Characterizing the Performance of Multiple-image Point-correspondence Algorithms using Self-Consistency

Yvan G. Leclerc, Q.-Tuan Luong and P. Fua *
Artificial Intelligence Center, SRI International, Menlo Park, CA
leclerc,luong@ai.sri.com
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

One way of characterizing the performance of a stereo algorith is to compare its matches against "ground truth." If sufficient quantities of accurate ground truth were available, estimating the distribution of errors over many image pairs of many scenes (within a class of scenes) would be relatively straightforward. This distribution could then be used to predict the accuracy of matches in new images. Unfortunately, acquiring ground truth for any scene is an expensive and problematic proposition at best.

Instead, we propose to estimate a related distribution, which can be derived automatically from the matches of many image pairs of many scenes, assuming only that the projection matrices for the image pairs (and their covariances) have been correctly estimated, up to an unknown projective transformation.

The related *self-consistency* distribution, as we call it, is the distribution of the normalized difference between triangulations of matches obtained when one image is fixed and the projection matrix of the second image is changed, averaged over all matches, many images, and many scenes.

*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.

Intuitively, a perfect stereo algorithm is one for which the triangulations are invariant to changes in the second image, that is, one for which the mean and variance of the self-consistency distribution are zero. The extent to which the distribution deviates from this is a measure of the accuracy of the algorithm. Of course, a stereo algorithm that is perfectly self-consistent in this sense can still have systematic biases. We will discuss such biases in a more complete version of this paper.

Although the self-consistency distribution is an important global characterization of a stereo algorithm, it would be better if we could refine the predicted accuracy of individual matches given just a pair of images. We propose to do this by estimating the self-consistency distribution as a function of some type of "score" (such as sum of squared difference) that can be computed for each match using only the image pair. Conditionalizing the self-consistency distribution like this not only allows us to better predict the accuracy of individual matches, it also allows us to compare different scoring functions to see which one best correlates with the self-consistency of matches.

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, 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.

In the remainder of this paper we will describe the algorithm we use for estimating the self-consistency distribution of general n-point correspondence algorithms given prior collections of images. This includes the development of a method to normalize the triangulation (or reprojection) differences due to the inherent errors arising from the nominal accuracy of the matches, the projection matrices, and their covariances. Monte Carlo experiments are presented to justify the normalization method. We will then show how the self-consistency distribution can be used to to compare stereo algorithms and scoring functions.

2 Previous Work in Estimating Uncertainty

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

The analytical approaches are based on the idea of error propagation [17]. When the output is obtained by optimizing a certain criterion (like a correlation measure), the shape of the optimization curve [5, 12, 8] 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 [15], 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.

3 The Self-Consistency Distribution

To understand the self-consistency distribution, consider the following thought experiment. Consider a match (m_A, m_B) derived from two images, A and B. Now, fix image A and m_A, vary the projection matrix of the second image to produce image B', and apply the stereo algorithm to images A and B' producing the match $(m_A, m_{B'})$. Because the coordinates of the matches are identical in image A, the two matches should triangulate to the same point in space, within the expected error induced by the nominal precision of the matches, the projection matrices, and their covariances.

The distribution of the difference between the triangulations of the two matches, after suitable normalization, averaged over all matches derived from many image pairs of many scenes, is what we call the self-consistency distribution for that algorithm. We will discuss in detail the normalization later in the paper.

When the triangulations are equal to within the expected error, the matches are said to be consistent. When an an algorithm produces matches that are always consistent in this sense, we say that the algorithm is self-consistent.

Note that the self-consistency distribution is directly applicable to change detection by using the $x\%$ confidence interval for a match. The $x\%$ confidence interval is the largest normalized distance that two matches (with the same coordinate in one image) can have $x\%$ of the time. Thus, two matches derived from images of a scene taken at different times can then be compared against this confidence interval to see if the scene has changed over time at that point (see [9]).

3.1 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.

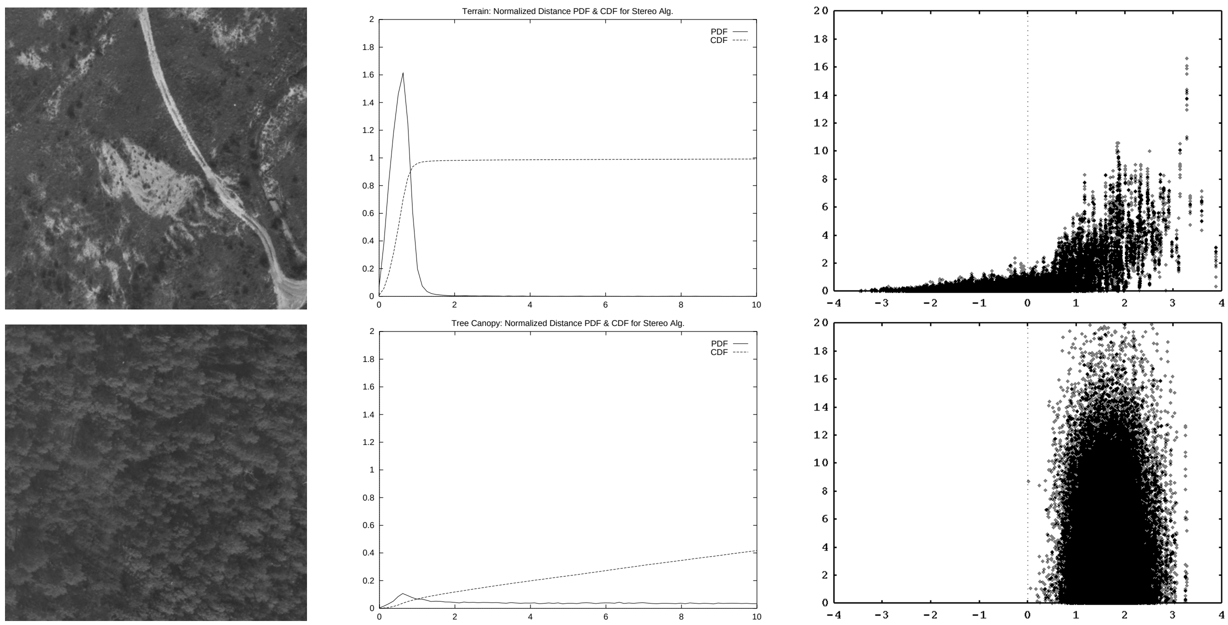
In the thought experiment above, we first found a match, fixed the coordinate of the match in one image, varied the camera parameter of the second image to get a second match, and then computed the normalized distance between their triangulations.

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]. 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 correspond to one instance of the thought experiment. 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 [16, 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. The histogram of these normalized differences, computed

[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.



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

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

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 Section 4, to the case of multiple matches.

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

3.2 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[6].. 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 neighbours. 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.

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.

3.3 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 below). This is illustrated in Figures 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 Section 5.3).

4 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 [13, 14] and does not require camera calibration. The Euclidean distance is not invariant to the choice of projective coordinates, but this dependance can often be reduced by using the normalization described below. Another way to do so, which actually cancels the dependance on the choice of projective coordinates, is to compute the difference between the reprojections instead of the triangulations, as described in more detail in [11]. This, however, does not cancel the dependance on the relative geometry of the cameras.

4.1 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*: $(\mathbf{M}_1 - \mathbf{M}_2)^T(\Lambda_1 + \Lambda_2)^{-1}(\mathbf{M}_1 - \mathbf{M}_2)$.

4.2 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 $\mathbf{\Lambda_x}$, 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 $\mathbf{J}_f(\mathbf{x}_0)\mathbf{\Lambda_x}\mathbf{J}_f(\mathbf{x}_0)^T$, 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 $[x_1, y_1, x_2, y_2, \dots x_n, y_n]$ 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 $w_i, i = 1 \dots n, w = x, y$. We also assume that the errors at each pixel are independent, uniform, and isotropic. The covariance matrix $\mathbf{\Lambda_x}$ 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.

5 Experiments

5.1 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.4 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 difference 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:

$$\chi_3^2 = \frac{1}{\sqrt{2\pi}}\sqrt{x}e^{-x/2}$$

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.

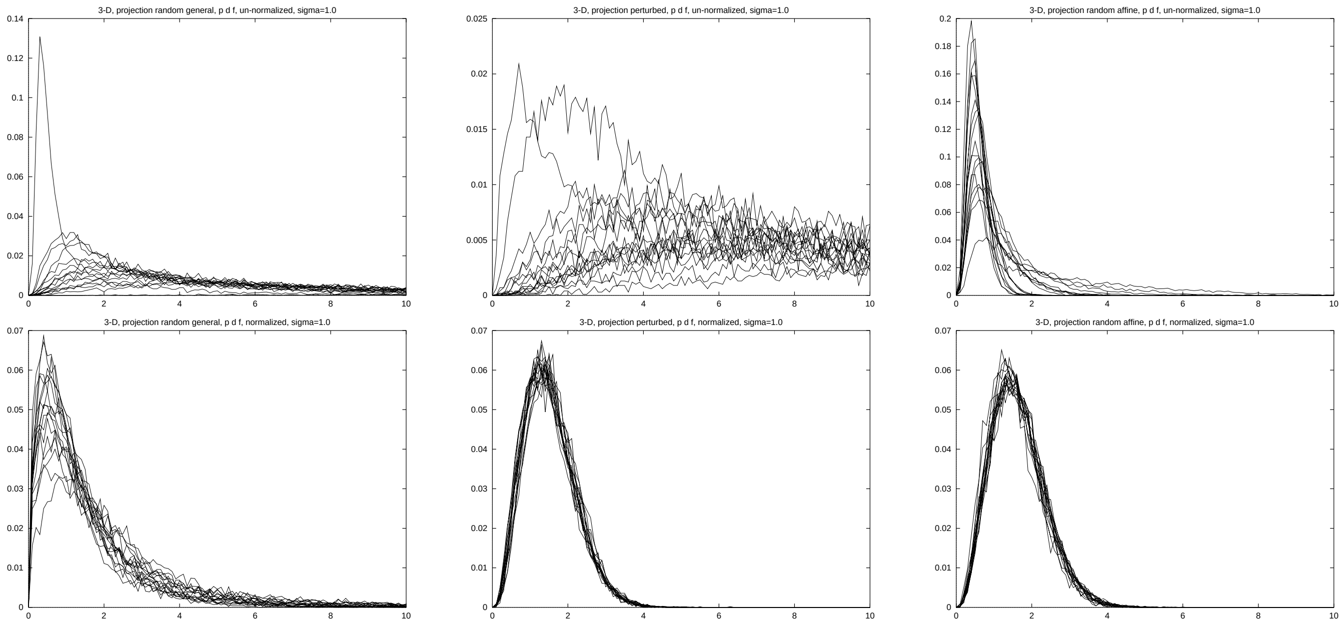
Comparison of the experimental and theoretical distributions Using the Mahalanobis distance, we then averaged the density curves within each set of experiments, and tried to fit the model described in Eq. 1 to the resulting curves, for six different values of the standard deviation, $\sigma = 0.5, 1, 1.5, 2, 2.5, 3$. 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.

5.2 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, [11, 10], 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



random general projections | perturbed projections | random affine projections

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

mesh algorithm [7], 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.

5.3 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) [5]. 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 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.

Another way that we can compare the scores is with a measure we call the *efficiency* of the scoring function. This is the number of match pairs for which the confidence interval is below some value d divided by the total number of match pairs having normalized distances less than d. Intuitively, the efficiency represents how well the scoring function can predict that match pairs will have normalized differences below some value given just the score. An ideal score would have an efficiency of 1 for all values of d.

The 99% efficiency of all three scores is illustrated in Figure 6. Note that, overall, the MDL score is somewhat more efficient than the SSD/GRAD score, both of which are significantly more efficient than the SSD score.

Our previous publication [11] compared two scoring function by comparing their cumulative distributions in two different scenes. Here we compare scores using the merged data from many scenes using confidence-interval and efficiency graphs, again providing a more comprehensive comparison than was possible before.

6 Conclusion and Perspectives

We have proposed the self-consistency methodology as a means of estimating the accuracy and reliability of point-correspondence algorithms algorithms, comparing different algorithms, and comparing different scoring functions. 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 two 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, 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 core idea of our methodology, which examines the *self-consistency* of an algorithm across independent experimental trials, 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 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:

$$C_j = M(\log \sigma_j + c) \quad (2)$$

where σ_j is the measured variance of the $g_i^j{}_{1 \le i \le N}$ 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

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

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

$$C_j^d = M(\log \sigma_j^d + c) \quad (4)$$

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

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

When there is a good match between images, the $g_i^j{}_{1 \le j \le N}$ 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 [9] for more details.

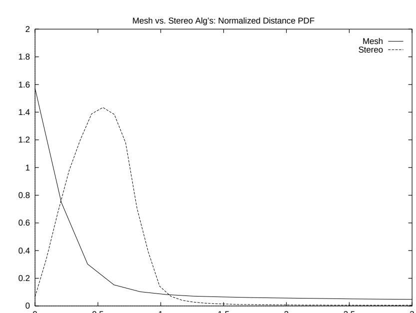
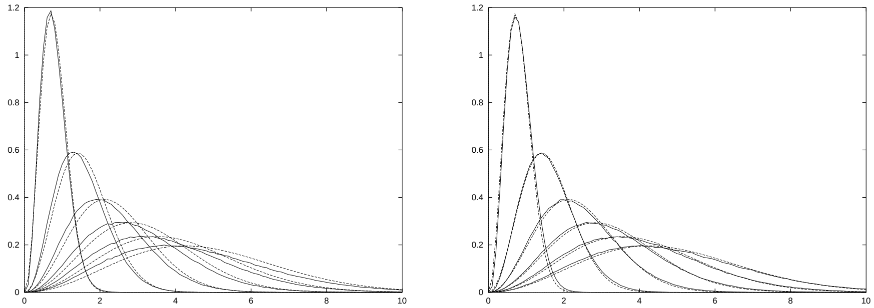


Figure 4: Comparing two stereo algorithms (Mesh vs Stereo) using the self-consistency distributions.

References

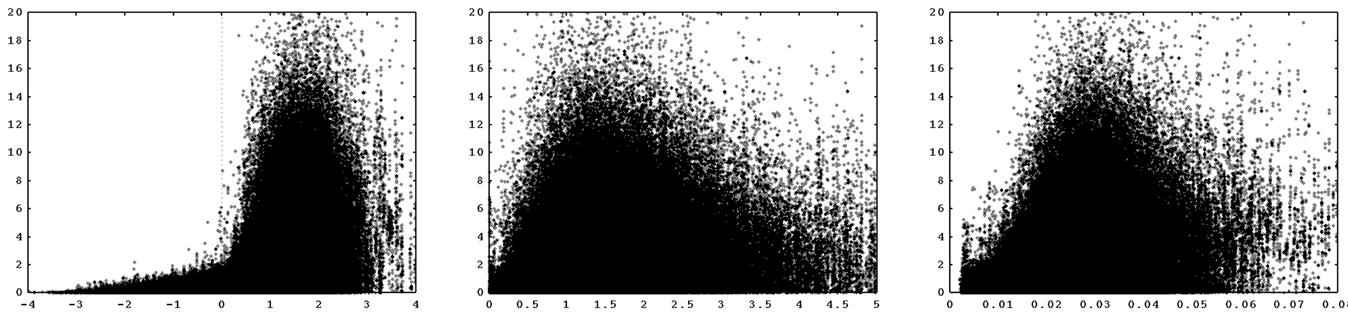
[1] P. Anandan. A computational framework and an algorithm for the measurement of visual motion. *IJCV*, 2:283–310, 1989.

[2] N. Ayache and F. Lustman. Fast and reliable passive trinocular stereovision. In *ICCV*, pages 422–427, 1987.



perturbed projections random affine projections

Figure 3: Averaged theoretical (solid) and experimental (dashed) curves.



MDL SSD/Grad SSD

Figure 5: Scatter diagrams for three different scores.

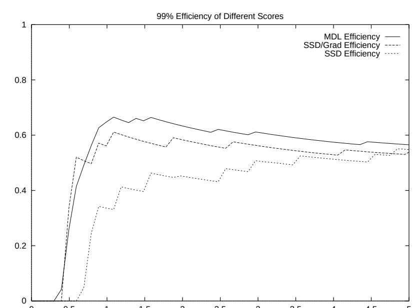


Figure 6: Comparing three scoring schemes (MDL vs. SSD/GRAD vs. SSD) using the efficiency measure.

[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] 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.

[6] P. Fua. Combining Stereo and Monocular Information to Compute Dense Depth Maps that Preserve Depth Discontinuities. In *Int. Joint Conf. on AI*, pages 1292–1298, Sydney, Australia, August 1991.

[7] P. Fua and Y. G. Leclerc. Object-Centered Surface Reconstruction: Combining Multi-Image Stereo and Shading. *IJCV*, 16:35–56, 1995.

[8] T. Kanade and M. Okutomi. A stereo matching algorithm with an adaptive window: Theory and experiment. *PAMI*, 16(9):920–932, 1994.

[9] 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.

[10] 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.

[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 *Proceedings of the One-day Workshop on Performance Characterisation and Benchmarking of Vision Systems*, Las Palmas de Gran Canaria, Canary Islands, Spain, 1999.

[12] L. Matthies. Stereo vision for planetary rovers: Stochastic modeling to near real-time implementation. *IJCV*, 8(1):71–91, July 1992.

[13] R. Mohr, F. Veillon, and L. Quan. Relative 3d reconstruction using multiple uncalibrated images. In *CVPR*, pages 543–548, NYC, 1993.

[14] R. Szeliski and S.B. Kang. Recovering 3d shape and motion from image streams using nonlinear least squares. *JVCIR*, pages 10–28, 1994.

[15] P.H.S. Torr and A. Zissermann. Performance characterization of fundamental matrix estimation under image degradation. *mva*, 9:321–333, 1997.

[16] M. Yachida, Y. Kitamura, and M. Kimachi. Trinocular vision: New approach for correspondence problem. In *ICPR*, pages 1041–1044, 1986.

[17] S. Yi, R.M. Haralick, and L.G. Shapiro. Error propagation in machine vision. *MVA*, 7:93–114, 1994.