Measuring the Self-Consistency of Stereo Algorithms

Yvan G. Leclerc[1], Q.-Tuan Luong[1], and P. Fua[2] *

[1] Artificial Intelligence Center, SRI International, Menlo Park, CA
leclerc,luong@ai.sri.com
[2] LIG, EPFL, Lausanne, Switzerland fua@lig.di.epfl.ch

Abstract. A new approach to characterizing the performance of point-correspondence algorithms is presented. Instead of relying on any "ground truth', it uses the self-consistency of the outputs of an algorithm independently applied to different sets of views of a static scene. It allows one to evaluate algorithms for a given class of scenes, as well as to estimate the accuracy of every element of the output of the algorithm for a given set of views. Experiments to demonstrate the usefulness of the methodology are presented.

1 Introduction and Motivation

Our visual system has a truly remarkable property: given a static natural scene, the perceptual inferences it makes from one viewpoint are almost always *consistent* with the inferences it makes from a different viewpoint. We call this property *self-consistency*.

The ultimate goal of our research is be able to design computer vision algorithms that are also self-consistent. The first step towards achieving this goal is to measure the self-consistency of the inferences of current computer vision algorithm over many scenes. An important refinement of this is to measure the self-consistency of subsets of an algorithm's inferences, subsets that satisfy certain measurable criteria, such as having a "high confidence."

Once we can measure the self-consistency of an algorithm, and we observe that this measure remains reasonably constant over many scenes (at least for certain subsets), then we can be reasonably confident that the algorithm will be self-consistent over new scenes. More importantly, such algorithms are also likely to exhibit the self-consistency property of the human visual system: *given a single view of a new scene, such an algorithm is likely to produce inferences that would be self-consistent with other views of the scene should they become available*

* This work was sponsored 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.

D. Vernon (Ed.): ECCV 2000, LNCS 1842, pp. 282–298, 2000.

later. Thus, measuring self-consistency is a critical step towards discovering (and eventually designing) self-consistent algorithms. It could also be used to learn the parameters of an algorithm that lead to self-consistency.

There are a number of caveats that one needs to make with regards to self-consistency.

First, self-consistency is a necessary, but not sufficient, condition for a computer vision algorithm to be correct. That is, it is possible (in principle) for a computer vision algorithm to be self-consistent over many scenes but be severely biased or entirely wrong. We conjecture that this cannot be the case for non-trivial algorithms. If bias can be ruled out, then the self-consistency distribution becomes a measure of the accuracy of an algorithm—one which requires no "ground truth."

Second, self-consistency must be measured over a wide variety of scenes to be a useful predictor of self-consistency over new scenes. In practice, one can measure self-consistency over certain classes of scenes, such as close-up views of faces, or aerial images of natural terrain.

In the remainder of this paper we develop a particular formalization of self-consistency and an instantiation of this formalism for the case of stereo (or, in general, multi-image point-correspondence) algorithms. We then present measurements of the self-consistency of some stereo algorithms to a variety of real images to demonstrate the utility of these measurements and compare this to previous work in estimating uncertainty.

2 A Formalization of Self-Consistency

We begin with a simple formalization of a computer vision algorithm as a function that takes an observation Ω of a world W as input and produces a set of hypotheses H about the world as output:

$$H = (h_1, h_2, \ldots, h_n) = F(\Omega, W).$$

An observation Ω is one or more images of the world taken at the same time, perhaps accompanied by meta-data, such as the time the image(s) was acquired, the internal and external camera parameters, and their covariances.

A hypothesis h nominally refers to some aspect or element of the world (as opposed to some aspect of the observation), and it nominally estimates some attribute of the element it refers to. We formalize this with the following set of functions that depend on both F and Ω:

1. $Ref(h)$, the referent of the hypothesis h (i.e., which element in the world that the hypothesis refers to).
2. $R(h, h') = Prob(Ref(h) = Ref(h')$, an estimate of the probability that two hypotheses h and h', (computed from two observations of the same world), refer to the same object or process in the world.
3. $Att(h)$, an estimate of some well-defined attribute of the referent.

4. $Acc(h)$, an estimate of the accuracy distribution of $Att(h)$. When this is well-modeled by a normal distribution, it can be represented implicitly by its covariance, $Cov(h)$.
5. $Score(h)$, an estimate of the confidence that $Att(h)$ is correct.

Intuitively, we can state that two hypotheses h and h', derived from observations Ω and Ω' of a static world W, are consistent with each other if they both refer to the same object in the world and the difference in their estimated attributes is small relative to their accuracies, or if they do not refer to the same object. When the accuracy is well modeled by a normal distribution, the consistency of two hypotheses, $C(h, h')$, can be written as

$$C(h,h') = R(h,h')(Att(h) - Att(h'))^T (Cov(h) + Cov(h'))^{-1} (Att(h) - Att(h'))^T$$

Note that the second term on the right is the Mahalanobis distance between the attributes, which we refer to as the normalized distance between attributes throughout this paper.

Given the above, we can measure the self-consistency of an algorithm as the histogram of $C(h, h')$ over all pairs of hypotheses in $H = F(\Omega(W))$ and $H' = F(\Omega'(W))$, over all observations over all suitable static worlds W. We call this distribution of $C(h, h')$ the self-consistency distribution of the computer vision algorithm F over the worlds W. To simplify the exposition below, we compute this distribution only for pairs h and h' for which $R(h, h') \approx 1$. We will discuss the utility of the full distribution in future work.

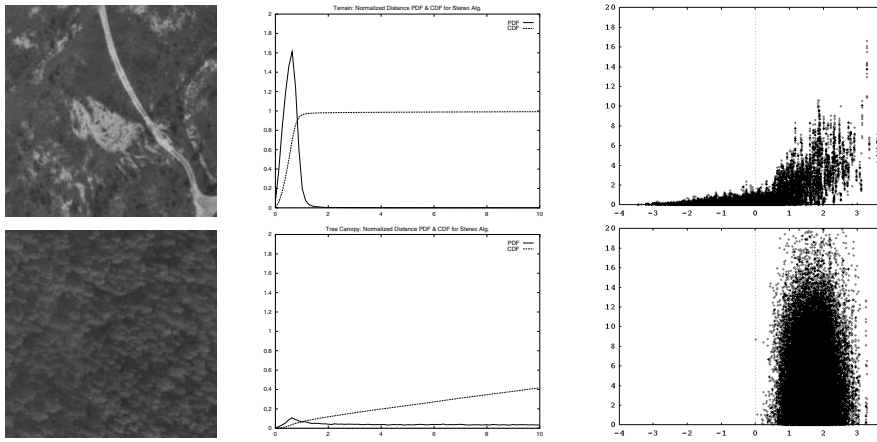
3 Self-Consistency of Stereo Algorithms

We can apply the above abstract self-consistency formalism to stereo algorithms ([14,12]). For the purposes of this paper, we assume that the projection matrices and associated covariances are known for all images.

The hypothesis h produced by a traditional stereo algorithm is a pair of image coordinates $(\mathbf{x_0}, \mathbf{x_1})$ in each of two images, (I_0, I_1). In its simplest form, a stereo match hypothesis h asserts that the closest opaque surface element along the optic ray through $\mathbf{x_0}$ is the same as the closest opaque surface element along the optic ray through $\mathbf{x_1}$. That is, the referent of h, $Ref(h)$, is the closest opaque surface element along the optic rays through both $\mathbf{x_0}$ and $\mathbf{x_1}$.

Consequently, two stereo hypotheses have the same referent if their image coordinates are the same in one image. In other words, if we have a match in image pair (I_0, I_1) and a match in image pair (I_1, I_2), then the stereo algorithm is asserting that they refer to the same opaque surface element when the coordinates of the matches in image I_1 are the same. Self-consistency, in this case, is a measure of how often (and to what extent) this assertion is true.

The above observation can be used to write the following set of associated functions for a stereo algorithm. We assume that all matches are accurate to within some nominal accuracy, σ, in pixels (typically $\sigma = 1$). This can be extended to include the full covariance of the match coordinates.



(a) sample image (b) self-consistency distributions (c) scatter diagrams with MDL score

Fig. 1. Results on two different types of images: terrain (top) vs. tree canopy (bottom).

1. $Ref(h)$, The closest opaque surface element visible along the optic rays through the match points.
2. $R(h, h') = 1$ if h and h' have the same coordinate (within σ) in one image; 0 otherwise.
3. $Att(h)$, The triangulated 3D (or projective) coordinates of the surface element.
4. $Acc(h)$, The covariance of $Att(h)$, given that the match coordinates are $N(\mathbf{x_0}, \sigma)$ and $N(\mathbf{x_0}, \sigma)$ random variables.
5. $Score(h)$, A measure such as normalized cross-correlation or sum of squared differences.

Without taking into account $Score(h)$, the self-consistency distribution is the histogram of normalized differences in triangulated 3D points for pairs of matches with a common point in one image (Sec. 4.2). Items 3 and 4 above will be expanded upon in Sec. 5. One way, further described in Sec. 4.3, to take into account $Score(h)$ is to plot a scatter diagram using as x-axis $Score(h)$, and as y-axis the normalized differences in triangulated 3D points.

4 The Self-Consistency Distribution

4.1 A Methodology for Estimating the Self-Consistency Distribution

Ideally, the self-consistency distribution should be computed using all possible variations of viewpoint and camera parameters (within some class of variations) over all possible scenes (within some class of scenes). However, we can compute an estimate of the distribution using some small number of images of a scene, and average this distribution over many scenes.

Here we start with some fixed collection of images assumed to have been taken at exactly the same time (or, equivalently, a collection of images of a static scene taken over time). Each image has a unique index and associated projection matrix and (optionally) projection covariances. We then apply a stereo algorithm independently to all pairs of images in this collection.[1] Each such pair of images is an observation in our formalism. The image indices, match coordinates, and score, are reported in *match* files for each image pair.

We now search the match files for pairs of matches that have the same coordinate in one image. For example, if a match is derived from images 1 and 2, another match is derived from images 1 and 3, and these two matches have the same coordinate in image 1, then these two matches have the same referent. Such a pair of matches, which we call a *common-point match set*, should be self-consistent because they should correspond to the same point in the world. This extends the principle of the trinocular stereo constraint [22,2] to arbitrary camera configurations and multiple images.

Given two matches in a common-point match set, we can now compute the distance between their triangulations, after normalizing for the camera configurations (see Sec. 5). The histogram of these normalized differences, computed over all common-point matches, is our estimate of the self-consistency distribution.

Another distribution that one could compute using the same data files would involve using all the matches in a common-point match set, rather than just pairs of matches. For example, one might use the deviation of the triangulations from the mean of all triangulations within a set. This is problematic for several reasons.

First, there are often outliers within a set, making the mean triangulation less than useful. One might mitigate this by using a robust estimation of the mean. But this depends on various (more or less) arbitrary parameters of the robust estimator that could change the overall distribution.

Second, and perhaps more importantly, we see no way to extend the normalization used to eliminate the dependence on camera configurations, described in Sec. 5, to the case of multiple matches.

Third, we see no way of using the above variants of the self-consistency distribution for change detection.

4.2 An Example of the Self-Consistency Distribution

To illustrate the self-consistency distribution, we first apply the above methodology to the output of a simple stereo algorithm [7]. The algorithm first rectifies the input pair of images and then searches for 7×7 windows along scan lines that maximize a normalized cross-correlation metric. Sub-pixel accuracy is achieved by fitting a quadratic to the metric evaluated at the pixel and its two adjacent

[1] Note that the "stereo" algorithm can find matches in $n > 2$ images. In this case, the algorithm would be applied to all subsets of size n. We use $n = 2$ to simplify the presentation here.

neighbors. The algorithm first computes the match by comparing the left image against the right and then comparing the right image against the left. Matches that are not consistent between the two searches are eliminated. Note that this is a way of using self-consistency as a filter.

The stereo algorithm was applied to all pairs of five aerial images of bare terrain, one of which is illustrated in the top row of Figure 1(a). These images are actually small windows from much larger images (about 9000 pixels on a side) for which precise ground control and bundle adjustment were applied to get accurate camera parameters.

Because the scene consists of bare, relatively smooth, terrain with little vegetation, we would expect the stereo algorithm described above to perform well. This expectation is confirmed anecdotally by visually inspecting the matches.

However, we can get a quantitative estimate for the accuracy of the algorithm for this scene by computing the self-consistency distribution of the output of the algorithm applied to the ten images pairs in this collection. Figure 1(b) shows two versions of the distribution. The solid curve is the probability density (the probability that the normalized distance equals x). It is useful for seeing the mode and the general shape of the distribution. The dashed curve is the cumulative probability distribution (the probability that the normalized distance is less than x). It is useful for seeing the median of the distribution (the point where the curve reaches 0.5) or the fraction of match pairs with normalized distances exceeding some value.

In this example, the self-consistency distribution shows that the mode is about 0.5, about 95% of the normalized distances are below 1, and that about 2% of the match pairs have normalized distances above 10.

In the bottom row of Figure 1 we see the self-consistency distribution for the same algorithm applied to all pairs of five aerial images of a tree canopy. Such scenes are notoriously difficult for stereo algorithms. Visual inspection of the output of the stereo algorithm confirms that most matches are quite wrong. This can be quantified using the self-consistency distribution in Figure 1(b). Here we see that, although the mode of the distribution is still about 0.5, only 10% of the matches have a normalized distance less than 1, and only 42% of the matches have a normalized distance less than 10.

Note that the distributions illustrated above are not well modelled using Gaussian distributions because of the predominance of outliers (especially in the tree canopy example). This is why we have chosen to compute the full distribution rather than use its variance as a summary.

4.3 Conditionalization

As mentioned in the introduction, the global self-consistency distribution, while useful, is only a weak estimate of the accuracy of the algorithm. This is clear from the above examples, in which the unconditional self-consistency distribution varied considerably from one scene to the next.

However, we can compute the self-consistency distribution for matches having a given "score" (such as the MDL-base score described in detail in Appendix A).

This is illustrated in Figure 1(c) using a scatter diagram. The scatter diagram shows a point for every pair of matches, the x coordinate of the point being the larger of the scores of the two matches, and the y coordinate being the normalized distance between the matches.

There are several points to note about the scatter diagrams. First, the terrain example (top row) shows that most points with scores below 0 have normalized distances less than about 1. Second, most of the points in the tree canopy example (bottom row) are not self-consistent. Third, none of the points in the tree canopy example have scores below 0. Thus, it would seem that this score is able to segregate self-consistent matches from non-self-consistent matches, even when the scenes are radically different (see Sec. 6.3).

5 Projection Normalization

To apply the self-consistency method to a set of images, all we need is the set of projection matrices in a common projective coordinate system. This can be obtained from point correspondences using projective bundle adjustment [16,19] and does not require camera calibration. The Euclidean distance is not invariant to the choice of projective coordinates, but this dependence can often be reduced by using the normalization described below. Another way to do so, which actually cancels the dependence on the choice of projective coordinates, is to compute the difference between the reprojections instead of the triangulations, as described in more detail in [14]. This, however, does not cancel the dependence on the relative geometry of the cameras.

5.1 The Mahalanobis Distance

Assuming that the contribution of each individual match to the statistics is the same ignores many imaging factors like the geometric configuration of the cameras and their resolution, or the distance of the 3D point from the cameras. There is a simple way to take into account all of these factors, applying a normalization which make the statistics invariant to these imaging factors. In addition, this mechanism makes it possible to take into account the uncertainty in camera parameters, by including them into the observation parameters.

We assume that the observation error (due to image noise and digitalization effects) is Gaussian. This makes it possible to compute the covariance of the reconstruction given the covariance of the observations. Let us consider two reconstructed estimates of a 3-D point, M_1 and M_2 to be compared, and their computed covariance matrices Λ_1 and Λ_2. We weight the squared Euclidean distance between M_1 and M_2 by the sum of their covariances. This yields the squared *Mahalanobis distance*: $(\mathbf{M}_1 - \mathbf{M}_2)^T(\Lambda_1 + \Lambda_2)^{-1}(\mathbf{M}_1 - \mathbf{M}_2)$.

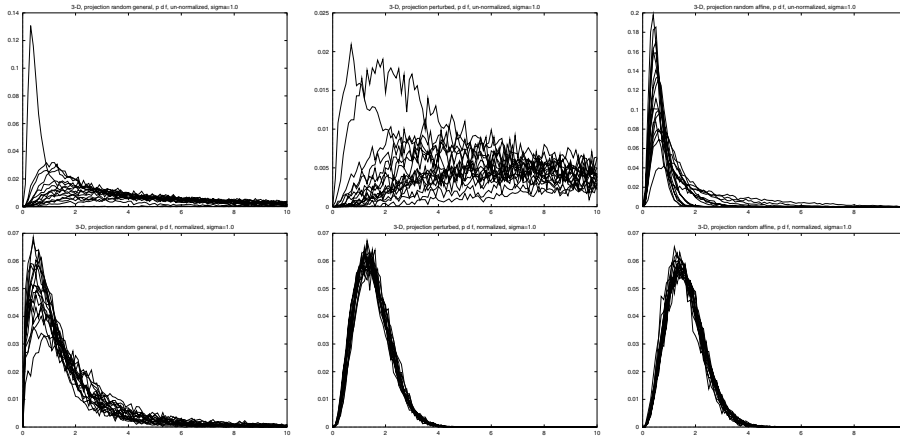
5.2 Determining the Reconstruction and Reprojection Covariances

If the measurements are modeled by the random vector $\mathbf{x}$, of mean $\mathbf{x}_0$ and of covariance $\mathbf{\Lambda}_\mathbf{x}$, then the vector $\mathbf{y} = f(\mathbf{x})$ is a random vector of mean is $f(\mathbf{x}_0)$

and, up to the first order, covariance $\mathbf{J}_f(\mathbf{x}_0)\mathbf{\Lambda}_\mathbf{x}\mathbf{J}_f(\mathbf{x}_0)^T$, where $\mathbf{J}_f(\mathbf{x}_0)$ is the Jacobian matrix of f, at the point $\mathbf{x}_0$.

In order to determine the 3-D distribution error in reconstruction, the vector $\mathbf{x}$ is defined by concatenating the 2-D coordinates of each point of the match, ie $[x_1, y_1, x_2, y_2, \ldots x_n, y_n]$ and the result of the function is the 3-D coordinates X, Y, Z of the point M reconstructed from the match, in the least-squares sense. The key is that M is expressed by a closed-form formula of the form $\mathbf{M} = (\mathbf{L}^T\mathbf{L})^{-1}\mathbf{L}^T\mathbf{b}$, where $\mathbf{L}$ and $\mathbf{b}$ are a matrix and vector which depend on the projection matrices and coordinates of the points in the match. This makes it possible to obtain the derivatives of M with respect to the $2n$ measurements $w_i, i = 1 \ldots n, w = x, y$. We also assume that the errors at each pixel are independent, uniform, and isotropic. The covariance matrix $\mathbf{\Lambda}_\mathbf{x}$ is then diagonal, therefore each element of Λ_M can be computed as a sum of independent terms for each image.

The above calculations are exact when the mapping between the vector of coordinates of m_i and M (resp. m'_j and M') is linear, since it is only in that case that the distribution of M and M' is Gaussian. The reconstruction operation is exactly linear only when the projection matrices are affine. However, the linear approximation is expected to remain reasonable under normal viewing conditions, and to break down only when the projection matrices are in configurations with strong perspective.



random general projections perturbed projections random affine projections

Fig. 2. Un-normalized (top) vs normalized (bottom) self-consistency distributions.

6 Experiments

6.1 Synthetic Data

In order to gain insight into the nature of the normalized self-consistency distributions, we investigate the case when the noise in point localization is Gaussian.

We first derive the analytical model for the self-consistency distribution in that case. We then show, using monte-carlo experiments that, provided that the geometrical normalization described in Sec.5 is used, the experimental self-consistency distributions fit this model quite well when perspective effects are not strong. A consequence of this result is that under the hypothesis that the error localization of the features in the images is Gaussian, the self-consistency distribution could be used to recover exactly the accuracy distribution.

Modeling the Gaussian self-consistency distributions. The squared Mahalanobis distance in 3D follows a chi-square distribution with three degrees of freedom:

$$\chi^2_3 = \frac{1}{\sqrt{2\pi}}\sqrt{x}e^{-x/2}$$

In our model, the Mahalanobis distance is computed between M, M', reconstructions in 3D, which are obtained from matches m_i, m'_j of which coordinates are assumed to be Gaussian, zero-mean and with standard deviation σ. If M, M' are obtained from the coordinates m_i, m'_j with a linear transformation A, A', then the covariances are $\sigma^2 AA^T$, $\sigma^2 A'A'^T$. The Mahalanobis distance follows the distribution:

$$d_3 = x^2/\sigma^3\sqrt{2/\pi}e^{-x^2/2\sigma^2} \tag{1}$$

Using the Mahalanobis distance, the self-consistency distributions should be *statistically* independent of the 3D points and projection matrices. Of course, if we were just using the Euclidean distance, there would be no reason to expect such an independence.

Comparison of the normalized and unnormalized distributions To explore the domain of validity of the first-order approximation to the covariance, we have considered three methods to generate random projection matrices:

1. General projection matrices are picked randomly.
2. Projection matrices are obtained by perturbing a fixed, realistic matrix (which is close to affine). Entries of this matrix are each varied randomly within 500% of the initial value.
3. Affine projection matrices are picked randomly.

Each experiment in a set consisted of picking random 3D points, random projection matrices according to the configuration previously described, projecting them, adding random Gaussian noise to the matches, and computing the self-consistency distributions by labelling the matches so that they are perfect.

To illustrate the invariance of the distribution that we can obtain using the normalization, we performed experiments where we computed both the normalized version and the unnormalized version of the self-consistency. As can be seen in Fig. 2, using the normalization reduced dramatically the spread of the self-consistency curves found within each experiment in a set. In particular, in the two last configurations, the resulting spread was very small, which indicates that the geometrical normalization was successful at achieving invariance with respect to 3D points and projection matrices.

Comparison of the experimental and theoretical distributions Using the Mahalanobis distance, we then averaged the density curves within each set of experiments, and tried to fit the model described in Eq. 1 to the resulting curves, for six different values of the standard deviation, $\sigma = 0.5, 1, 1.5, 2, 2.5, 3$. As illustrated in Fig. 3, the model describes the average self-consistency curves very well when the projection matrices are affine (as expected from the theory), but also when they are obtained by perturbation of a fixed matrix. When the projection matrices are picked totally at random, the model does not describe the curves very well, but the different self-consistency curves corresponding to each noise level are still distinguishable.

6.2 Comparing Two Algorithms

The experiments described here and in the following section are based on the application of stereo algorithms to seventeen scenes, each comprising five images, for a total of 85 images and 170 image pairs. At the highest resolution, each image is a window of about 900 pixels on a side from images of about 9000 pixels on a side. Some of the experiments were done on Gaussian-reduced versions of the images. These images were controlled and bundle-adjusted to provide accurate camera parameters.

A single self-consistency distribution for each algorithm was created by merging the scatter data for that algorithm across all seventeen scenes. In previous papers, [14,11], we compared two algorithms, but using data from only four images. By merging the scatter data as we do here, we are now able to compare algorithms using data from many scenes. This results in a much more comprehensive comparison.

The merged distributions are shown in Figure 4 as probability density functions for the two algorithms. The solid curve represents the distribution for our deformable mesh algorithm [8], and the dashed curve represents the distribution for the stereo algorithm described above.

Comparing these two graphs shows some interesting differences between the two algorithms. The deformable mesh algorithm clearly has more outliers (matches with normalized distances above 1), but has a much greater proportion of matches with distances below 0.25. This is not unexpected since the strength of the deformable meshes is its ability to do very precise matching between images. However, the algorithm can get stuck in local minima. Self-consistency now allows us to quantify how often this happens.

But this comparison also illustrates that one must be very careful when comparing algorithms or assessing the accuracy of a given algorithm. The distributions we get are very much dependent on the scenes being used (as would also be the case if we were comparing the algorithms against ground truth—the "gold standard" for assessing the accuracy of a stereo algorithm). In general, the distributions will be most useful if they are derived from a well-defined class of scenes. It might also be necessary to restrict the imaging conditions (such as resolution or lighting) as well, depending on the algorithm. Only then can the distribution be used to predict the accuracy of the algorithm when applied to images of similar scenes.

6.3 Comparing Three Scoring Functions

To eliminate the dependency on scene content, we propose to use a score associated with each match. We saw scatter diagrams in Figure 1(c) that illustrated how a scoring function might be used to segregate matches according to their expected self-consistency.

In this section we will compare three scoring functions, one based on Minimum Description Length Theory (the MDL score, Appendix A), the traditional sum–of–squared–differences (SSD) score, and the SSD score normalized by the localization covariance (SSD/GRAD score) [6]. All scores were computed using the same matches computed by our deformable mesh algorithm applied to all image pairs of the seventeen scenes mentioned above. The scatter diagrams for all of the areas were then merged together to produce the scatter diagrams show in Figure 5.

The MDL score has the very nice property that the confidence interval (as defined earlier) rises monotonically with the score, at least until there is a paucity of data, when then score is greater than 2. It also has a broad range of scores (those below zero) for which the normalized distances are below 1, with far fewer outliers than the other scores.

The SSD/GRAD score also increases monotonically (with perhaps a shallow dip for small values of the score), but only over a small range.

The traditional SSD score, on the other hand, is distinctly not monotonic. It is fairly non-self-consistent for small scores, then becomes more self-consistent, and then rises again.

6.4 Comparing Window Size

One of the common parameters in a traditional stereo algorithm is the window size. Figure 6 presents one image from six urban scenes, where each scene comprised four images. Figure 7 shows the merged scatter diagrams (a) and global self-consistency distributions (b) for all six scenes, for three window sizes (7×7, 15×15, and 29×29). Some of the observations to note from these experiments are as follows.

First, note that the scatter diagram for the 7×7 window of this class of scenes has many more outliers for scores below -1 than were found in the scatter diagram

for the terrain scenes. This is reflected in the global self-consistency distribution in (b), where one can see that about 10% of matches have normalized distances greater than 6. The reason for this is that this type of scene has significant amounts of repeating structure along epipolar lines. Consequently, a score based only on the quality of fit between two windows (such as the MDL-based score) will fail on occasion. A better score would include a measure of the uniqueness of a match along the epipolar line as a second component. We are currently exploring this.

Second, note that the number of outliers in both the scatter diagram and the self-consistency distributions decreases as window size decreases. Thus, large window sizes (in this case) produce more self-consistent results. But it also produces fewer points. This is probably because this stereo algorithm uses left-right/right-left equality as a form a self-consistency filter.

We have also visually examined the matches as a function of window size. When we restrict ourselves to matches with scores below -1, we observe that matches become sparser as window size increases. Furthermore, it appears that the matches are more accurate with larger window sizes. This is quite different from the results of Faugeras *et al.*— [5]. There they found that, in general, matches became denser but less accurate as window size increased. We believe that this is because an MDL score below -1 keeps only those matches for which the scene surface is approximately fronto-parallel within the extent of the window, which is a situation in which larger window sizes increases accuracy. This is borne out by our visual observations of the matches. On the other hand, this result is basically in line with the results of Szeliski and Zabih [18,20], who show that prediction error decreases with window size. Deeper analysis of these results will be done in future work.

6.5 Detecting Change

One application of the self-consistency distribution is detecting changes in a scene over time. Given two collections of images of a scene taken at two points in time, we can compare matches (from different times) that belong to the same surface element to see if the difference in triangulated coordinates exceeds some significance level. This gives a mechanism for distinguishing changes which are significant from changes which are due to modelization uncertainty. More details, and experimental results are found in [13].

7 Previous Work in Estimating Uncertainty

Existing work on estimating uncertainty without ground truth falls into three categories: analytical, statistical, and empirical approaches.

The analytical approaches are based on the idea of error propagation [23]. When the output is obtained by optimizing a certain criterion (like a correlation measure), the shape of the optimization curve [6,15,9] or surface [1] provides estimates of the covariance through the second-order derivatives. These approaches make it possible to compare the uncertainty of different outputs given by

the same algorithm. However, it is problematic to use them to compare different algorithms.

Statistical approaches make it possible to compute the covariance given only one data sample and a black-box version of an algorithm, by repeated runs of the algorithm, and application of the law of large numbers [4].

Both of the above approaches characterize the performance of a given output only in terms of its expected variation with respect to additive white noise. In [21], the accuracy was characterized as a function of image resolution. The bootstrap methodology [3] goes further, since it makes it possible to characterize the accuracy of a given output with respect to IID noise of unknown distribution. Even if such an approach could be applied to the multiple image correspondence problem, it would characterize the performance with respect to IID sensor noise. Although this is useful for some applications, for other applications it is necessary to estimate the expected accuracy and reliability of the algorithms as viewpoint, scene domain, or other imaging conditions are varied. This is the problem we seek to address with the self-consistency methodology.

Our methodology falls into the real of empirical approaches. See [17], for a good overview of such approaches.

Szeliski [18] has recently proposed prediction error to characterize the performance of stereo and motion algorithms. Prediction error is the difference between a third real image of a scene and a synthetic image produced from the disparities and known camera parameters of the three images. This approach is especially useful when the primary use of stereo is for view interpolation, since the metric they propose directly measures how well the algorithm has interpolated a view compared to a real image of that same view. In particular, their approach does not necessarily penalize a stereo algorithm for errors in constant-intensity regions, at least for certain viewpoints. Our approach, on the other hand, attempts to characterize self-consistency for all points. Furthermore, our approach attempts to remove the effects of camera configuration is computing the measure over many observations and scenes.

Szeliski and Zabih have recently applied this approach to comparing stereo algorithms [18,20]. A comprehensive comparison of our two methodologies applied to the the same algorithms and same datasets should yield interesting insights into these two approaches.

An important item to note about our methodology is that the projection matrices for all of the images are provided and assumed to be correct (within their covariances). Thus, we assume that a match produced by the stereo algorithm always lies on the epipolar lines of the images. Consequently, a measure of how far matches lie from the epipolar line, is not relevant.

8 Conclusion and Perspectives

We have presented a general formalization of a perceptual observation called self-consistency, and have proposed a methodology based on this formalization

as a means of estimating the accuracy and reliability of point-correspondence algorithms algorithms, comparing different stereo algorithms, comparing different scoring functions, comparing window sizes, and detecting change over time. We have presented a detailed prescription for applying this methodology to multiple-image point-correspondence algorithms, without any need for ground truth or camera calibration, and have demonstrated it's utility in several experiments.

The self-consistency distribution is a very simple idea that has powerful consequences. It can be used to compare algorithms, compare scoring functions, evaluate the performance of an algorithm across different classes of scenes, tune algorithm parameters (such as window size), reliably detect changes in a scene, and so forth. All of this can be done for little manual cost beyond the precise estimation of the camera parameters and perhaps manual inspection of the output of the algorithm on a few images to identify systematic biases.

Readers of this paper are invited to visit the self-consistency web site to download an executable version of the code, documentation, and examples at http://www.ai.sri.com/sct/ described in this paper.

Finally, we believe that the general self-consistency formalism developed in Sec. 2, which examines the *self-consistency* of an algorithm across independent experimental trials of different viewpoints of a static scene, can be used to assess the accuracy and reliability of algorithms dealing with a range of computer vision problems. This could lead to algorithms that can learn to be self-consistent over a wide range of scenes without the need for external training data or "ground truth."

A The MDL Score

Given N images, let M be the number of pixels in the correlation window and let g_i^j be the image gray level of the i^{th} pixel observed in image j. For image j, the number of bits required to describe these gray levels as IID white noise can be approximated by:

$$C_j = M(\log \sigma_j + c) \tag{2}$$

where σ_j is the measured variance of the $g_{i\,1 \le i \le N}^j$ and $c = (1/2)\log(2\pi e)$.

Alternatively, these gray levels can be expressed in terms of the mean gray level $\overline{g_i}$ across images and the deviations $g_i^j - \overline{g_i}$ from this average in each individual image. The cost of describing the means, can be approximated by

$$\overline{C} = M(\log \overline{\sigma} + c) \tag{3}$$

where $\overline{\sigma}$ is the measured variance of the mean gray levels. Similarly the coding length of describing deviations from the mean is given by

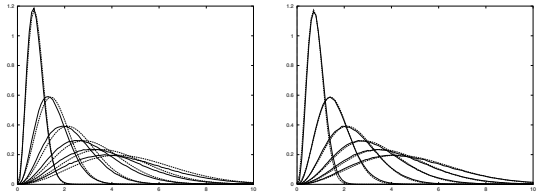
$$C_j^d = M(\log \sigma_j^d + c) \tag{4}$$

where σ_j^d is the measured variance of those deviations in image j. Note that, because we describe the mean across the images, we need only describe $N - 1$ of the C_j^d. The description of the Nth one is implicit.

The MDL score is the difference between these two coding lengths, normalized by the number of samples, that is

$$Loss = \overline{C} + \sum_{1 \le j \le N-1} C_j^d - \sum_{1 \le j \le N} C_j \quad . \tag{5}$$

When there is a good match between images, the $g_{i\,1 \le j \le N}^{j}$ have a small variance. Consequently the C_j^d should be small, $\overline{C}$ should be approximately equal to any of the C_j and $Loss$ should be negative. However, C_j can only be strongly negative if these costs are large enough, that is, if there is enough texture for a reliable match. See [10] for more details.



perturbed projections random affine projections

Fig. 3. Averaged theoretical (solid) and experimental (dashed) curves.

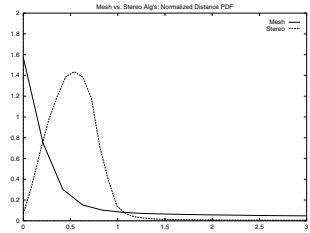
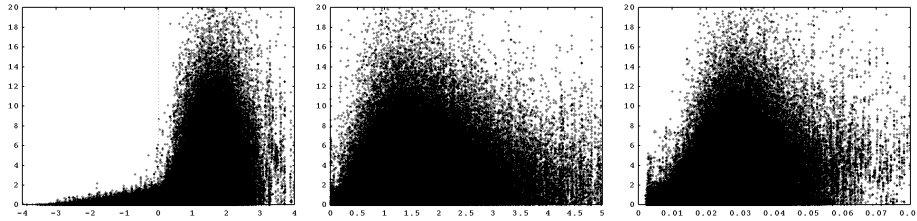


Fig. 4. Comparing two stereo algorithms (Mesh vs Stereo) using the self-consistency distributions.



MDL SSD/Grad SSD

Fig. 5. Scatter diagrams for three different scores.

Fig. 6. Three of six urban scenes used for the window comparisons. Each scene contained 4 images.

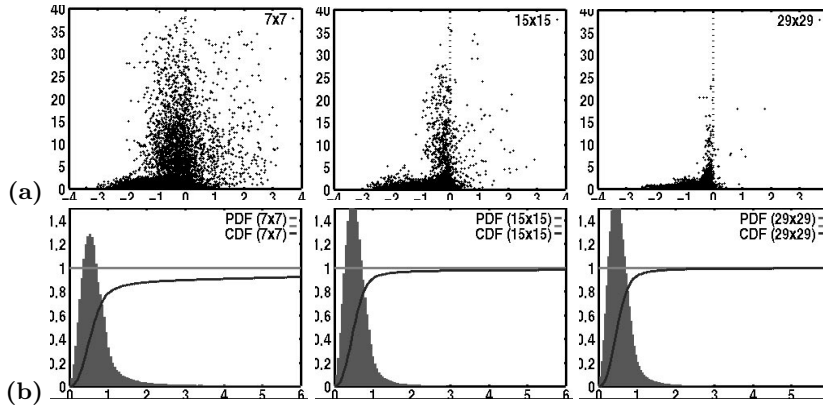


Fig. 7. Comparing three window sizes. (a) The combined self-consistency distributions of six urban scenes for window sizes 7×7, 15×15, and 29×29. (b) The scatter diagrams for the MDL score for these urban scenes.

References

1. P. Anandan. A computational framework and an algorithm for the measurement of visual motion. *IJCV*, 2:283–310, 1989.
2. N. Ayache and F. Lustman. Fast and reliable passive trinocular stereovision. In *ICCV*, pages 422–427, 1987.
3. K. Cho, P. Meer, and J. Cabrera. Performance assessment through bootstrap. *PAMI*, 19(11):1185–1198, November 1997.
4. G. Csurka, C. Zeller, Z. Zhang, and O. Faugeras. Characterizing the uncertainty of the fundamental matrix. *CVGIP-IU*, 1996.
5. O. Faugeras, P. Fua, B. Hotz, R. Ma, L. Robert, M. Thonnat, and Z. Zhang. Quantitative and Qualitative Comparison of some Area and Feature-Based Stereo Algorithms. In W. Forstner and S. Ruwiedel, editors, *International Workshop on Robust Computer Vision: Quality of Vision Algorithms*, pages 1–26, Karlsruhe, Germany, March 1992.
6. W. Forstner. On the geometric precision of digital correlation. In *International archives of photogrammetry and remote sensing*, volume 24-III, pages 176–189, Helsinki, 1982.

7. P. Fua. Combining Stereo and Monocular Information to Compute Dense Depth Maps that Pres erve Depth Discontinuities. In *Int. Joint Conf. on AI*, pages 1292–1298, Sydney, Australia, August 1991.
8. P. Fua and Y. G. Leclerc. Object-Centered Surface Reconstruction: Combining Multi-Image Stereo and Shading. *IJCV*, 16:35–56, 1995.
9. T. Kanade and M. Okutomi. A stereo matching algorithm with an adaptive window: Theory and experiment. *PAMI*, 16(9):920–932, 1994.
10. Y. Leclerc, Q.-T. Luong, and P. Fua. A framework for detecting changes in terrain. In *Proc. Image Understanding Workshop*, pages 621–630, Monterey, CA, 1998.
11. Y. Leclerc, Q.-T. Luong, and P. Fua. Self-consistency: a novel approach to characterizing the accuracy and reliability of point correspondence algorithms. In *Proc. Image Understanding Workshop*, pages 793–807, Monterey, CA, 1998.
12. Y. Leclerc, Q.-T. Luong, and P. Fua. Characterizing the performance of multiple-image point-correspondence algorithms using self-consistency. In *Proceedings of the Vision Algorithms: Theory and Practice Workshop (ICCV99)*, Corfu, Greece, September 1999.
13. Y. Leclerc, Q.-T. Luong, and P. Fua. Detecting changes in 3-d shape using self-consistency. In *Proc. Conference on Computer Vision and Pattern Recognition*, Hilton Head, South Carolina, June 1999.
14. Y. Leclerc, Q.-T. Luong, and P. Fua. Self-consistency: a novel approach to characterizing the accuracy and reliability of point correspondence algorithms. In *Proceedings of the One-day Workshop on Performance Characterisation and Benchmarking of Vision Systems*, Las Palmas de Gran Canaria, Canary Islands, Spain, 1999.
15. L. Matthies. Stereo vision for planetary rovers: Stochastic modeling to near real-time implementation. *IJCV*, 8(1):71–91, July 1992.
16. R. Mohr, F. Veillon, and L. Quan. Relative 3d reconstruction using multiple uncalibrated images. In *CVPR*, pages 543–548, NYC, 1993.
17. W. Førstner. Diagnostics and performance evaluation in computer vision. In *Performance versus Methodology in Computer Vision, NSF/ARPA Workshop*, Seattle, WA, 1994.
18. R. Szeliski. Prediction error as a quality metric for motion and stereo. In *ICCV99*, Corfu, Greece, September 1999.
19. R. Szeliski and S.B. Kang. Recovering 3d shape and motion from image streams using nonlinear least squares. *JVCIR*, pages 10–28, 1994.
20. R. Szeliski and R. Zabih. An experimental comparison of stereo algorithms. In *Proceedings of the Vision Algorithms: Theory and Practice Workshop (ICCV99)*, Corfu, Greece, September 1999.
21. P.H.S. Torr and A. Zissermann. Performance characterization of fundamental matrix estimation under image degradation. *mva*, 9:321–333, 1997.
22. M. Yachida, Y. Kitamura, and M. Kimachi. Trinocular vision: New approach for correspondence problem. In *ICPR*, pages 1041–1044, 1986.
23. S. Yi, R.M. Haralick, and L.G. Shapiro. Error propagation in machine vision. *MVA*, 7:93–114, 1994.