Reprinted (with corrections) from: *Proceedings: 1998 Image Understanding Workshop,* Monterey (CA): Morgan Kaufmann. 1998.

Self-Consistency: A Novel Approach to Characterizing the Accuracy and Reliability of Point Correspondence Algorithms

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

We propose a methodology that estimates the accuracy and reliability of the results of any multiple-image point correspondence algorithm, without the need for ground truth or camera calibration. The key concept behind our methodology is what we call the *self-consistency* of an algorithm across independent experimental trials (independent applications of the algorithm to subsets of images in a given collection of images). We describe precisely the *self-consistency distributions* and how to compute them. We then explain why we conjecture that they provide a reasonable approximation of the absolute error distributions. Experiments illustrate the usefulness of these distributions at ranking the quality of algorithms and the predictive power of scoring schemes.

1 Introduction

The point correspondence problem—finding points in two or more images that correspond to the same point in the world—is critical in computer vision. Although there have been many papers written about the accuracy and reliability of point correspondence algorithms, they have primarily been concerned with a limited case: estimating the expected variation in the image coordinates of the corresponding points as a function of the amount of white noise added to the images (see Sec. 2).

Unfortunately, such estimates tell us very little about the expected accuracy and reliability of the algorithms as viewpoint, scene domain, or other imaging conditions are varied. This is the problem we address here.

In principle, one could estimate the accuracy and reliability of an algorithm by collecting statistics about the difference between the correspondences produced by the algorithm and previously established "ground truth." By doing this over a sufficiently large collection of images (within a given class of scenes and imaging conditions), one could, in principle, arrive at a fairly accurate estimate of the accuracy and reliability of the algorithm when applied to new images of scenes within the given class. Unfortunately, collecting sufficiently accurate "ground truth" for a large collection of images is extremely difficult.

We propose here a methodology that can estimate at least the expected upper bound on the accuracy and reliability of an algorithm in the sense defined above, but without using "ground truth." Under some fairly general circumstances, this methodology makes it possible to derive estimates of the actual reliability and accuracy of the algorithm, and not just

*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.

their upper bounds.

Specifically, we address the following two questions: "How often do the n-tuples of image points produced by a point correspondence algorithms actually correspond to the same point in the world?" and "What is the expected accuracy of the coordinates of a nominally correct n-tuple?". When an algorithm associates a score with each n-tuple, the latter question can also be stated in the following form: "What is the expected accuracy of the coordinates of a nominally correct n-tuple for a given score?"

The key concept behind our methodology is what we call the *self-consistency* of an algorithm across independent experimental trials (independent applications of the algorithm to subsets of images in a given collection of images). As described in Sec. 3, self-consistency is a necessary condition for a point correspondence algorithm to be correct; it extends conceptually the principle of the trinocular stereo constraint [Yachida *et al.*, 1986, Ayache and Lustman, 1987]. It can be measured using only the outputs of the algorithm and the *projectively coherent* projection matrices associated with the images (Sec. 4), and does not require camera calibration.

We define in Sec. 5 two different self-consistency distributions, called the *bounding distribution* and the *difference distribution*. When applied to a sufficiently broad set of independent experiments, such as all pairs of images from a collection of images of a given scene, the statistics of the degree to which an algorithm is self-consistent for the first distribution are upper bounds on the expected accuracy and reliability of the algorithm. With the second distribution, under some hypothesis it is possible to derive not only an upper bound, but also an estimate of the expected accuracy and reliability of the algorithm. This is the first method that we are aware of that allows us to estimate the accuracy of point correspondence algorithms without comparison to either "ground truth" or evaluation by humans.

In the remainder of this paper, we will present a brief summary of previous work in estimating the uncertainty in point correspondence algorithms, an overview of our methodology, a discussion of projections and point correspondence algorithms. We then describe the two types of self-consistency distributions, and explain how they are normalized. We then present experimental results which illustrate the predictive power of the self-consistency distributions.

2 Previous Work in Estimating Uncertainty

Existing work on estimating uncertainty of 3–D reconstruction falls into two categories: analytical approaches and statistical approaches.

The analytical approaches are based on the idea of error propagation ([Yi *et al.*, 1994] gives a general account). In the case of stereo correspondence, as well as for some optical flow algorithms, the correspondence is obtained by optimizing a certain criterion (in general based on image correlation). In this case the shape of the optimization curve or surface provides estimates of the covariance. This has been justified in a probabilistic framework by [Forstner, 1982, Matthies, 1992, Kanade and Okutomi, 1994] in the case of fronto-parallel surfaces, for a one-dimensional correlation curve. In the full two-dimensional case, this shape is characterized by second-order derivatives of the optimization surface [Anandan, 1989]. One of the ways to do that is to compute the Hessian (2nd derivative) matrix at the peak, and then the eigenvectors and eigenvalues of this matrix. The eigenvectors gives the directions of principal curvature and the eigenvalues tells how curved it is. These approaches make it possible to compare the uncertainty of different correspondences given by the same algorithm. However, it is problematic to use them to compare different algorithms, especially since the implementation is far from being trivial because it requires the explicit computation of second order derivatives, a task which is quite difficult if instead of point-to-point correlation a more complicated optimization scheme is used.

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 [Csurka *et al.*, 1996] or of the theory of the bootstrap [Cabrera and Meer, 1996].

Both of the above approaches characterize the accuracy of a given point correspondence in terms of its expected variation with respect to additive white noise. This is because the standard approach is to use additive white noise as an approximate model of the difference between corresponding image "windows." But since the primary difference between corresponding image windows is due to change in viewpoint rather than to sensor noise, the white noise model can lead to severe inaccuracies in estimating uncertainty.

Our approach, on the other hand, is to characterize the accuracy of an algorithm by performing independent experiments over many real images in which the scene and viewpoint are varied, rather than (implicitly or explicitly) performing experiments over many images in which additive white noise is varied. Thus, we expect to be able to estimate uncertainty more accurately.

3 Overview and Motivation for Our Methodology

There are three basic observations behind our methodology for estimating the accuracy and reliability of any point correspondence algorithm. These observations, and the brief description of the methodology provided here, will be expanded upon in the following sections.

In the following, an "n-tuple" refers to a set of two or more image points, one point per image, found by a point correspondence algorithm. All of the image points in such an n-tuple are supposed to correspond to the same point in the world. In general, an n-tuple can contain fewer points than the number of images used by the algorithm.

The first observation is that, when two or more n-tuples all share a common point in one image, then they must all represent the same point in space. We shall call such a set of n-tuples a *common-point n-tuple set.* A common-point n-tuple set shall be called *self-consistent* if it does indeed represent the same point in space.

The second observation is that we can verify that a common-point n-tuple set is consistent with the hypothesis that they correspond to a single point in space by using only the projection matrices $\mathbf{P}_i$; even when they are only *projectively coherent* (see Sec. 4). That is, we can verify the *self-consistency* of a common-point n-tuple set without the prior establishment of "ground truth" for the scene, and without need for camera calibration. In the most general case of non-Euclidean projection matrices, this is done by comparing the image coordinate of a point in one n-tuple against the predicted coordinates of all other pairs of points in the other n-tuples using the projection matrices. When the predicted coordinates are all equal to the coordinate of that point, and this is true for all the points in the set, then the common-point n-tuple set is self-consistent.

Clearly, when the projection matrices are exact, a necessary condition for a point correspondence algorithm to be correct is that all common-point n-tuple sets must be self-consistent. This leads to the third observation.

The third observation is that applying the correspondence algorithm independently to subsets of images within a collection provides independent evidence of the correctness of the algorithm. For example, one can apply the algorithm independently to subsets of images of a collection of images of a scene, gather together the common-point n-tuple sets, and then compute the percentage of common-point n-tuples that are self-consistent.

We can generalize the above notion of self-consistency by allowing for a certain amount of error between the predicted coordinates and the n-tuple coordinates. This can either be done in image space, or, when the projection matrices are Euclidean, in Euclidean space, as explained in Sec. 5. By expressing the self-consistency counts as a function of the amount of error, we create the *self-consistency distributions.* Unlike the exact case, there is not a unique way to define self-consistency up to a certain amount of error. We propose two different self-consistency distributions.

The first of them, the *bounding distribution* can be proved to be an upper bound on accuracy, in a sense to be made precise in the next section. We conjecture that *self-consistency is not only a necessary condition for an algorithm to be correct, but that it is also a sufficient condition, at least for certain classes of scenes and imaging conditions.* Thus, if our conjecture is correct, the percentage of self-consistent n-tuples would not just be an upper bound on the percentage of correct n-tuples, but would also be a tight estimate on that percentage. Similarly, the self-consistency distributions would not just be upper bounds on the correct distributions, but would be good estimates of the correct distributions. The second distribution, the *difference distribution*, is such that under certain circumstances, the absolute accuracy distribution can be derived from it. While so far we have established this result only for a special case, we suspect that these circumstances are fairly broad and that we will eventually be able to derive an even tighter estimate of absolute accuracy from the difference distribution. This is the first method that we are aware of that allows us to estimate the accuracy of point correspondence algorithms without comparison to either "ground truth" or evaluation by humans.

To obtain the self-consistency distributions, we first apply the point correspondence algorithm to all pairs of images/projection matrices in a test suite of images, as defined later in Sec. 5.4. If the algorithm can be applied to n-tuples of images, then we apply the algorithm to all subsets of images instead of just all pairs.

Second we group together all n-tuples of correspondences that share a common image point (i.e., that are within a disk of diameter ϵ_0). If the set of common-point n-tuples is too small relative to the number of remaining n-tuples, then the methodology is not applicable to the algorithm. However, in the case of algorithms which produce dense depth maps, it is always possible to create sets of n-tuples with common image points by sampling one of the views. Algorithms which produce sparse correspondences rely on operators to detect distinguished points of interest. Such an algorithm would be suspect, in our opinion, if the points of interest detected would not overlap significantly. It may be possible to adjust the algorithm so as to ensure that a large number of common-point n-tuples are produced.

Eventually, the self-consistency distributions are computed using the definitions in Sec. 5, where a few algorithmic details are also provided.

4 Projection Matrices and Point Correspondence Algorithms

In this section, we describe in more detail the two input components of our methodology: the projection matrices which we use to evaluate the self-consistency statistics, and the point correspondence algorithms that we are trying to assess.

4.1 Projections in the calibrated and uncalibrated case

The 3×4 projection matrix $\mathbf{P}_i$ transforms a point M in projective space to a point m in image I_i. Importantly, as we shall see, the projection matrices can be used to predict the coordinate of a point in one image given the corresponding points in two other images.

In the vast majority of the three-dimensional reconstruction literature, the projection matrices are supposed to be known up to a Euclidean transformation, representing the arbitrary choice of the Euclidean world coordinate system. Such matrices can be obtained from standard resection and block-adjustment algorithms when both the internal orientation of a camera and the 3–D coordinates of a relatively small number of points is known. These points are often selected manually or semi-automatically. The knowledge of a sufficiently large number of 3–D points make it also possible to determine the internal orientation by calibration.

When the internal orientation of a camera is unknown and/or it is not possible to determine the Cartesian coordinates of selected points, it is still possible to compute *projectively coherent* projection matrices using just selected correspondences [Mohr *et al.*, 1993, Szeliski and Kang, 1994]. That is, it is possible to compute a set of projection matrices $\mathbf{P}_i$ that are the product of the unknown Euclidean projection matrices $\tilde{\mathbf{P}}_i$ and a fixed, but unknown, general non-singular transformation of projective space represented by a 4×4 matrix $\mathbf{H}$. The set of matrices $\mathbf{P}_i$ is projectively coherent in the sense that they allow a description of 3–D space which is exact up to the choice of a projective coordinate system in the world. Note that one can also compute the $\mathbf{P}_i$ directly from the complete set of fundamental matrices $\mathbf{F}_{ij}$, or equivalently, the complete set of epipolar constraints [Luong and Viéville, 1996], and vice-versa. In particular, the set of projectively coherent projection matrices is sufficient to provide the multi-view epipolar geometry which is needed for many point correspondence algorithms.

The important property of the projectively coherent projection matrices that we will use here is that we can use them to predict the coordinate of a point in one image given the coordinates of that same point in at least two other images. The idea is to reconstruct a 3–D point from the corresponding points, using standard triangulation techniques. If the projection matrices are not Euclidean, the projective coordinates of this point will be related to Euclidean coordinates by an unknown transformation, but since this transformation is fixed, it does not matter because by reprojecting this point into a third image with a coherent projection matrix, the unknown transformation is canceled, and the result of the triangulation-reprojection is the same as the one obtained with the Euclidean projection matrices. An alternative way to do the prediction would be to compute the trifocal tensor of the two source views and the target view from the projection matrices [Hartley, 1995, Shashua, 1994].

4.2 Point correspondence algorithms

For this paper, a point correspondence algorithm takes as input a set of N images G_i and associated projectively coherent projection matrices $\mathbf{P}_i$, $i = 1 \ldots N, N > 2$.

The output of a point correspondence algorithm is a set of n-tuples U_j of image coordinates and, optionally, a score S_j, $j = 1 \ldots M$. Each n-tuple U_j is supposed to represent the projected image coordinates of a single point in the world. The number of image coordinates in an n-tuple may vary from point to point, but must be greater than one and less than or equal to N.

It is assumed that any n-tuple returned by the correspondence algorithm can be triangulated and reprojected into the images of the n-tuple with an error no greater than the *nominal accuracy* ϵ_0 of the algorithm (typically 1 pixel). This nominal accuracy is used to define what is meant by a set of common points, namely all points in an image that lie within a disk of diameter ϵ_0. A uniform nominal accuracy for all images may not be possible when the camera viewpoints are very different. We will address this issue in a more comprehensive version of this article.

The optional score vector S_j represents a measure of the algorithm's confidence in the corresponding n-tuple. We can collect the statistics as a function of this score by only comparing common-pair n-tuples that have approximately the same score. The extent to which our estimates of accuracy and reliability correlate with the score will be a measure of the predictive power of the score.

5 Accuracy, Reliability, and the Self-consistency Distributions

5.1 The accuracy distributions

One can define the accuracy and reliability in terms of the relationship between points in the world and the putative corresponding n-tuple of image points:

- **Reliability** The probability that a point in an n-tuple refers to the putative corresponding point in the world.
- **2–D Image Accuracy Distribution** The probability that an image point in an n-tuple is within ϵ of the projected coordinate of the true world point, as a function of ϵ.
- **3–D Euclidean Accuracy Distribution** (When Euclidean projective matrices are available.) The probability that the triangulated world coordinate of any pair of image points in an n-tuple is within ϵ of the true world coordinate of the corresponding world point, as a function of ϵ.

Analytically, for a set $\mathcal{S}_i$ of n_i common-point n-tuples, the accuracy is defined as:

$$\begin{aligned} A(\mathcal{S}_i, \epsilon) &= P(d(z_i^j, z_i^*) < \epsilon, z_i^j \in \mathcal{S}_i) \\ &\approx \frac{1}{n_i} \sum_{z_i^j \in \mathcal{S}_i} h(d(z_i^j, z_i^*) < \epsilon) \quad (1) \end{aligned}$$

where $h(\mathcal{P})$ is the function which is 1 if proposition $\mathcal{P}$ is true, 0 otherwise, and z_i^* the true value. The accuracy distribution is empirically obtained by taking the average of the terms previously defined, weighted by the cardinal n_i of each set.

As mentioned in the introduction, computing such a distribution directly is virtually impossible because of the difficulty of establishing accurate "ground truth" for a large number of images.

5.2 The bounding self-consistency distribution

Let us define a first measure of the self-consistency of sets of common-point n-tuples, which follows closely the previous presentation. This measure is called *bounding* because we will see that it represents actually an upper bound on accuracy. We need to define two new terms for this:

First, we define the *maximally enclosing ϵ–disk* for a common-point n-tuple set as the disk of diameter ϵ that contains the greatest number of reprojected points in a given image.

Second, we define the *maximally enclosing ϵ–ball* for a given common-point n-tuple set as the ball of diameter ϵ that contains the greatest number of triangulated points for the given set (only when Euclidean projective matrices are available).

With the above, we can define the following.

- **2–D Image Bounding Self-Consistency Distribution** The probability that an image point in a common-point n-tuple set is within the maximally enclosing ϵ–disk for that set, as a function of ϵ.
- **3–D Euclidean Bounding Self-Consistency Distribution** (When Euclidean projective matrices are available.) The probability that an n-tuple in a common-point n-tuple set lies within the maximally enclosing ϵ–ball for that set, as a function of ϵ.

This makes it possible to use the above estimates of reliability:

- **2–D Image Self-Consistency Reliability** The value of the 2–D image self-consistency accuracy distribution for $\epsilon = \epsilon_0$.
- **3–D Euclidean Self-Consistency Reliability** The value of the 3–D Euclidean self-consistency distribution for $\epsilon = \epsilon_0$.

Analytically, for a set $\mathcal{S}_i$ of n_i common point n-tuples, the bounding self-consistency count is given by:

$$SC_1(\mathcal{S}_i, \epsilon) = \frac{1}{n_i} \max_{z_i^0} \sum_{z_i^j \in \mathcal{S}_i} h(d(x_i^j, z_i^0) < \epsilon) \quad (2)$$

Like previously, the bounding self-consistency distribution is obtained by taking the average of the terms previously defined, weighted by the cardinal n_i of each set. By comparing the definitions in Eq. (1) and in Eq. (2), it is clear that $A(\mathcal{S}_i, \epsilon) \leq SC_1(\mathcal{S}_i, \epsilon)$, since z_i^0 can be chosen to be z_i^*. In other worlds, the bounding self-consistency distribution provides us indeed with an upper bound on the accuracy distribution.

However, one of the drawbacks of this definition is that the computation of the distribution is intensive, since it implies the determination of a maximum. The naive implementation involves trying all the subsets of $\mathcal{S}_i$ to see if they lie in a maximally enclosing ϵ-disk, which is exponential in the cardinal of $\mathcal{S}_i$. We have improved on this implementation by a bucketing technique. Instead of disks and balls, we consider squares and cubes. It can be shown that to find the maximum number of points lying in a ϵ-cube it is sufficient to check the n_i^3 cubes which have a vertex located on a point of the grid defined by the Cartesian product of the n_i points of $\mathcal{S}_i$. In the 2D case, due to the necessity to identify points, the complexity does not drop, and also remains a polynomial expression of order 3. Since this is per value of ϵ and per set of common-point n-tuple, the resulting computation is quite expensive. This has prompted us to introduce another definition of a self-consistency distribution, which turns out to have other distinct advantages.

5.3 The difference self-consistency distribution

We introduce the simpler definitions, which unlike the previous one, is quite easy to compute:

- **2–D Image Difference Self-Consistency Distribution** The probability that two image points in a common-point n-tuple set are within ϵ of each other.
- **3–D Euclidean Difference Self-Consistency Distribution** (When Euclidean projective matrices are available.) The probability that two reconstructed points in a common-point n-tuple set are within ϵ of each other.

Analytically, for a set $\mathcal{S}_i$ of n_i common point n-tuples, this gives

$$\begin{aligned} SC_2(\mathcal{S}_i, \epsilon) &= P(d(z_i^j, z_i^k) < \epsilon, (z_i^j, z_i^k) \in \mathcal{S}_i \times \mathcal{S}_i) \\ &\approx \frac{1}{n_i^2} \sum_{(z_i^j, z_i^k) \in \mathcal{S}_i \times \mathcal{S}_i} h(d(z_i^j, z_i^k) < \epsilon) \end{aligned} \quad (3)$$

Like previously, the *difference self-consistency distribution* is obtained by taking the average of the terms previously defined, weighted by the square cardinal n_i^2 of each set. Estimates of reliability are also derived similarly. Like with the bounding distribution, there is a relation between $A(\mathcal{S}_i, \epsilon)$ and $SC_2(\mathcal{S}_i, \epsilon)$, which is more difficult to express but might yield a tighter estimate. We notice in the 1D case that $z_i^j - z_i^k = (z_i^j - z_i^*) - (z_i^k - z_i^*)$. Thus SC_2 is the distribution of the differences between the random variables of the accuracy distribution. When they are identical, the density function of SC_2 is the convolution of the density function of A with itself. If we assume that the density function of A is Gaussian, then it is possible to derive A from SC_2 by deconvolution. The same is true if we assume that the density function of A is a uniform interval. We conjecture that this property holds for a wide range of distributions and we are working to characterize them. Another advantage of the difference distribution is that it is directly relevant for some tasks where what matter is the detection of differences, like the problem of change detection [Leclerc *et al.*, 1998].

5.4 Conditionalization

The above definitions can be conditionalized in two ways, globally by classes of scenes, and locally by score.

Our methodology attempts to estimate the accuracy and reliability of a given point correspondence algorithm for a given specified class of scenes and imaging conditions. Clearly, both the class of scenes and the range of imaging conditions must be well defined for our statistics to be good predictors of the performance of the algorithm on images within the class. Let us call the collection of scenes we shall be using the *test suite of scenes*, and the collection of images of a scene the *test suite of images.* As in any statistical estimation task, it is critical that our test suites represent a broad and unbiased sample of scenes/images within the specified class of scenes and imaging conditions. Just as importantly, however, the accuracy of the projection matrices within the suite must be quite close to the accuracy of the projection matrices that will be used by the algorithm. Assuming that the above caveats hold, we expect that, when we have a sufficiently large test suite of scenes with sufficiently large test suites of images of a sufficiently restrictive class (e.g., aerial images of rural areas taken within some range of elevations and within some period of time), the self-consistency statistics derived from the collection of test suites will be a good predictor of the self-consistency of the algorithm on new images within the class. Furthermore, if our conjecture is true, the statistics will be good predictors of the accuracy and reliability of the algorithm on new images within the class.

We can also conditionalize on the score for the n-tuples obtained from the correspondence algorithm. That is, we can estimate these probabilities for a given value of the score by using only n-tuples whose score is that value (or within some δ of that value, as desired). There are two important applications of this idea. First, it can be used to compare the predictive power of the score associated with a given algorithm in terms of accuracy. Second, and in combination with the conditionalization with respect to type of scenes, it could make it possible to predict an expected accuracy per score, provided that the set of scenes can be partitioned in subsets which experimentally turn out to yield a constant self-consistency for a given score. Several variation factors have to be controlled to achieve this final goal, as described in [Leclerc *et al.*, 1998]. Let us focus next on the geometric factors.

6 Normalizing for geometric imaging invariance

So far, we have assumed that the contribution of each individual n-tuple to the statistics is the same. However this ignores many imaging factors, the most important of them being the number of points in the n-tuple, the distance of the 3D point from the cameras, the geometric configuration of the cameras and the individual resolution of each of them. In this section we describe a simple mechanism to take into account all of these factors in a principled way. This makes it possible to apply 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.

6.1 The Mahalanobis distance

The idea is to start from our observed data, which is the coordinates in the different images of the matched points of the n-tuple, and then use our knowledge of the projection matrices in order to determine what would be the distribution error in reconstruction for a given distribution error in the observation. Assuming a standard distribution error in the observation and varying the imaging factors with each n-tuple will then yield different distribution errors in reconstruction, the parameters of which can be used to achieve the normalization with respect to imaging factors. In other terms, we are computing the change in reconstruction per unit change in image coordinates. The same reasoning is just carried one step further when reprojection is needed for the 2-D self-consistency distributions.

In order to make the calculations tractable, we will assume for the remaining of this section that the observation error (due to image noise and digitalization effects) is Gaussian. As it will be seen in the next paragraph, 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. Assuming that the two estimates M_1 and M_2 are statistically independent, the covariance matrix of the difference is $\Lambda_1+\Lambda_2$. For the purpose of normalizing the differences in the 3-D self-consistency distribution we can therefore weight the squared Euclidean distance between M_1 and M_2 by the sum of their covariances. This yields the squared *Mahalanobis distance*, a random variable which, up to the first order, is a χ^2 with in this case 3 degrees of freedom:

$$\delta_M^2(M_1, M_2) = (\mathbf{M}_1 - \mathbf{M}_2)^T(\Lambda_1 + \Lambda_2)^{-1}(\mathbf{M}_1 - \mathbf{M}_2) \tag{4}$$

For the 2-D self-consistency distribution, an additional step consisting of the reprojection is just applied, and then the Mahalanobis distance is used between 2-D points.

6.2 Determining the reconstruction and reprojection covariances

Under the Gaussian hypothesis, if the measurements are modeled by the random vector $\mathbf{x}$, of mean $\mathbf{x}_0$ and of covariance $\mathbf{\Lambda_x} = E((\mathbf{x} - \mathbf{x}_0)^T(\mathbf{x} - \mathbf{x}_0))$, then for any function f, the vector $\mathbf{y} = f(\mathbf{x})$ is a random vector whose first and second order moments can be expressed very simply, up to a first order approximation, as functions of the first and second order moments of $\mathbf{x}$. In effect, the mean is $f(\mathbf{x}_0)$ and the covariance matrix:

$$\mathbf{\Lambda_y} = \mathbf{J}_f(\mathbf{x}_0)\mathbf{\Lambda_x}\mathbf{J}_f(\mathbf{x}_0)^T \tag{5}$$

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 n-tuple, 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 n-tuple, in the least-squares sense. Let $\mathcal{P}^i$ be the 3×4 projection matrix of view i, such that the 3-D point $[X, Y, Z]$ is projected to the 2-D point of coordinates:

$$\begin{aligned} x_i &= \frac{\mathcal{P}^i_{11}X + \mathcal{P}^i_{12}Y + \mathcal{P}^i_{13}Z + \mathcal{P}^i_{14}}{\mathcal{P}^i_{31}X + \mathcal{P}^i_{32}Y + \mathcal{P}^i_{33}Z + \mathcal{P}^i_{34}} \\ y_i &= \frac{\mathcal{P}^i_{21}X + \mathcal{P}^i_{22}Y + \mathcal{P}^i_{23}Z + \mathcal{P}^i_{24}}{\mathcal{P}^i_{31}X + \mathcal{P}^i_{32}Y + \mathcal{P}^i_{33}Z + \mathcal{P}^i_{34}} \end{aligned} \tag{6}$$

Each of these equations can be rearranged as a linear equation in the variables X, Y, Z. Since each view yields two equations, for a number of views $n \geq 2$, we have an overdetermined system, and $\mathbf{M} = [X, Y, Z]^T$ is obtained as the solution of a least-squares system:

$$\min_{\mathbf{y}} \|\mathbf{Ly} - \mathbf{b}\|^2 \tag{7}$$

Where:

$$\mathbf{L} = \begin{bmatrix} \vdots & \vdots & \vdots \\ x_i\mathcal{P}^i_{31} - \mathcal{P}^i_{11} & x_i\mathcal{P}^i_{32} - \mathcal{P}^i_{12} & x_i\mathcal{P}^i_{33} - \mathcal{P}^i_{13} \\ y_i\mathcal{P}^i_{31} - \mathcal{P}^i_{21} & y_i\mathcal{P}^i_{32} - \mathcal{P}^i_{22} & y_i\mathcal{P}^i_{33} - \mathcal{P}^i_{23} \\ \vdots & \vdots & \vdots \end{bmatrix}$$

$$\mathbf{b} = \begin{bmatrix} \vdots \\ \mathcal{P}^i_{14} - x_i\mathcal{P}^i_{34} \\ \mathcal{P}^i_{24} - y_i\mathcal{P}^i_{34} \\ \vdots \end{bmatrix}$$

The solution of the least-squares problem in Eq (7) can be considered as a function g (that we call the reconstruction function) parameterized by the $\mathcal{P}^i_{jk}$ which maps the $2n$-dimensional vector $\mathbf{x}$ to the 3-D vector:

$$\mathbf{M} = (\mathbf{L}^T\mathbf{L})^{-1}\mathbf{L}^T\mathbf{b}$$

This explicit formula makes it possible to obtain the derivatives of M with respect to the $2n$ measurements $w_i, i = 1 \ldots n, w = x, y$ by:

$$\begin{aligned} \frac{\partial \mathbf{M}}{\partial w_i} &= \\ &\left\{ -(\mathbf{L}^T\mathbf{L})^{-1}(\frac{\partial \mathbf{L}}{\partial w_i}^T\mathbf{L} + \mathbf{L}^T\frac{\partial \mathbf{L}}{\partial w_i})(\mathbf{L}^T\mathbf{L})^{-1} \right\} \mathbf{L}^T\mathbf{b} \\ &+(\mathbf{L}^T\mathbf{L})^{-1}(\frac{\partial \mathbf{L}}{\partial w_i}\mathbf{b} + \mathbf{L}^T\frac{\partial \mathbf{b}}{\partial w_i}) \end{aligned}$$

We also assume that the errors at each pixel are independent, uniform (with value σ), and isotropic. The covariance matrix $\mathbf{\Lambda_x}$ is then diagonal with elements σ^2, so that Eq. (5) reduces to:

$$\begin{aligned} \Lambda_M &= \sigma^2\mathbf{J}_g(\mathbf{x}_0)\mathbf{J}_g(\mathbf{x}_0)^T \\ &= \sigma^2 \begin{bmatrix} \sum\left(\frac{\partial X}{\partial w_i}\right)^2 & \sum\frac{\partial X}{\partial w_i}\frac{\partial Y}{\partial w_i} & \sum\frac{\partial X}{\partial w_i}\frac{\partial Z}{\partial w_i} \\ \sum\frac{\partial Y}{\partial w_i}\frac{\partial X}{\partial w_i} & \sum\left(\frac{\partial Y}{\partial w_i}\right)^2 & \sum\frac{\partial Y}{\partial w_i}\frac{\partial Z}{\partial w_i} \\ \sum\frac{\partial Z}{\partial w_i}\frac{\partial X}{\partial w_i} & \sum\frac{\partial Z}{\partial w_i}\frac{\partial Y}{\partial w_i} & \sum\left(\frac{\partial Z}{\partial w_i}\right)^2 \end{bmatrix} \end{aligned} \tag{8}$$

The above calculations are significantly simplified in the case when the 3-D (X, Y) coordinates remain fixed, so that only the elevation Z varies. Each of the lines in Eq. (6) can be rearranged as a linear equation in the only variable

Z. The solution of the least-squares problem in Eq (7) is given by:

$$Z = \frac{\mathbf{L}^T \mathbf{b}}{\mathbf{L}^T \mathbf{L}}$$

Where:

$$\mathbf{L} = \begin{bmatrix} \vdots \\ x_i \mathcal{P}^i_{33} - \mathcal{P}^i_{13} \\ y_i \mathcal{P}^i_{33} - \mathcal{P}^i_{23} \\ \vdots \end{bmatrix}$$

$$\mathbf{b} = \begin{bmatrix} \vdots \\ \mathcal{P}^i_{11} X + \mathcal{P}^i_{12} Y + \mathcal{P}^i_{14} - x_i(\mathcal{P}^i_{31} X + \mathcal{P}^i_{32} Y + \mathcal{P}^i_{34}) \\ \mathcal{P}^i_{21} X + \mathcal{P}^i_{22} Y + \mathcal{P}^i_{24} - y_i(\mathcal{P}^i_{31} X + \mathcal{P}^i_{32} Y + \mathcal{P}^i_{34}) \\ \vdots \end{bmatrix}$$

Eq. (8) simplifies into:

$$\sigma_Z^2 = \sigma^2 \|\mathbf{D}g(\mathbf{x}_0)\|^2 = \sigma^2 \sum_{1 \le i \le n} \left\{ \left(\frac{\partial Z}{\partial x_i}\right)^2 + \left(\frac{\partial Z}{\partial y_i}\right)^2 \right\}$$

The squared Mahalanobis distance becomes:

$$\delta_M^2(Z_1, Z_2) = \frac{(Z_1 - Z_2)^2}{\sigma_{Z_1}^2 + \sigma_{Z_2}^2}$$

One further reprojection stage is necessary when considering the 2D reprojection statistics: we are now in possession of the reconstruction M and its covariance matrix Λ_M, and we apply the same basic mechanism to the projection functions h_i parameterized by the $\mathcal{P}^i_{jk}$ which map the 3-D vector M to the 2-D vectors $[x_i, y_i]$ as defined by Eq. (6). Since the functions h_i are much simpler, we won't detail the calculation.

6.3 Taking into account the uncertainty in camera parameters

If estimates of the uncertainty of the internal or external parameters of the cameras is available, it can also be incorporated in the calculations above. The simplest way is as follows. First, these estimates are used to determine the covariance matrices $\Lambda_{\mathcal{P}^i}$ of the vector of the elements of the projection matrices $\mathcal{P}^i$, which is possible since the elements of the projection matrix can be considered as functions of the internal and external parameters. Second, the reconstruction function g and reprojection functions h_i remain the same, but the $\mathcal{P}^i_{jk}$ are now considered to be variables instead of parameters. The vector $\mathbf{x}$ is extended to include these $11n$ variables in addition to the $2n$ w_i. Similar calculations are then carried on, resulting in a formula more complicated than Eq. (8) but still manageable if we assume that the projection matrices are uncorrelated to each other and to the pixel error:

$$\Lambda_M = \sigma^2 \begin{bmatrix} \sum \left(\frac{\partial X}{\partial w_i}\right)^2 & \sum \frac{\partial X}{\partial w_i} \frac{\partial Y}{\partial w_i} & \sum \frac{\partial X}{\partial w_i} \frac{\partial Z}{\partial w_i} \\ \sum \frac{\partial Y}{\partial w_i} \frac{\partial X}{\partial w_i} & \sum \left(\frac{\partial Y}{\partial w_i}\right)^2 & \sum \frac{\partial Y}{\partial w_i} \frac{\partial Z}{\partial w_i} \\ \sum \frac{\partial Z}{\partial w_i} \frac{\partial X}{\partial w_i} & \sum \frac{\partial Z}{\partial w_i} \frac{\partial Y}{\partial w_i} & \sum \left(\frac{\partial Z}{\partial w_i}\right)^2 \end{bmatrix} +$$

$$\sum_{1 \le i \le n} \left\{ \begin{bmatrix} \cdots & \frac{\partial X}{\partial \mathcal{P}^i_{jk}} & \cdots \\ \cdots & \frac{\partial Y}{\partial \mathcal{P}^i_{jk}} & \cdots \\ \cdots & \frac{\partial Z}{\partial \mathcal{P}^i_{jk}} & \cdots \end{bmatrix} \Lambda_{\mathcal{P}^i} \begin{bmatrix} \vdots & \vdots & \vdots \\ \frac{\partial X}{\partial \mathcal{P}^i_{jk}} & \frac{\partial Y}{\partial \mathcal{P}^i_{jk}} & \frac{\partial Z}{\partial \mathcal{P}^i_{jk}} \\ \vdots & \vdots & \vdots \end{bmatrix} \right\}$$

7 Experiments

7.1 Ranking the quality of two algorithms

We first illustrate our methodology with an experiment in order to show that the computed statistics make it possible to rank the quality of the results of two correspondence algorithms.

We consider two variations of the terrain mesh surface-reconstruction algorithm [Fua and Leclerc, 1995], called algorithms I and II below. The terrain mesh algorithm produces a triangulated mesh representation of a hypothesized surface in the world given an arbitrary number of views of that surface. To get point correspondences, we reproject the vertices of the mesh in the images. In order to make sure that we get a sufficient number of common-point n-tuples, we constrain the vertices to vary only in a direction parallel to the first camera optical axis.

These two variations of the mesh algorithm were applied on the set of four aerial images shown Fig. 1, using the subsets 1-2, 1-3, 1-4, 1-2-3, 1-2-4. The aerial images are of a road cut into the side of a steep hill.

- Algorithm I is the standard version of the mesh algorithm in which deviation from a smooth surface is penalized during the optimization process.

- Algorithm II has no smoothness constraint, so that it tends to "fit to the noise," thereby producing a lumpier surface.

For this set of images, algorithm I gives results which are visibly better than algorithm II, as illustrated in Fig. 2. It is clearly seen that the mesh produced by algorithm II is lumpier and that the cut in the hill corresponding to the road is not accurately modelled. Another observation which can be done visually is that algorithm I seems to have performed better on pairs 1-2 and 1-3 than on pairs 1-4, possibly because image 4 is noisier or because its projection matrix is less accurate.

The (un-normalized) 2–D image bounding self-consistency distributions for the subsets of experiments { 1-2, 1-3 }, { 1-2, 1-4 }, {1-2, 1-3, 1-2-3 }, { 1-2, 1-3, 1-2-4 } are shown in Fig. 3. It is clear that these distributions capture the superiority of the results of algorithm I over algorithm II, and also the fact that calculations involving image 4 yield worse results.

7.2 Ranking the predictive power of two scoring schemes

In the second set of experiments, we used the normalized difference self-consistency distribution. The same facet-based 3D model was used. Two matching scores are considered,

- A variant of the classical *sum of squared differences* (SSD), the sum, for a given facet, over all sample points, of the sum, over all images, of the squared difference between the predicted and observed image values.
- The *Coding Loss*, a new measure based on Minimum Description Length (MDL) theory, which is the difference between the code length of the image pixels using the facet model and the code length using independent noise models.

The SSD measure is not a good measure to characterize the accuracy of the reconstruction, because 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. The MDL measure was designed to avoid this drawback, as explained in more detail in [Leclerc *et al.*, 1998]. We therefore expect the predictive power of the MDL measure in terms of accuracy to be superior to the predictive power of the SSD measure.

To support the above, we applied both measures to two sets of images of Fort Benning.

- images of a vegetation-free area (Benning 1)
- images of a tree canopy area (Benning 4) where we would not expect stereo to work well because planar facets are very poor models of tree canopies.

The results are shown in Figure 4. First, without taking into account the scores, we can see that the tree canopy images yield a significantly lower self-consistency than the vegetation-free area, confirming that the self-consistency correlates well with the reconstruction performance.

Second, concentrating on the scores, we can clearly see on the vegetation-free graphs that indeed the coding loss (MDL measure) correlates better with the degree of self-consistency than the SSD measure: as the coding loss becomes more positive, the self-consistency decreases monotonically, which is not the case for the SSD.

Furthermore, on the tree canopy graph, with the coding loss, all facets are found to have bad scores and it can be seen that this translates into low self-consistency. On the other hand, although the SSD score also yielded a low degree of self-consistency, this was with a mix of good and bad scores. In addition, we note that the best SSD score in the vegetation-free area yielded the least self-consistent distribution, while in the tree canopy area, the same score yielded the most self-consistent distribution. More experimental data and details on this set of experiments can be found in [Leclerc *et al.*, 1998].

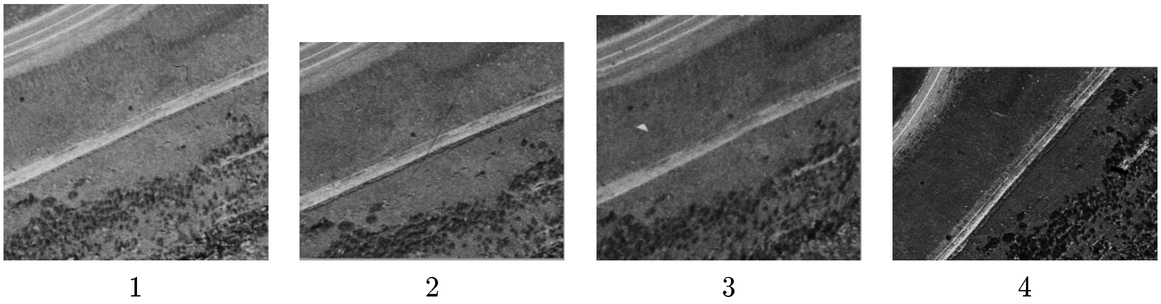


Figure 1: The 4 aerial images of the side of a steep hill used in the experiment. The diagonal line near the center of each image is a dirt road that has been cut into the side of the hill.

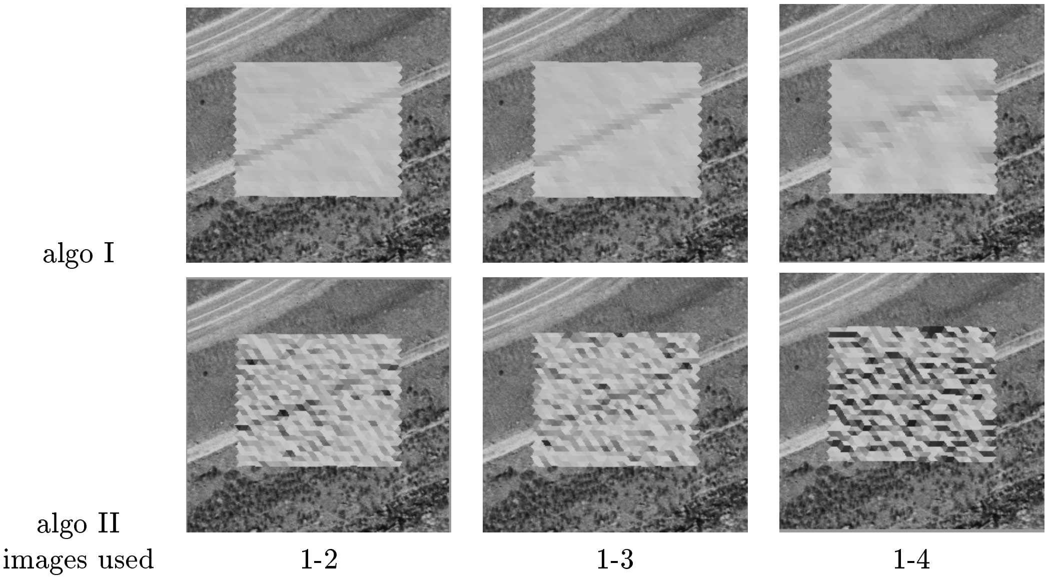
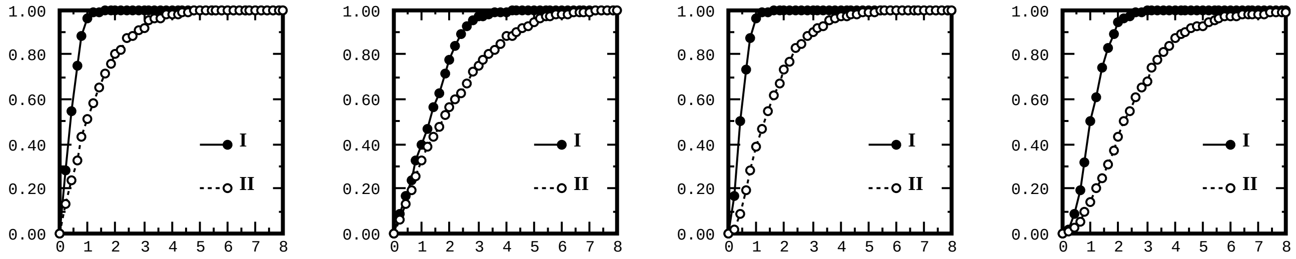


Figure 2: Results of algorithms I and II, shown as a shaded model overlaid on image 1 (the grey value represents orientation). Note that the road cut is clearly visible in the first two images of the upper row, indicating that images 1, 2, and 3 are better than image 4 for reconstruction. Note also that the results for algorithm I are clearly better than for algorithm II.



{ 1-2, 1-3 } { 1-2, 1-4 } {1-2, 1-3, 1-2-3 } { 1-2, 1-3, 1-2-4 }

Figure 3: The 2–D image bounding self-consistency distributions. The curves represent the fraction of sets of common-point n-tuples that are ϵ-self-consistent as a function of ϵ, expressed in pixels. For a given ϵ, the larger this value, the more self-consistent the result of the algorithm is. Note that these statistics clearly reflect the visual observations of Figure 2.

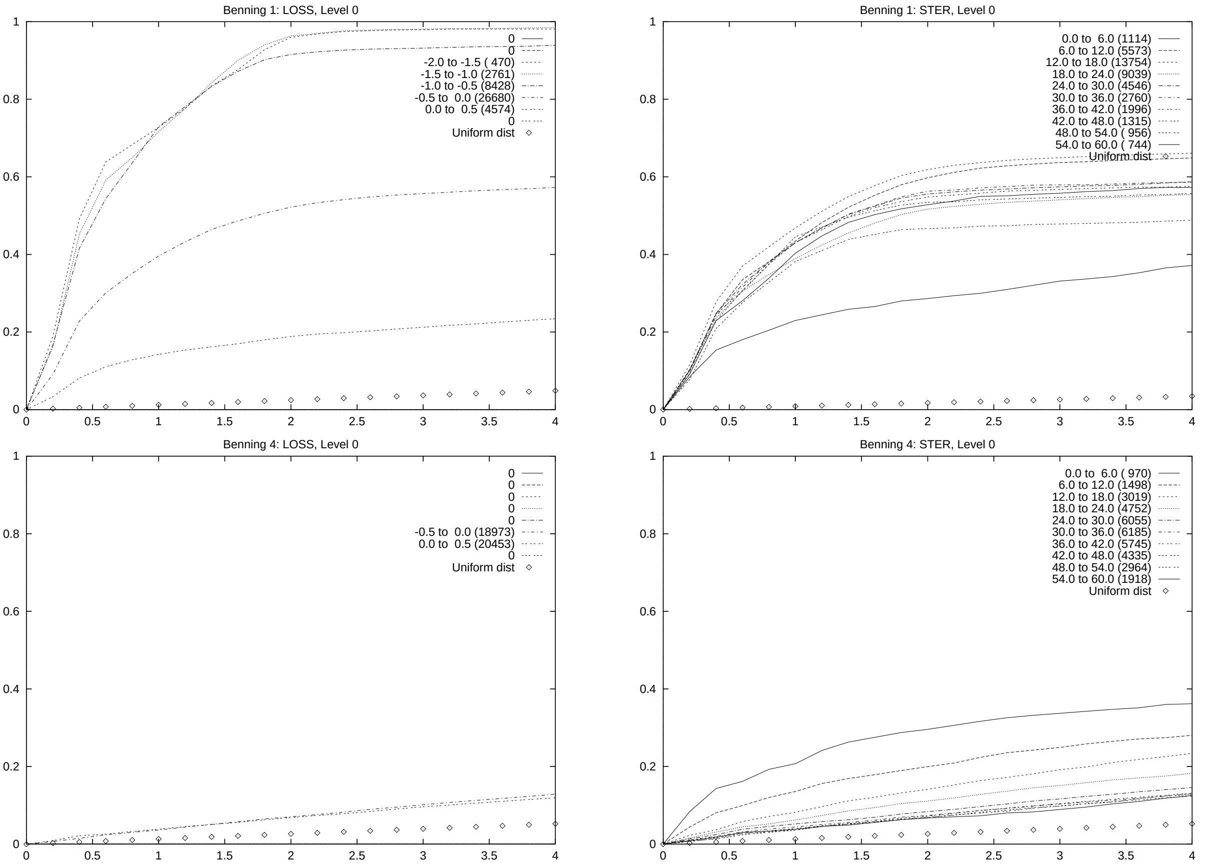


Figure 4: The 3–D normalized Euclidean difference self-consistency distributions parameterized by the score. The curves represent the probability as a function of the Mahalanobis distance between elevations. Top: un-forested terrain. Bottom: tree canopy. Left: coding loss MDL-based score. Right: SSD score. See text for interpretation.

8 Conclusion and Perspectives

We have proposed a methodology that estimates the accuracy and reliability of the results of any multiple-image point correspondence algorithm, without any need for ground truth or camera calibration.

The statistics produced by our methodology can be used in a number of ways by computing the statistics as a function of:

- various imaging conditions, such as time of day, relative viewing angles, or elevation for aerial images.
- the internal parameters of a given algorithm. This provides a quantitative method for fine-tuning an algorithm.
- a "score" associated with each n-tuple. This is especially important when we wish to compare the outputs of a point correspondence algorithm applied to different sets of images of the same scene. See [Leclerc *et al.*, 1998] for an application to change detection based on this idea.
- different subsets of images. As we saw in our preliminary experiment, this can provide a basis for choosing the best images for a given reconstruction task, or at least eliminating those that are clearly inappropriate.
- different point correspondence algorithms. This can provide a basis for comparing different algorithms.

In addition, we believe that the core idea of our methodology, which examines the *self-consistency* of an algorithm across independent experimental trials, can be generalized in order to provide a means to assess accuracy and reliability of algorithms dealing with a range of computer vision problems.

References

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

[Ayache and Lustman, 1987] N. Ayache and F. Lustman. Fast and reliable passive trinocular stereovision. In *Proceedings of the 1st International Conference on Computer Vision*, pages 422–427, London, England, June 1987. IEEE Computer Society Press.

[Cabrera and Meer, 1996] J. Cabrera and P. Meer. Unbiased estimation of ellipses by bootstrapping. *PAMI*, 18(7):752–756, July 1996.

[Csurka *et al.*, 1996] Gabriella Csurka, Cyril Zeller, Zhengyou Zhang, and Olivier Faugeras. Characterizing the uncertainty of the fundamental matrix. *Computer Vision, Graphics, and Image Processing. Image Understanding*, 1996.

[Forstner, 1982] 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.

[Fua and Leclerc, 1995] P. Fua and Y. G. Leclerc. Object-Centered Surface Reconstruction: Combining Multi-Image Stereo and Shading. *The International Journal of Computer Vision*, 16:35–56, September 1995.

[Hartley, 1995] R.I. Hartley. A linear method for reconstruction from lines and points. In *Proc. International Conference on Computer Vision*, pages 882–887, Cambridge, MA, 1995.

[Kanade and Okutomi, 1994] T. Kanade and M. Okutomi. A stereo matching algorithm with an adaptive window: Theory and experiment. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(9):920–932, September 1994.

[Leclerc *et al.*, 1998] Y. Leclerc, Q.-T. Luong, and P. Fua. A framework for detecting changes in terrain. In *ARPA Image Understanding Workshop*, Monterey, CA, 1998.

[Luong and Viéville, 1996] Q.-T. Luong and T. Viéville. Canonical representations for the geometries of multiple projective views. *Computer Visio and Image Understanding*, 64(2):193–229, 1996.

[Matthies, 1992] L. Matthies. Stereo vision for planetary rovers: Stochastic modeling to near real-time implementation. *The International Journal of Computer Vision*, 8(1):71–91, July 1992.

[Mohr *et al.*, 1993] R. Mohr, F. Veillon, and L. Quan. Relative 3d reconstruction using multiple uncalibrated images. In *Proc. Conference on Computer Vision and Pattern Recognition*, pages 543–548, NYC, 1993.

[Shashua, 1994] A. Shashua. Algebraic functions for recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(8):779–789, 1994.

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

[Yachida *et al.*, 1986] M. Yachida, Y. Kitamura, and M. Kimachi. Trinocular vision: New approach for correspondence problem. In *International Conference on Pattern Recognition*, pages 1041–1044. IEEE, October 1986. Paris, France.

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