

A Framework for Detecting Changes in Terrain

Yvan G. Leclerc, Q.-Tuan Luong, and Pascal Fua

Artificial Intelligence Center, SRI International, Menlo Park, CA 94025

{leclerc, luong}@ai.sri.com

LIG, EPFL, Lausanne, Switzerland, fua@lig.di.epfl.ch

Abstract

We present a framework for reliably and robustly detecting changes in terrain (or other 3-D objects) over time. We first present our framework, which consists of a method for modeling terrain using a novel image-matching measure called the *coding loss*, a method for estimating the accuracy of the resulting terrain models called *self-consistency*, and method for detecting changes based on this estimate. We then present experiments using our framework.

1 Introduction

The primary objective of this project is to develop and implement a method to model and detect changes in the shape and/or material properties of terrain over time.

The basic approach we have adopted is to first model the shape and material properties of a given area of terrain, at given points in time, using an augmented version of our deformable meshes (Fua and Leclerc 1994; Fua and Leclerc 1995; Fua and Leclerc 1996). These models include a novel measure of the expected accuracy of their parameters called the *self-consistency distribution*. Using this distribution, the parameters are then compared against each

other in a statistically rigorous fashion to determine which areas of the terrain (if any) have changed. In areas where the shape and material property estimates have not changed, according to this criterion, the estimates can then be combined (again, in a statistically rigorous fashion) to improve accuracy.

There are two primary contributions that we have made in this project that are described in this paper.

The first contribution is a new image-matching measure that can be used for any multi-image stereo reconstruction algorithm, including our deformable meshes. It is based on Minimum Description Length (MDL) theory, and is called the *coding loss*. It is a measure of how much more efficiently all of the images can be encoded with a model (such as the facet of a deformable mesh) than without a model. As shown below, this measure is advantageous because it is quite good at discriminating between matches that are likely to be incorrect (such as those derived from texture-less areas) from those that are likely to be correct.

The second contribution is a methodology for estimating probability distributions of the accuracy and *self-consistency* (or precision) of the parameters of individual facets within a deformable mesh. The basic idea is that the observed variation in estimated terrain model parameters (as a function of imaging parameters, local image-matching measures, and other variables) from previous sites can be a useful predictor of the variation in these parameters for a new site. Thus, these expected variations (i.e., probability distributions) can be used to compare model parameters of a given

This project was supported in part by the Defense Advanced Research Projects Agency under contract F33615-97-C-1023 monitored by Wright Laboratory. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Defense Advanced Research Projects Agency, the United States Government, or SRI International.

Reprinted from the Proceedings of the 1998 DARPA Image Understanding Workshop
site derived from images taken at different points in time. This is based on a more general self-consistency methodology described in a separate paper in these proceedings (Leclerc, Luong et al. 1998).

In the remainder of this paper we described our coding loss image-matching measure, our self-consistency distribution, our methodology for estimating these distributions, and some experimental results.

2 Augmented Deformable Meshes

A deformable mesh is a parameterized model of the terrain derived from prior data (which can include previously computed terrain models) and controlled imagery. It is composed of a triangulated mesh comprising a number of triangular facets that cover a given area. Within each facet, the shape of the terrain is modeled as a plane parameterized by the coordinates of the three vertices of the facet.

In our augmented deformable mesh, the spatially varying material property of the terrain, which may include such components as wavelength-dependent reflectance, emissivity, or temperature of the terrain, is parameterized as a set of one or more vectors per facet. Each vector represents an estimate of the material properties in a triangular neighborhood of a point in the facet. These points are evenly spaced within the facet, and the triangular neighborhoods completely tile the facet.

Ideally, such a mesh should exactly predict the image values of all pixels covered by the z-buffered projection of a facet. We use an optimization procedure to approximate this as closely as possible. In brief, we begin with an initial estimate of the terrain model. At every step of the optimization, we adjust the parameters of the model (in general, the coordinates of the vertices of the facets and the material property parameters) so as to minimize some measure of the difference between predicted and observed image values. As described in previous papers this optimization can include other information, such as exact and approximate *a priori* geometric constraints (Fua and Leclerc 1996; Fua and Brechbuhler 1997), and can be extended to include refinement of the camera parameters (Fua and Leclerc 1994).

For the purposes of this project, the coordinate system used in the deformable terrain mesh is a local vertical coordinate system (LVCS), where the (x,y) plane is tangential to the ground plane, and the z axis is the up (gravity vector) direction. To simplify the exposition below, we will assume that the (x,y) coordinates of the vertices of the facets remain fixed for a given terrain area, and that the elevation (z coordinate) alone varies in order to model the shape of the terrain. Thus, the parameters of a facet derived from different image sets taken at different points in time correspond to the same element of a terrain, and thus can be compared directly to see if a change in the terrain has occurred.

The optimization procedure above produces excellent results in many cases. However, it occasionally falls into local minima that produce inaccurate facets. Until recently, we had no automatic procedure for determining when this happened or, more generally, estimating the accuracy of the resulting facets. Thus, it was difficult to reliably compare facets derived from different images to see if any change had occurred.

The inability to estimate the accuracy of our mesh models (which is an unsolved problem for all stereo reconstruction algorithms) is the reason we developed the coding loss image-matching measure and the self-consistency distributions described below. Note that both of these developments are applicable to other types of stereo reconstruction algorithms, as detailed in (Leclerc, Luong et al. 1998).

With these developments, which are still in progress, we believe that we will not only be able to detect changes in terrain, but we will be able to state the confidence with which the changes have been detected.

3 The Coding Loss

As described above, the mesh optimization procedure minimizes some measure of the difference between predicted and observed image values. Let's call this the image-matching measure. Note that since this measure is used in the context of our minimization procedure, the measure should be small or negative for "good" matches, and high for "bad" matches.

The image-matching measure that we have used in the past has been the square of the L_2 norm, or the sum of squared differences (SSD). For a given facet this is the sum, over all sample points, of the sum, over all images, of the squared difference between the predicted and observed image values.

The problem with the SSD measure is that it's ambiguous. That is, a low SSD measure can occur not only when the facet is correctly located (as expected), but also when the facet is incorrectly located and the terrain is spatially uniform (such as an unmarked parking lot, a flat sandy area, or a uniformly colored planar roof).

Intuitively, then, we want an image-matching measure that is low only when the match between the predicted and observed pixel values is close *and* the pixel values form a sufficiently complex pattern that it is unlikely to be matched elsewhere.

We have developed a measure that satisfies this intuitive requirement reasonably well. We call this measure the coding loss.

The coding loss is based on Minimum Description Length (MDL) theory (Rissanen 1987). In MDL theory, quantized observations of a random process are encoded using a model of that process. This model is typically divided into two components: a parameterized predictor function $M(\mathbf{z})$ and the residuals (differences) between the observations and the values predicted by that function. The residuals are typically encoded using an i.i.d. noise model (Leclerc 1989). MDL is basically a methodology for computing the parameters \mathbf{z} that yield the optimal code length for this model and for a given encoding scheme. This optimal code length is the minimum number of bits required to encode the parameters \mathbf{z} and the corresponding residuals, such that the resulting encoding is a loss-less encoding of the observations.

In our case, the process we are observing is the terrain. The observations are the pixel values in all of the images covered by a given facet. The parameters \mathbf{z} are the vertex coordinates and material property vectors of the facet. The parameterized predictor function $M(\mathbf{z})$ for the facet predicts the covered image pixel values

given the facet parameters and the fixed camera parameters.

The coding loss is the difference between the code length of the image pixels using the facet model and the code length using independent noise models. To see why this might be a good measure, consider the following cases:

- **The facet is correct and the pixel value variation is large.** Here, we expect the residuals to be small or zero (costing about zero bits to encode), and the material property samples to vary about as much as the image pixels (costing, say, C bits to encode). Thus, the cost of encoding the image pixels using the facet model will be about C bits, while the cost of encoding the image pixels in the n image will be about nC bits. Thus, the coding loss is expected to be a large negative value: $C - nC = (1 - n)C$ bits.
- **The facet is incorrect and the pixel value variation is large.** Here, we expect the material property samples as well as the residuals to vary about as much as the image pixel values. Thus, the material property samples and the residuals for a given image will each cost about C bits to encode. Thus, the cost of encoding the image pixels using the facet model will be about $(n + 1)C$, while the cost of encoding the pixel values without the facet model will be about nC . Thus, the coding loss is expected to be a large positive value: $(n + 1)C - nC = C$ bits.
- **The facet is incorrect and the pixel value variation is small.** Here, we expect the cost of encoding the property samples and the residuals to be about the same and small, say, c bits. Similarly, the cost of encoding the pixel values in a given image will be about c bits. Thus, the coding loss is expected to be a small positive value: $(n + 1)c - c = nc$ bits.

Note that cases 1 and 3 above are expected to have significantly different values for the coding loss, whereas they are expected to have about the same value for the SSD measure (namely, 0). Thus, we would expect (and have determined experimentally, see below) the coding loss to be more effective in distinguishing between correct and incorrect facet parameters.

There are many ways of encoding both the facet parameters \mathbf{z} and the residuals. The choice of encoding schemes can have a significant effect on the overall code length and, more importantly, the values of the parameters that minimize the code length. To date we have used a very simple encoding scheme, which basically assumes that the parameters and residuals are well encoded using an i.i.d. white noise model. Even though this is clearly not an optimal encoding scheme, the results are very good. Furthermore, we believe that better encoding schemes will significantly improve our results, and we plan on searching for improved encoding schemes.

4 Self-Consistency

Formally, the expected accuracy and self-consistency probability distributions mentioned in the introduction are the following conditional probability distributions:

$$A_z(x; \phi) = \text{Prob}(\mathbf{z}_i^k - z_i^* < x \mid \phi_i^k = \phi)$$

$$C_z(x; \phi) = \text{Prob}(\mathbf{z}_i^j - \mathbf{z}_i^k < x \mid \phi_i^{j,k} = \phi)$$

Where:

\mathbf{z}_i^k is a random variable representing the output of a procedure for estimating parameter z of facet f_i using image set \mathbf{S}^k . For example, \mathbf{z}_i^k could be the elevation (z -coordinate) of one of the vertices of facet f_i .

z_i^* is the correct (ground-truth) value of parameter z of facet f_i .

ϕ_i^k is called the facet's condition vector given the images \mathbf{S}^k . It is a vector of measurements and prior information, such as the image-matching measure for facet f_i and images \mathbf{S}^k and the values and co-variances of the camera parameters of the images in \mathbf{S}^k .

$\phi_i^{j,k}$ is the joint condition vector for facet f_i given images \mathbf{S}^j and \mathbf{S}^k .

Intuitively, $A_z(x; \phi)$ is a family of functions, parameterized by ϕ , that describe the expected variation in the estimate of a given facet parameter (e.g., the elevation of one of its vertices) about ground truth. If our terrain estimation procedure were perfect, the difference between the estimated parameter and ground truth would always equal zero, of course. What this function captures is the expected variation in the difference due to such factors as the inherent inaccuracies of the terrain estimation procedure and inherent ambiguities in the scene.

The condition vector ϕ_i^k for a given facet f_i is basically an index that tells us which member of the family of expected variation functions to use for a given facet f_i . Specifically, we use the member for which $\phi = \phi_i^k$.

As indicated, the elements of the condition vector depend on such things as the image-matching measure and the camera parameters. Thus, the condition vector and, hence, the expected accuracy distribution, vary from facet to facet in a given model.

Similarly, $C_z(x; \phi)$ is a family of functions that describe the expected variation in the difference between estimates of a given facet parameter of a given facet, when the estimates are derived from two image sets of a constant, unchanging terrain. Again, if our terrain estimation procedure were perfect, this difference would always equal zero.

Once we have obtained good estimates of the self-consistency distribution function $C_z(x; \phi)$, we can use it for change detection as follows.

Given $C_z(x; \phi)$ for a given parameter z , we can compute the interval within which the difference in estimated parameters ($\mathbf{z}_i^j - \mathbf{z}_i^k$) should lie, say, 99% of the time (when the estimates are derived from two different image sets \mathbf{S}^j and \mathbf{S}^k of an unchanged terrain, and their joint condition vector $\phi_i^{j,k}$ equals ϕ). This is the .99 significance interval for the parameter. If the difference in estimated parameters lies outside of this interval it is deemed to be a statistically significant difference, and we infer that parameter z (corresponding to some aspect of the terrain,

Reprinted from the Proceedings of the 1998 DARPA Image Understanding Workshop such as its elevation) has changed. Otherwise, the parameter difference is insignificant, and we can use the conditional probability distributions to optimally combine the parameters to produce a new estimate that is more accurate (i.e., one that has a lower-variance probability distribution).

Note that the joint condition vector ϕ_i^{jk} should contain all of the information required to make these distributions completely independent of the specific terrain being imaged. Such information may include, but may not be limited to:

1. the number of images and the number of pixels in each image used to derive the facet parameters, which may differ from facet to facet due to occlusions and view direction
2. the camera parameters, both external and internal
3. the camera parameter co-variances
4. co-variances of prior data used to initialize the modeling system (such as DTED or IFSARE)
5. the image matching measure
6. the angle of the sun, estimated cloud cover, and time of day for each image
7. internal parameters of the terrain modeling algorithm
8. the type of terrain

5 Estimating the Self-Consistency Distributions

Estimating probability distributions, such as the two mentioned above, requires counting the outcome of a sufficient number of appropriate experiments. For example, to estimate the probability that some random variable \mathbf{x} is less than 3.7, we count the number of times that \mathbf{x} is less than 3.7 in a set of N experiments (where \mathbf{x} may be some function of some of the observations derived from the experiment), and divide this count by N . When N is 1000, statistical estimation theory tells us that this estimate will be within about 3% of the correct probability 19 times out of 20.

An important assumption in the above example is that the distribution is stationary, which means that the experiments we perform yield a random sample drawn from the same, unchanging, distribution. Thus, it is important

to verify that the distribution remains constant across the set of experiments being performed.

For conditional distributions, these experiments need to be performed for all values of the condition variables, unless there is reason to believe that one can estimate the functional form of the distribution.

Thus there are two problems that must be solved before one can estimate and use $C_z(x; \phi)$. The first problem is choosing a joint condition vector $\phi_i^{j,k}$ that yields stationary distributions. The second problem is defining a procedure for estimating the functions. These are described below.

5.1 Choosing the condition vector ϕ_i^{jk}

As mentioned above, one should choose a condition vector such that the resulting conditional distribution is invariant to all conditions not explicitly included in it. For example, the resulting conditional distribution should be the same for all terrains and all imaging conditions. It is probably not possible to satisfy this constraint exactly given the current state-of-the-art in computer vision. However, we are striving to satisfy this as much as possible by examining factors that are both measurable and likely to have a significant impact on the conditional distribution.

So far, we have been examining the factors that affect one parameter, namely, the accuracy of the elevation of a terrain point. Intuitively, this is related to the accuracy with which its corresponding image projections can be localized in the images. There are both geometric and photometric factors that will affect this localization.

First, a point cannot be accurately localized in elevation if changing its elevation does not change its projection in any image by very much.

Second, a point cannot be accurately localized in elevation if more than one elevation produces a low value of its image-matching measure; or equivalently, when the image-matching measure is ambiguous.

A more precise way of stating the first point is that the accuracy of the elevation of a point p_i is inversely proportional to the change in elevation per unit change in image coordinates. This scale-factor, s_i^{jk} , can be also be derived so that the camera parameter co-variances are taken into account; this is formally derived in (Leclerc, Luong et al. 1998). Thus, to a first-order approximation, we can normalize our statistic $\mathbf{z}_i^j - \mathbf{z}_i^k$ by s_i^{jk} to produce a new statistic, $(\mathbf{z}_i^j - \mathbf{z}_i^k) / s_i^{jk}$, that will be invariant to camera parameters and their co-variances.

As for the second point, we have so far considered only one element of the condition vector, namely the image-matching measure.

5.2 A procedure for estimating the conditional distribution functions

As stated earlier, we would like to estimate both the accuracy distribution

$$A_z(x; \phi) = \text{Prob}(\mathbf{z}_i^k - z_i^* < x \mid \phi_i^k = \phi)$$

and the self-consistency distribution

$$C_z(x; \phi) = \text{Prob}(\mathbf{z}_i^j - \mathbf{z}_i^k < x \mid \phi_i^{j,k} = \phi).$$

$A_z(x; \phi)$ is the conditional distribution of the difference between the estimated elevation \mathbf{z}_i^k and the ground-truth elevation z_i^* as a function of the condition vector ϕ_i^k . If we had an oracle that could provide incontrovertible ground-truth for any point, and we were able to image any scene with the full range of camera parameters and co-variances, we could estimate the distribution using a simple counting method. Furthermore, we could test the hypothesis that the distribution is stationary over the space of all scenes by comparing the distributions estimated from different scenes. This is clearly infeasible.

However, we *can* estimate the self-consistency distribution $C_z(x; \phi)$, given a sufficient number of experiments. Before describing how we do this in detail below, note that

$$(\mathbf{z}_i^j - \mathbf{z}_i^k) = (\mathbf{z}_i^j - z_i^*) - (\mathbf{z}_i^k - z_i^*).$$

That is, $C_z(x; \phi)$ is the distribution of the differences between the random variables $\mathbf{z}_i^k - z_i^*$ of the accuracy distribution. When both $\mathbf{z}_i^j - z_i^*$ and $\mathbf{z}_i^k - z_i^*$ are drawn from the same zero-mean, symmetric distribution (such as a Normal distribution), then it is possible to approximately infer the accuracy distribution from the self-consistency distribution. We will explore this avenue later on in this project.

Once we have chosen the condition vector it is relatively straightforward (though computationally expensive) to estimate the self-consistency distribution for a given model parameter. For brevity of exposition, the procedure below is described for the elevation parameter of the deformable mesh terrain estimation algorithm using the coding loss as the single element of the condition vector. However, it is applicable to any parameter of any multi-image terrain estimation algorithm in which a single-element condition vector is defined.

1. Obtain n overlapping images of a given area of terrain.
2. Use a bundle-adjustment procedure to obtain accurate camera parameters and co-variances.
3. For each subset of the n images, \mathbf{S}^j , apply the terrain estimation procedure.
4. For each vertex v_i^j of the resultant mesh, compute the associated condition vector ϕ_i^j . Call the single element of the condition vector l_i^j .
5. For each pair of image subsets \mathbf{S}^j and \mathbf{S}^k , and for each pair of vertices v_i^j and v_i^k , compute the normalized elevation difference, $d_i^{jk} = (z_i^j - z_i^k) / s_i^{jk}$, and the larger of the condition values, $l_i^{jk} = \max(l_i^j, l_i^k)$.
6. For a given value l , and a given value x , count the number of times that $d_i^{jk} < x$ and $l_i^{jk} = l$. Divide this count by the total number of times that $l_i^{jk} = l$. In practice, this ratio is computed for a range of values l (i.e., $l - \delta < l_i^{jk} \leq l + \delta$) so that we can get a reasonable number of elements to count.

Reprinted from the Proceedings of the 1998 DARPA Image Understanding Workshop
 The ratio computed above is an estimate of the self-consistency distribution $C_z(x; \phi)$.

include the self-consistency curve for a uniform random distribution, drawn with diamond-shaped

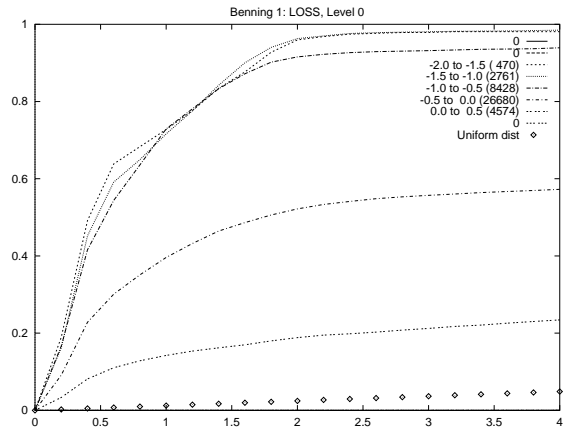
6 Experimental Results

We have performed a series of experiments testing both the utility of the coding loss versus the SSD measures, and the self-consistency of the mesh optimization procedure. These tests were performed over several areas of the Fort Benning data set prepared for the APGD project.

The first experiments were performed to test the utility of the coding loss measure versus the SSD measure as elements of the condition vector, and to test the scaling procedure used to compute the normalized elevation difference d_i^{jk} . To this end, we applied a variation of the mesh algorithm that uses a global search over independent circular facets (or disks). Though this is not necessarily a good terrain estimation algorithm *per se* (self-occlusions and continuity of terrain are ignored, for example), it does remove any source of bias introduced by the mesh optimization algorithm. Thus, we can compare different measures in an unbiased fashion. This approach will be used to test other measures during the course of the project to see whether they are an improvement over the current coding loss. As mentioned earlier, we expect that more complete image models within the coding loss framework will yield significant improvements.

6.1 Illustration and explanation of the self-consistency graphs

To illustrate the results of these experiments, consider Graph 1 below. It represents the set of self-consistency distributions for different ranges of the coding loss for the global search described above. The x-axis is the normalized difference in elevation and the y-axis is probability. For example, the topmost curve (labeled “-2.0 to -1.5 (470)”) is the probability that $d_i^{jk} < x$ for all facets in which the coding loss was between -2.0 and -1.5. In this case, there were 470 such facets, and approximately 95% of these facets had a normalized difference of less than 2.0. For this area and for the particular images used, one normalized unit on the graph corresponds to approximately .125 meters. For reference, we



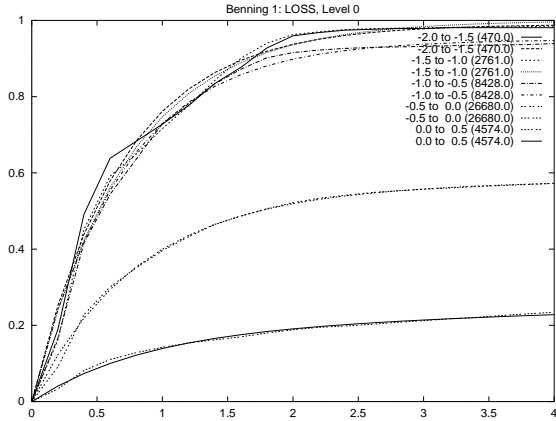
Graph 1 Self-consistency distributions for ten pairs of images of a vegetation-free area of Fort Benning points.

Note that, as the coding loss becomes more positive, the self-consistency decreases monotonically. That is, the probability that a facet has a normalized difference less than, say, 2.0 decreases monotonically. In particular, note that when the coding loss is close to 0 (in the range -0.5 to 0.0) the probability has decreased dramatically, to about 0.5. This is as we had expected. Namely, large negative coding losses should occur only when facets are uniquely identifiable, which in turn should yield self-consistent facets. As the coding loss becomes more positive, the uniqueness becomes weaker, and the facets should be less self-consistent. Furthermore, note that the self-consistency of all of the facets is significantly greater than that of a uniform random process (illustrated by the diamond-point curve).

6.2 A model of the self-consistency distributions

Another important point to note about the graphs is that the curves appear to have two distinct components. The first component, when $x < 2$, is a steeply rising curve. The second component, when $x > 2$, is approximately a straight line (actually, a very shallow quadric).

These two components are well modeled by a mixture of an exponential and a uniform distribution. The two parameters of this mixture are the width of the exponential distribution, σ , and the ratio of the exponential to the uniform distribution, λ .



Graph 2 Same as for Graph 1, but with mixture distributions fit to each curve. Note that the fits are quite close to the experimental curves.

The distribution models are illustrated in Graph 2, where each experimentally derived curve was fit with the mixture distribution described above. Note that the fits are quite close to the experimental curves. The parameters to the fits are in Table 1.

Coding loss	# of facets	λ	σ
-3.0 to -2.5	8	0.99997	0.2399734
-2.5 to -2.0	76	0.999989	0.58511
-2.0 to -1.5	470	0.989633	0.681756
-1.5 to -1.0	2761	0.999999	0.72731
-1.0 to -0.5	8428	0.947399	0.682391
-0.5 to 0.0	26680	0.55482	0.80687
0.0 to 0.5	4574	0.190606	0.881329
0.5 to 1.0	25	0.042182	0.121673

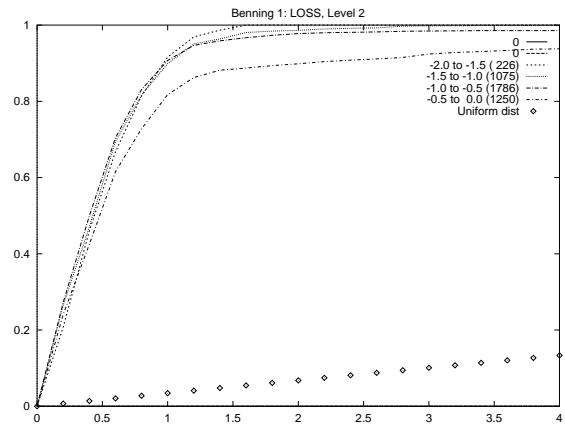
Table 1 Parameters of the fitted distributions. Column 2 is the number of facets within the coding loss range in column 1. Column 3 is the ratio of exponential to uniform distribution. Note that this decreases monotonically until coding losses approach 0. Column 4 is the width of the exponential distribution. Note that this width is approximately constant until coding losses approach 0.

Note that, as the coding loss becomes more positive, the tails of the curves (when $x > 2$) decrease monotonically. This is an indication that the ratio of exponential to uniform distribution, λ , decreases monotonically with coding loss, as seen in Table 1. Also note that

the knee of the curve occurs at the same value ($x \approx 1.5$) for all negative coding loss curves. This is an indication that σ is approximately constant when the coding loss is negative, as can be seen in Table 1.

We hypothesize that the reason the self-consistency distribution is composed of a mixture of these two distributions is that there are two distinct error processes at work. In the first case, the facet is basically correctly localized in elevation, and the differences across image pairs are due to inadequacies of the facet model and/or small errors in the camera parameters, resulting in a unimodal (in particular, exponential) distribution about the correct elevation. In the second case, the facet is basically incorrectly localized, and the projected facet corresponds to different, non-overlapping, parts of the terrain in the two images. Since there is no reason that one elevation should be favored over another in this case, we would expect the distribution to be approximately uniform over all elevations.

In summary, our experiments have allowed us to not only measure the self-consistency of facets, but have allowed us to model the expected distributions as a mixture of an exponential and uniform distribution. Furthermore, we have made two observations. First, the width, σ , of the exponential distribution remains



Graph 3 Self-consistency distributions for the same area as in Graph 1. Note that although the images used were one quarter the resolution of those used in Graph 1, the knee of the curves is still at about 1.5.

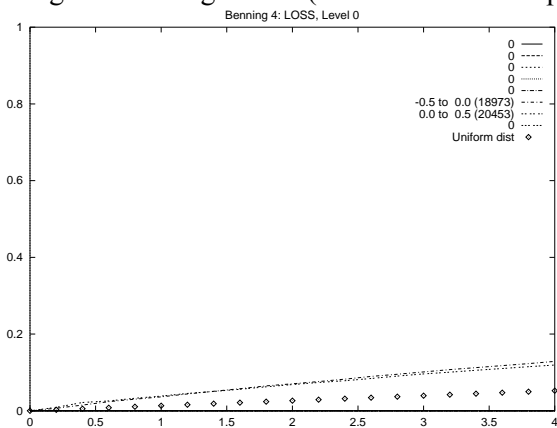
approximately constant when the coding loss is negative. Second, as the coding loss becomes more positive, the proportion of facets falling in the uniform distribution increases monotonically. These observations have held across many experiments not shown here.

6.3 Correctness of the Elevation Difference Normalization

Now consider Graphs 1 and 3. Both graphs are for the same area and images, but were derived using two different image resolutions differing by a factor of four. The important point to note about these graphs is that the knee of each curve is located at about the same value of 1.5 for all curves. This is a strong indication that the normalization has worked correctly. This observation holds over many other image sets not illustrated here.

6.4 Comparison of the Coding Loss and SSD measures

By comparison, Graph 4 shows the self-consistency curves for the same area using the SSD measure. The results are significantly poorer, since even the best SSD curve (0.0 to 6.0) is about the same as one of the worst curves for negative coding losses (-0.5 to 0.0 in Graph

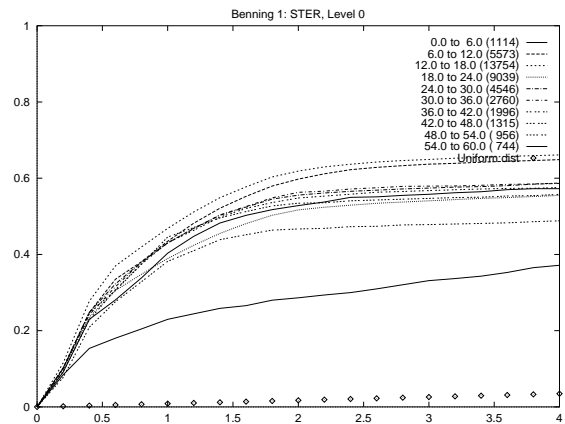


Graph 4 Self-consistency curves for a tree canopy area of Fort Benning. Note that there are no facets with coding losses below -0.5. Thus, even though the resulting facets are quite inaccurate, the coding loss is a clear indicator of this.

1). This is as expected given the arguments we presented earlier about the ambiguity of the SSD measure.

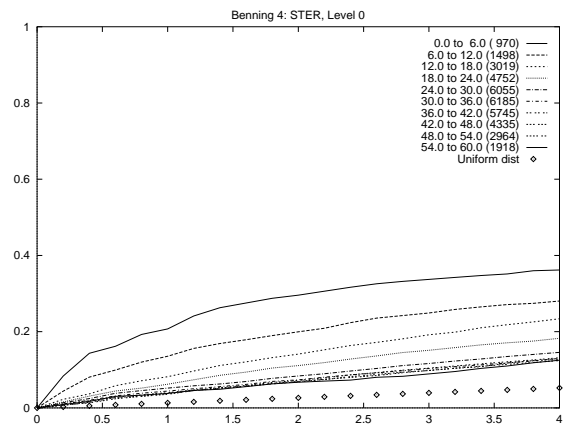
In Graphs 5 and 6 we see the self-consistency curves for a forested area of Fort Benning, where the only visible part of the terrain is the tree canopy. We would not expect our deformable mesh model to work well here (at least at high resolutions) because planar facets are very poor high-resolution models of tree canopies. This is clearly seen in both graphs because none of the curves rises above a

probability of 0.4. That is, more than 60% of facets have a normalized difference greater than 4.0.



Graph 5 Self-consistency curves for the same area and images as for Graph 1, but using the SSD measure. Note that the results are much poorer than for the coding loss, with at most 60% of the facets being within 4 normalized units of each other, compared to 90% or more for coding losses below -0.5. In fact, it is not much better than the coding losses in the range -0.5 to 0.0.

However, these graphs show that the coding loss is clearly a better predictor of the self-consistency than the SSD measure, even in this difficult case. In Graph 5, all of the facets had coding losses above -0.5, which we have seen from previous graphs occurs when the facets are inconsistent. However, Graph 6 shows a highly inconsistent curve with an SSD measure in the range 0 to 6, which had previously indicated more self-consistent results.



Graph 6 Same as for Graph 5, but using the SSD measure. Note in particular that all ranges of the SSD measure perform poorly. This shows that the SSD measure is a poor indicator of the self-consistency of the results.

7 Preliminary Change Detection Results

In parallel to this task, we have applied the preliminary results of the above framework to detecting changes in an urban scene using data from a non-DARPA client of SRI. This data comprises aerial photographs of an urban area taken a few months apart. We manually identified areas in which buildings were either added or removed, and then applied the deformable mesh algorithm to estimate the shape of the "terrain" (the topmost surfaces of the houses, streets, etc.) using the two sets of images. We then used the significance test, described in detail above, to identify changes. Unfortunately, we cannot publish the results due to client confidentiality constraints. However, we can state that the preliminary change detection results proved to be quite promising.

8 Summary and Conclusions

In summary, we have presented a number of novel elements required to attain reliable and robust detection of changes in terrain. These include: a new image-matching measure called the coding loss; a novel framework for estimating the accuracy and reliability of terrain estimate procedures, applicable to other stereo reconstruction procedures; a method for normalizing the effects of camera parameters and their co-variances; and a procedure for applying the self-consistency framework.

The experimental results based on the above framework are promising. They show that the coding loss is a good (though not perfect) predictor of the self-consistency of facets, and that it is significantly better than the traditional SSD measure. We have been able to model the self-consistency distributions quite well using a mixture of exponential and uniform distributions, and have demonstrated that the elevation normalization procedure works well.

There is still much work to be done, however. In particular, it appears that the image model used in the coding loss here does not produce perfectly stationary distributions. That is, although the shape of the self-consistency curves is well modeled by the mixture distribution over many areas of Fort Benning, the distribution parameters for a given coding

loss differ somewhat from area to area. However, the monotonic behaviour of λ and the constant value of σ do hold reasonably well over the different areas. We believe that a better image model can significantly improve the stationarity of the conditional distributions, which will yield more robust and sensitive change detection.

In addition, the experiments described here use only a single, independent measure for each facet. It is clear from previous work that conditionalizing the distribution on the measures of neighboring facets can also significantly improve both its stationarity and accuracy.

In conclusion, we believe that the work described here will provide an excellent framework for reliably and robustly detection changes in terrain. Furthermore, this framework will prove to be very useful for other 3-D reconstruction algorithms.

References

- Fua, P. and C. Brechbuhler (1997). "Imposing Hard Constraints on Deformable Models Through Optimization in Orthogonal Subspaces." CVGIP:IU 24(1): 19--35.
- Fua, P. and Y. G. Leclerc (1994). Registration Without Correspondences. CVPR, Seattle, WA.
- Fua, P. and Y. G. Leclerc (1995). "{Object-Centered Surface Reconstruction: Combining Multi-Image Stereo and Shading}." IJCV 16: 35--56.
- Fua, P. and Y. G. Leclerc (1996). "Taking Advantage of Image-Based and Geometry-Based Constraints to Recover 3--D Surfaces." Computer Vision and Image Understanding 64(1): 111--127.
- Leclerc, Y. G. (1989). "Constructing Simple Stable Descriptions for Image Partitioning." IJCV 3(1): 73-102.
- Leclerc, Y. G., Q. T. Luong, et al. (1998). Self-consistency: A novel approach to characterizing the accuracy and reliability of point correspondence algorithms. DARPA Image Understanding Workshop, Monterey, CA, Morgan Kauffman.
- Rissanen, J. (1987). "Minimum-Description-Length Principle." Encyclopedia of Statistical Sciences 5: 523--527.