the variances within each mask [Cochran and Medioni, 1992]. In this formula, $\rho(i,j,\delta i,\delta j)$ does not depend on the positions of the pixels within the mask, and all pixels contribute equally to the match score. Pixels near the center of the mask, however, are less affected by perspective distortions, and more emphasis should be given to these pixels. This can be done by assigning a weight to each pixel that depends on its position within the mask. Thus, the weighted average $E[I_R(i,j); A]$ of the pixel values within an arbitrarily shaped mask Q centered at pixel $(i,j)_R$ is

$$E\left[I_R(i,j); A\right] = \frac{1}{N} \sum_{i'} \sum_{j'} I_R(i',j') A(i-i', j-j')$$

where $(i',j') \in Q$, $A(i-i', j-j')$ are the mask weights that depend on the distance from summation index (i',j') to the mask center (i,j), and N is the total number of pixels within the mask. Furthermore, to ensure that the mask does not attenuate or amplify the image, the weights are normalized so that the average weight is unity.

$$1 = \frac{1}{N} \sum_{i'} \sum_{j'} A(i',j')$$

Likewise, the weighted variation at pixel $(i',j')_R$ is $I_R(i',j')A(i-i',j-j') - E\left[I_R(i,j); A\right]$, and the weighted variance $Var\left[I_R(i,j); A\right]$ and covariance $Cov\left[I_R(i,j), I_L(i+\delta i, j+\delta j)\right]$ are given by

$$Var\left[I_R(i,j); A\right] = \frac{1}{N-1} \times \sum_{i'} \sum_{j'} \left[I_R(i',j')A(i-i',j-j') - E\left[I_R(i,j); A\right]\right]^2$$

and

$$Cov\left[I_R(i,j), I_L(i+\delta i, j+\delta j); A\right] = \frac{1}{N-1} \sum_{i'} \sum_{j'} \left[I_R(i',j')A(i-i',j-j') - E\left[I_R(i,j); A\right]\right] \times \left[I_L(i'+\delta i, j'+\delta j)A(i-i',j-j') - E\left[I_L(i+\delta i, j+\delta j); A\right]\right]$$

and the weighted cross-correlation match score is given by

$$\rho(i,j,\delta i,\delta j; A) = \frac{Cov\left[I_R(i,j), I_L(i+\delta i, j+\delta j); A\right]}{\sqrt{Var\left[I_R(i,j); A\right] Var\left[I_L(i+\delta i, j+\delta j); A\right]}} \tag{2}$$

For the analyses presented in this paper two weighting functions are use, Gaussian weights given by

$$A(i-i', j-j') = \frac{2n+1}{2^{2n}} \cdot \frac{2m+1}{2^{2m}} \times \frac{2n!}{(n-i+i')!\,(n+i-i')!} \times \frac{2m!}{(m-j+j')!\,(m+j-j')!}$$

and uniform weights given by $A(i-i', j-j') = 1$, for $-n \leq (i-i') \leq n$ and $-m \leq (j-j') \leq m$. Note that when the uniform weights are used, the weighted cross-correlation match score (Equation 2) reduces to the conventional cross-correlation match score (Equation 1).

By assigning a weight of zero to all pixels that lie outside Q, the array of weighted averages $\mathbf{E}(I; A)$ for all pixels in an image may be computed by convolving I with the kernel A and dividing by N.

$$\mathbf{E}(I; A) = \frac{1}{N} I * A$$

Similarly, the computation formulas for the variance array $\mathbf{Var}(I; A)$ and covariance array $\mathbf{Cov}(I_R, S(I_L, \delta i, \delta j; A)$ are

$$\mathbf{Var}(I; A) = \frac{1}{N-1}\left[I^2 * A^2 - \frac{1}{N}(I * A)^2\right] \tag{3}$$

and

$$\mathbf{Cov}(I_R, S(I_L, \delta i, \delta j); A) = \frac{1}{N-1}\Big[(I_R \cdot S(I_L, \delta i, \delta j)) * A^2 - \frac{1}{N}(I_R * A) \cdot (S(I_L, \delta i, \delta j) * A)\Big]$$

where $S(I_L, \delta i, \delta j)$ is an operator that shifts an image by δi pixels in the i-dimension and δj pixels in the j-dimension. Implementation of the shift operator is important to the performance of the subpixel image matching algorithm (see Section 3 for details).

If the surface is stationary or the images are taken simultaneously, the computations may be simplified by applying epipolar constraints [Slama, 1980]. By resampling the R and L images so that each image line corresponds to an epipolar line, the vertical component of the disparity becomes identically zero, i.e., $\delta j = 0$, for all (i, j). In the following sections it is assumed that the epipolar constraints apply, thus δj is dropped from all equations and δi is replaced with δ.

3 Subpixel image matching

Reconstruction accuracy depends directly on the disparity map accuracy; therefore, significant improvements can be achieved by computing disparities to subpixel accuracy. Subpixel registration schemes rely on the assumption that near the true disparity $\tilde{\delta}$, the computed match scores are estimates of a smooth function $\tilde{\rho}(\delta)$ [Faugeras, 1993]. Thus, an estimate of the optimal disparity δ^* is found by approximating $\tilde{\rho}(\delta)$ with a model $f(\delta; c_0, c_1, \cdots)$, solving for the coefficients $c_0, c_1, \cdots$, and setting δ^* to the value of δ that optimized the model, $f'(\delta^*; c_0, c_1, \cdots) = 0$. For the subpixel matching algorithm described below, a parabolic model $f = c_0 \cdot \delta^2 + c_1 \cdot \delta + c_2$ is used; and δ^* is found by solving a least squares problem for the coefficients (c_0, c_1, c_2) and then setting $\delta^* = -2c_1/c_0$ [Tian and Huhns, 1986]. The error sources associated with this scheme are modeling error (i.e., the difference between $\tilde{\rho}(\delta)$ and $f(\delta)$) and contamination of pixel values by random noise. A detailed analysis of the effect of these error sources is beyond the scope of this paper. However, it is important to note that the effects of modeling errors and random noise become more pronounced as δ moves away from $\tilde{\delta}$ (the true disparity).

Typically, match scores are evaluated at a series of integer disparities about the previous best guess of the true disparity δ_0^*. If the interval is too narrow, an insufficient number of samples are used to estimate the location of the peak; and if the interval is too wide, the match score estimates at the ends of the interval may not be statistically significant. In either case, the location of the peak is poorly defined. For example, if $\delta = \delta_0^* + [-2, -1, 0, 1, 2]$ then only 5 observations are used to compute 3 parameters. If the range is extended to $\delta = \delta_0^* + [-4, -3, -2, -1, 0, 1, 2, 3, 4]$ the number of samples is increased to 9, but large modeling and random noise errors at the ends of the interval may contaminate the match scores used to estimate the location of the peak.

One method for solving this problem is to use a smaller disparity search range and evaluate the match scores at subpixel intervals. If the desired width of the search range is approximately ± 1.5 about the previous best guess δ_0^* and the interval between pixels is split p times, where p is an odd integer, the search range is

$$\delta_0^* - \left(\frac{3p+1}{2p}\right) + \frac{n}{p}, \quad n = 0, \cdots, 3p+1$$

For example, if $p = 5$ the disparity values are $\delta i = \delta_0^* + [-\frac{8}{5}, -\frac{7}{5}, \cdots, -\frac{1}{5}, 0, \frac{1}{5}, \cdots, \frac{7}{5}, \frac{8}{5}]$ and 17 match scores in an interval 3.2 pixels wide are used to estimate δi^*.

In the computational formulas (Equation 3) the disparities are not specified directly. Instead a subpixel shift operator $S(I_L, \delta, 0)$ is used to shift the entire image by δ pixels in the i-dimension before the convolutions with A and A^2 are computed. The shift operation is implemented by convolving I_L with an asymmetric kernel B_δ, i.e., $S(I_L, \delta, 0) = I_L * B_\delta$. For example, if $\delta = 1.2$, then $B_\delta = (0.2, 0.8, 0, 0, 0)$; and if $\delta = -0.9$, then $B_\delta = (0, 0, 0.1, 0.9, 0)$.

4 Pyramid processing

When imaging terrain it is generally true that large objects have large disparities and small objects have small disparities. When the resolution of the R and L images are reduced, smaller features disappear. Thus, only small scale disparities are lost when the low resolution images are correlated. Once the large scale disparities are recovered, the small scale disparities are recovered by processing the high resolution images and restricting the disparity search to perturbations about the previously recovered disparities. This refinement process results in a significant reduction in the amount of computation, which in addition to saving time also reduces the chance of encountering false matches. The sequential processing from low to high resolution image pairs is referred to as hierarchical, or pyramid processing [Anandan, 1989]. Note that pyramid schemes will fail when small features have large disparities (e.g., telephone poles). This happens because in the low resolution images small features are not visible and in the high resolution images the disparity search range is not sufficient to match the feature.

An image pyramid is a set of images $I^{(0)}, I^{(1)}, \dots$ of progressively diminishing resolution that are derived from a common parent image I. Resolution reduction is accomplished by smoothing and the previous layer and then selecting every other pixel. For the data presented in this paper, 4 level pyramids are used, the images are reduced by convolving with a 3×3 Gaussian kernel and selecting every other pixel.

Starting with the lowest resolution images (at the top level), an iterative process is carried out in which a disparity map is computed, expanded to match the size at the next lower level, and refined. This process continues until the final disparity map at the base level is computed. The disparity search range at all levels, except the top level, is $\pm\left(\frac{3p+1}{2p}\right)$ (p is the interval splitting factor described in Section 3). The disparity search range at the top level is set so that the disparity range at the bottom will cover the anticipated range.

At pyramid level k, the initial disparity array $D_0^{(k)}$ is formed by copying the disparities computed at the previous level $D^{(k+1)}$ into every other entry in $D_0^{(k)}$, i.e., $D_0^{(k)}(2i, 2j) = D^{(k+1)}(i, j)$, filling in the missing values in $D_0^{(k)}$ by linear interpolation, and then multiplying the entries in $D_0^{(k)}$ by two. Next, we could simply use $D_0^{(k)}$ to initialize the disparity search at level k, and compute the disparity array $D^{(k)}$ directly by matching $I_R^{(k)}$ and $I_L^{(k)}$ with the search range at pixel (i, j) given by

$$D_0^{(k)}(i,j) - \left(\frac{3p+1}{2p}\right) + \frac{n}{p}, n = 0, \cdots, 3p+1 \quad (4)$$

Or better yet, we could unwarp $I_L^{(k)}$ by making the substitution

$$I_L^{(k)}(i,j) \rightarrow I_L^{(k)}(i + D_0^{(k)}(i,j), j) \quad (5)$$

for all pixels in $I_L^{(k)}$, then compute an incremental disparity array $\Delta D^{(k)}(i,j)$ by matching $I_R^{(k)}$ and $I_L^{(k)}$ (which has just been unwarped) with the search range at pixel (i, j) given by

$$-\left(\frac{3p+1}{2p}\right) + \frac{n}{p}, n = 0, \cdots, 3p+1 \quad (6)$$

and finally update the initial guess to form the disparity array at level k.

$$D^{(k)}(i,j) = D_0^{(k)}(i,j) + \Delta D^{(k)}(i,j). \quad (7)$$

This procedure removes the perspective distortion associated with larger features. Before unwarping $I_R^{(k)}(i,j)$ and $I_L^{(k)}(i + D_0^{(k)}(i,j), j)$ view the same general spot on the surface. Whereas, after unwarping the large scale disparities are removed and $I_R^{(k)}(i,j)$ and $I_L^{(k)}(i,j)$ view the same general spot on the surface. Using $D_0^{(k)}$ to unwarp $I_L^{(k)}$ is similar to the method proposed by Schenk et al. (1980) in which $D_0^{(k)}$ is used to compute an approximate orthonormal image pair from $I_R^{(k)}$ and $I_L^{(k)}$ and then $\Delta D^{(k)}$ is computed by matching the approximate orthonormal images.

5 Terrain reconstruction

The following is a description of the basic steps taken by the terrain reconstruction system (for a detailed description see Schultz (1994)).

1. Resample the raw R and L images into epipolar coordinates.
2. Create n level image pyramids $I_R^{(0)} \cdots I_R^{(n-1)}$ and $I_L^{(0)} \cdots I_L^{(n-1)}$ (see Section 4 for details).
3. Compute the top level disparity map $D^{(n-1)}$ from $I_R^{(n-1)}$ and $I_L^{(n-1)}$ using the weighted cross-correlation match score, subpixel image matching, and hierarchical techniques described in Sections 2, 3, 4.
4. Initialize the level counter $k = n - 2$.
5. Create the initial guess $D_0^{(k)}$ by expanding $D^{(k+1)}$.
6. Unwarp $I_L^{(k)}$.

7. Compute the incremental disparity map $\Delta D^{(k)}$ by matching $I_R^{(k)}$ and $I_L^{(k)}$.

8. Update the disparity map $D^{(k)} = D_0^{(k)} + \Delta D^{(k)}$.

9. Test for the bottom level. If $k > 0$, decrement the level counter $k = k - 1$ and go to step 5, otherwise continue.

10. At the base level, calculate the world coordinate vector $\vec{\mathbf{X}}(i,j) = (X_R^{(0)}(i,j), Y_R^{(0)}(i,j), Z_R^{(0)}(i,j))$ for pixels where $D^{(0)}(i,j)$ exists. This is done by solving the stereo observation equations for all pixel pairs where a correspondence exists [Slama, 1980].

11. Create the orthonormal elevation map $\overline{Z}$ and image $\overline{I}$ by resampling the elevations $Z_R^{(0)}$ and pixel intensities $I_R^{(0)}$ onto a regularly spaced grid in world coordinates.

6 Results

The terrain reconstruction system was tested by processing three sequences of simulated images and one real image pair. To evaluate the performance of system as a function of base-to-height ratio b/h, and with and without the weighted correlation mask and subpixel image matching algorithms, a series of simulated images were analyzed. For each simulation the camera models and locations along with a random surface were specified, and an R and L image pair synthesized using a ray tracing program. Then from the camera models and synthesized images, the surface was recovered and compared to the original simulated one. The same random surface (shown in Figure 1) was used for all simulations. The horizontal dimensions of the surface is $1m \times 1m$, the rms surface height is 1.33cm, and the surface height spectrum is proportional to $k^{-4}cm^{-1}$, where k is the spatial frequency. Furthermore for all simulations, the cameras were located 10m above the surface, the focal length and orientation of the cameras were adjusted so that the entire surface fit within the camera field-of-view, and the optic axis passed through the center of the surface. A series of nine synthesized image pairs were generated for $b/h = (0.25, 0.50, 0.75, 1.00, 1.25, 1.50, 1.75, 2.00, 2.25)$. The simulated image pair for $b/h = 2.25$ is shown in Figure 2. The affects of perspective distortion are clearly visible in this image pair.

The sequence was processed by the terrain reconstruction system described in Section 5 for three sets of parameters, (1) Gaussian weights and $p = 9$, (2) uniform weights and $p = 9$, and (3) uniform weights and $p = 1$.

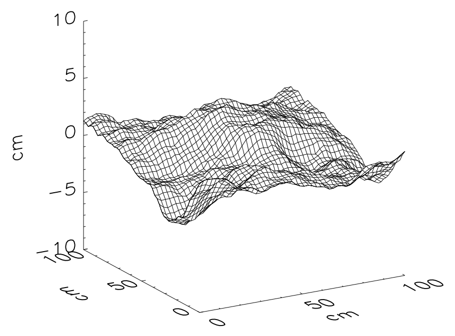

Figure 1: *The random surface used in all simulations.*

Reconstruction errors are reported in terms of the percent of the scene recovered r and the normalized elevation error s. The normalized elevation error s is the standard deviation of the elevation errors for all nodes where an elevation was computed, divided by a normalization factor s_0, i.e.,

$$s = \frac{1}{s_0} STDEV\left(\tilde{Z}(i,j) - \overline{Z}(i,j)\right)$$

where the standard deviation $STDEV$ is computed only for nodes where the recovered elevations $\overline{Z}(i,j)$ exists (see Section 5, Step 11), $\tilde{Z}(i,j)$ are the known elevations, and s_0 is a normalization factor that compensates for the natural improvement of vertical resolution with b/h. The length s_0 is equal to the height of the volume traced out by the intersection of the field-of-view of the pixels at the center of the R and L images [Matthies and Shaffer, 1987].

The simulation results are summarized in Figure 3, where r and s are plotted as a function of b/h for the three sets of parameters described above. Inspection of Figure 3 reveals that when the weighted cross-correlation match score and subpixel image matching algorithms are implemented, r and s do not depend on b/h for values of b/h at least as large as 2.25. If instead, a conventional cross-correlation match score is used, r remains constant and s grow slowly with b/h. If, in addition, integer image shifts are used instead of subpixel shifts, r and s grow more quickly with b/h.

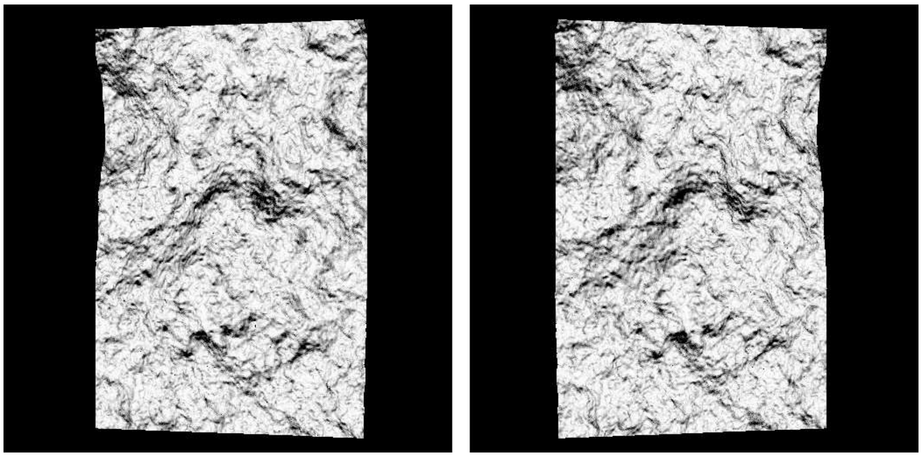

Figure 2: *Synthesized image pair with a 2.25 base-to-height ratio showing a significant amount of perspective distortion.*

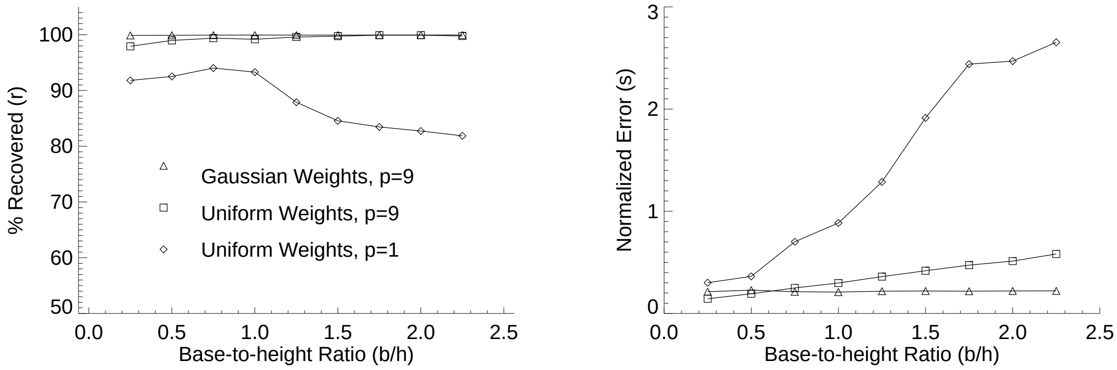

Figure 3: *The percent recovered* (r) *and normalized reconstruction error* (s) *as a function of base-to-height* (b/h) *ratio for Gaussian weights and* $p = 9$*, uniform weights and* $p = 9$*, and uniform weights and* $p = 1$*.*

In addition to the simulated data, a pair of high altitude photographs shown in Figure 4 of a building, parking lot and surrounding terrain of the Martin Marietta UGV site also were processed. The digitized images along with the camera parameters were supplied by the U. S. Army Topographic Engineering Center. The first of these images was arbitrarily assigned to the R image, while the other one was assigned to the L image. The image pair was then processed using the terrain reconstruction algorithms described above. The images were taken with the cameras looking straight down, with a base-to-height ration of 0.6295. Four level pyramids, Gaussian weights, and subpixel matching with $p = 5$ were used. At the top level the disparity search range was set to $(-12\frac{3}{5}, 13\frac{3}{5})$, and the window sizes for the 4 levels were 5×5, 9×7, 13×11 and 25×21.

The reconstructed orthonormal elevation map $\overline{Z}$ and image $\overline{I}$ are shown in Figure 5. Figures 4 and 5 appear to be rotated and reversed relative to each other because the high altitude images come from digitized negatives and the orthonormal views are displayed in world coordinates. Figure 6 shows rendered views of three areas in the test site—the building, parking lot, and a rock formation. In the rendered view of the building, sharp boundaries, especially corners, are not accurately reconstructed. However, many details of the structure, such as the flat roof and ventilation equipment, are clearly visible. In the rendered view of the parking lot, the basic shape of the cars are visible, however, the light pole is missing (only its shadow remains). This is an expected artifact because the light pole is a small feature with a large disparity. In the rendered view of the rock formation, there does not appear to be any artifacts. This part of the test site has ideal conditions for terrain reconstruction. Notice that the shading on the rocks, vegetation, gully and bare ground

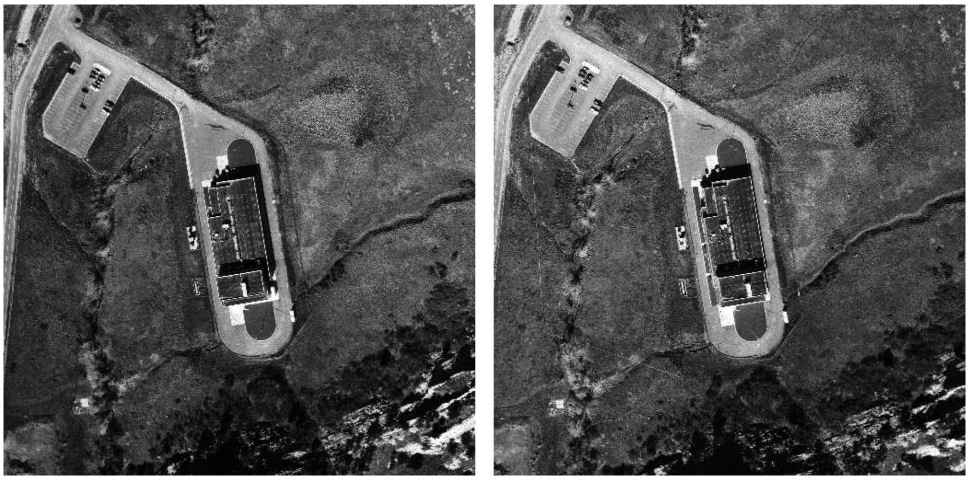

Figure 4: *Two overlapping high altitude photographs of the test site.*

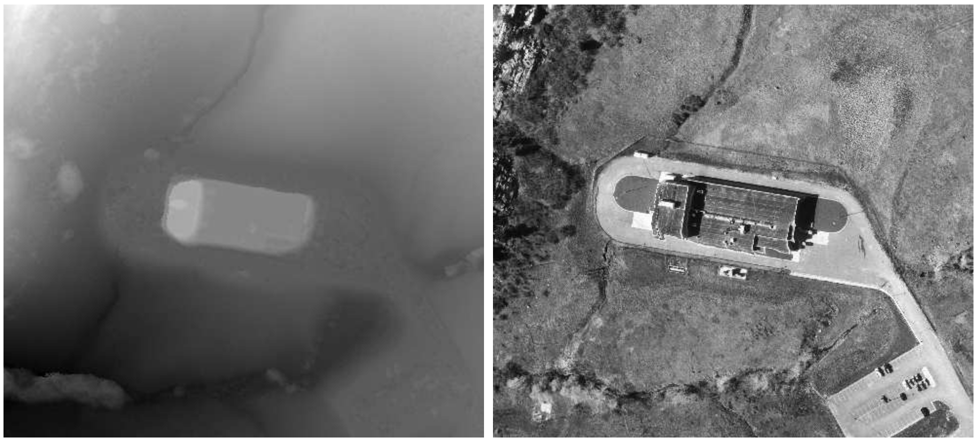

Figure 5: *The elevation map and orthonormal view of the test site.*

are consistent with the shapes of these objects.

7 Conclusions

The algorithms described in this paper were designed specifically to reconstruct terrain from oblique views. Based on analyses of simulated and real data, it appears that terrain can be successfully reconstructed from images taken from widely varying viewpoints. These procedures are especially valuable in operational scenarios, such as stealth navigation and unmanned ground vehicles, where terrain maps must be reconstructed from image data gathered from distant reconnaissance sources. We are currently setting up a series of laboratory experiments to evaluate the performance of the terrain reconstruction system under a variety of operational conditions including b/h, lens focal length, and terrain type. In additions we are in the process of integrating the terrain reconstruction and the UMass automatic building model acquisition systems [Collins et al., 1994, Jaynes et al., 1994].

Acknowledgement

I thank Robert Collins, Bruce Draper, Allen Hanson, Ed Riseman and Richard Weiss for their many helpful discussions and comments during the preparation of this paper.

References

[1] P. Anandan. A computational framework and an algorithm for the measurement of visual motion. *International Journal of Computer Vision*, 2:283–310, 1989.

[2] S. D. Cochran and G. Medioni. 3-d surface description from binocular stereo. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, PAMI-14(10):981–994, October 1992.

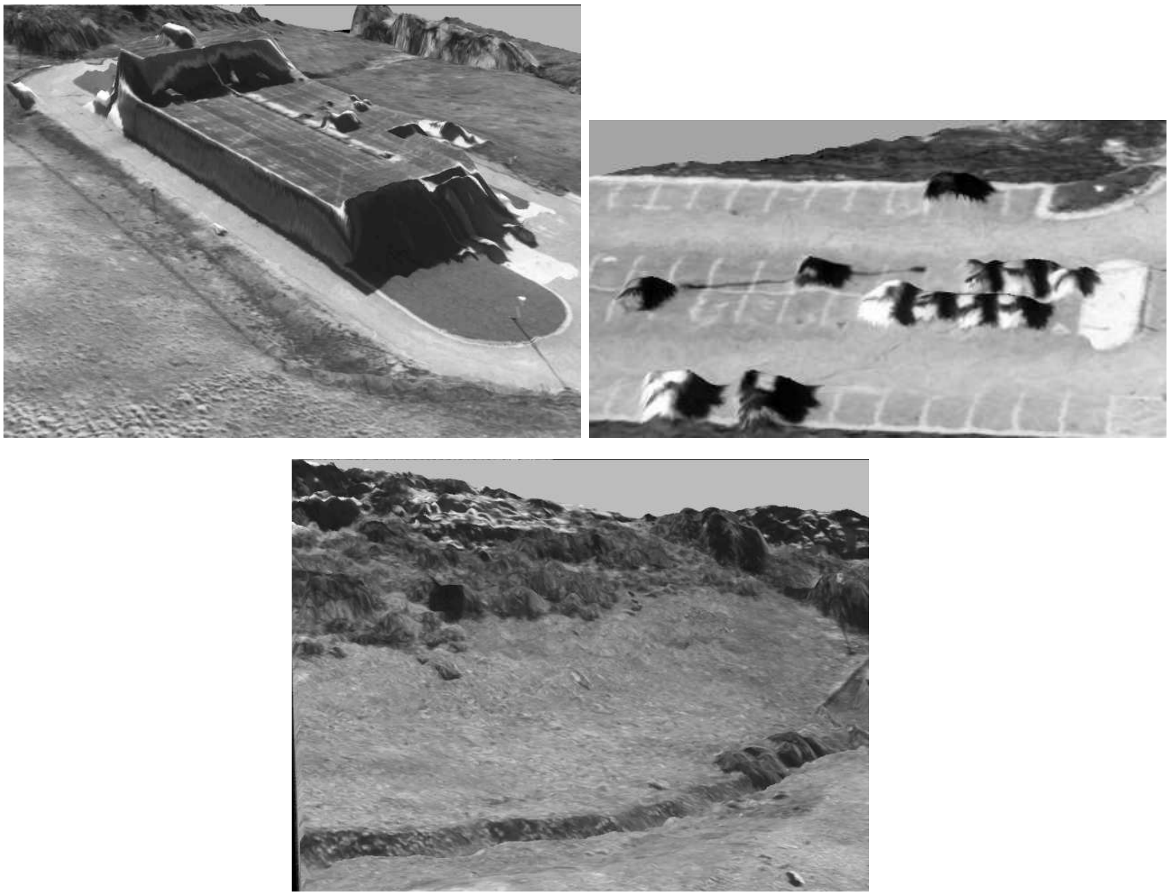

Figure 6: *Rendered views of the test site showing the building, parking lot, and a rock formation.*

[3] R. Collins, A. Hanson, and E. Riseman. Site model acquistion under the umass radius project. In *Arpa Image Understanding Workshop*, Monteray CA, November 1994.

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

[5] C. Jaynes, F. Stolle, and R. Collins. Task driven perceptual organization for extraction of rooftop polygons. In *Arpa Image Understanding Workshop*, Monteray CA, November 1994.

[6] L. Matthies and S. A. Shaffer. Error modeling in stereo navigation. *IEEE Journal of Robotics and Automation*, RA-3(3):239–248, June 1987.

[7] H. Mostafavi. Image correlation with geometric distortion. part ii: Effects on local accuracy. *IEEE Transactions on Aerospace and Electronic Systems*, AES-14(3):494–500, May 1978.

[8] T. Schenk, J. Li, and C. K. Toth. Hierarchical approach to reconstruct surfaces by using iteratively rectified imagery. In *Proc. SPIE. Close-Range Photogrammetry Meets Machine Vision*, pages 464–470, Zurich, 1990.

[9] H. Schultz. Terrain reconstruction. Technical Report UM-CS, University of Massachusetts, Department of Computer Science, Amherst, MA, 1994 in preparation.

[10] C. C. Slama, editor. *Manual of Photogrammetry, fourth Edition.* American Society of Photogrammetry, Falls Church, VA, 1980.

[11] Q. Tian and M. N. Huhns. Algorithms for subpixel registration. *Computer Vision, Graphics and Image Processing*, 35:220–233, 1986.