Performance Characterization Protocol In Computer Vision

Robert M. Haralick*
Intelligent Systems Laboratory
Department of Electrical Engineering
University of Washington
Seattle, WA, 98195
haralick@ee.washington.edu

Abstract

Computer vision algorithms are composed of different sub-algorithms often applied in sequence. Determination of the performance of a total computer vision algorithm is possible if the performance of each of the sub-algorithm constituents is given. The problem, however, is that for most published algorithms, there is no performance characterization which has been established in the research literature. This is an awful state of affairs for the engineers whose job it is to design and build image analysis or machine vision systems.

For these engineers, the issue is how to quickly design machine vision systems which work efficiently and which meet requirements. To do this requires an engineering basis which describes precisely what is the task to be done, how this task can be done, what is the criterion function, and what is the performance of the algorithm under various kinds of random degradations of the input data.

In this paper, we discuss the meaning of performance characterization in general, and then discuss the protocol details under which an algorithm's performance can be characterized.

1 Introduction

Our question is what methodology should researchers be using to facilitate computer vision engineers to quickly design machine vision systems which work efficiently and which meet requirements. Quick efficient designs requires an engineering basis which describes precisely what is the task to be done, how this task can be done, what is the random degradation the input data undergoes, and what is the resulting output random degradation characteristics under different algorithm tuning parameters. Once this is known and a criterion function is specified, it should be possible to compute the performance of the algorithm with respect to the criterion function under various kinds of random degradations of the input data and under different values of algorithm tuning parameters. Such an analysis can be thought of as a system's engineering open loop analysis. Analyzing the more complex adaptive algorithms requires being able to do a closed loop engineering analysis. But to perform a closed loop engineering analysis requires being able to first do an open loop engineering analysis. In this paper, we concentrate on open loop analyses.

Our perspective is as follows: Computer vision algorithms have multiple steps. Each step typically has some tuning parameters. The input data to each step can be considered to be randomly perturbed. The random perturbation on the output data produced by each step is a function of the input random perturbation and the tuning parameters. Associated with the purpose of the vision algorithm is a criterion function. The tuning parameters must be chosen to optimize the criterion function for the given kinds of input random perturbations.

To initiate our analysis, we will first expand on the meaning of performance characterization in general, and then discuss the general theoretical and/or experimental protocol under which an algorithm performance can be characterized.

2 Performance Characterization

What does performance characterization mean for an algorithm which might be used in a machine vision system? The algorithm is designed to accomplish a specific task. If the input data is perfect and has no noise and no random variation, the output produced by the algorithm ought also to be perfect. Otherwise, there is something wrong with the algorithm.

So measuring how well an algorithm does on perfect input data is not interesting. Performance characterization has to do with establishing the correspondence of the random variations and imperfections which the algorithm produces on the output data caused by the random variations and the imperfections on the input data. This means that to do performance characterization, we must first specify a model for the ideal input world in which only perfect input data exist. Then we must give a random perturbation model which specifies how the imperfect perturbed input data arises from the perfect input data. There may be multiple random perturbation models: one for the objects of each class and one for the clutter. Similarly, for the output, we need to specify a model for the ideal output world in which only perfect output data exist. Then we must give a random perturbation model which specifies how the imperfect perturbed output

*Funding from DARPA contract 92-F1428000-000 is gratefully acknowledged.

data arises from the perfect output data.

Notice that because the input and output perfect worlds may be different we must have different models. It is typically the case that an algorithm changes the data unit. For example, an edge-linking process changes the data from the unit of pixel to the unit of a group of pixels. An arc segmentation/extraction process applied to the groups of pixels produced by an edge linking process produces fitted curve segments. This data unit change means that the representation used for the random variation of the output data set may have to be entirely different than the representation used for the random variation of the input data set. In our edge-linking/arc extraction example, the input data might be described by the false alarm/misdetection characteristics produced by the preceding edge operation, as well as the standard deviation in the position and orientation of the correctly detected edge pixels. The random variation in the output data from the extraction process, on the other hand, must be described in terms of fitting errors (random variation in the fitted coefficients) and segmentation errors.

To make this more concrete, suppose, for the sake of argument that a surface defect is a small dark area in a smooth lighter background. This is the idealization. Next we must state the random perturbation model. The random perturbation model describes the density, size, and brightness of the defects. It can do this with a spatial Poisson process. For each size and brightness combination of defect, a number is chosen from an associated Poisson distribution. This number is the number of defects of that kind per unit area with which the surface will be infected. Then the random population of images becomes that obtained by infecting surfaces with a uniform distribution, planting the chosen random number of defects on each unit area of the surface. Then some model of texture needs to be given. There could be one texture for the background and another texture for the defect. This would then constitute a model of the population of images to be processed for defect inspection.

Suppose now that the first operation to be performed on the images from this population is edge detection. By whatever edge detector and edge detector algorithm parameter values used, the edge detector has a performance. There will be some defect edges which are missed and some defect edges which are detected. There will be some background or clutter edges detected. From the performance characteristics of the edge detector and the known random perturbation characteristics of the image model, it will be possible to infer the fraction of misdetected edges and the fraction of false alarms. In addition it will be possible to infer the edge direction distribution for each true detected edge relative to what its true direction is and the edge direction distribution for each falsely detected edge.

Suppose that the next operation is a spoke filter. Then utilizing the information in edge direction, it will be possible to infer for each pixel location for any image the distribution of counts that the given pixel has coming from detected edges in some neighborhood around it. In particular, a distribution of counts due to false background edges for pixels in and around a defect can be determined and a distribution of counts for pixels in the open background area can be determined. Similarly, a distribution of counts due to correct edge detections for pixels in and around a defect and for pixels in the open background area can be determined.

Suppose that the final operation is a detection operation. Suppose that the detection operation is one which looks for relative maximal counts and if the maximal count is great enough declares a defect. Now from the distribution of counts that defect pixels have and the distribution of counts that non-defect pixels have, it should be possible to compute the misdetection and false alarm characteristics of the final defect detection step. And this characterization will be a parametric characterization with the parameters consisting of the Poisson density parameters, the background brightness, the defect brightness and size, and all algorithm tuning parameters.

Next consider the case for segmentation errors. The representation of the segmentation errors must be natural and suitable for the input of the next process in high-level vision which might be a model-matching process, for example. What should this representation be to make it possible to characterize the identification accuracy of the model matching as a function of the input segmentation errors and fitting errors? Questions like these, have typically not been addressed in the research literature. Until they are, analyzing the performance of a machine vision algorithm will be in the dark ages of an expensive experimental trial-and-error process. And if the performance of the different pieces of a total algorithm cannot be used to determine the performance of the total algorithm, then there cannot be an engineering design methodology for machine vision systems.

This problem is complicated by the fact that there are many instances of algorithms which compute the same sort of information but in forms which are actually non-equivalent. For example, there are arc extraction algorithms which operate directly on the original image along with an intermediate vector file obtained in a previous step and which outputs fitted curve segments. There are other arc extraction algorithms which operate on groups of pixels and which output arc parameters such as center, radius, and endpoints in addition to the width of the original arc.

What we need is the machine vision analog of a system's engineering methodology. This methodology has been extremely successful in the analysis, synthesis, and simulation of the most complex engineering systems ever designed, built, and put into operation. This methodology can be encapsulated in a protocol which has a modeling component, an estimation component, a validation componenet, a theoretical error propagation component, an experimental component, and a data analysis component. The next section describes in greater detail these components of an image analysis engineering protocol.

3 Protocol

The protocol has six components: modeling, annotating, estimating, validating, propagating, and optimizing. In addition there is an experimental component and a data analysis component. This protocol applies to each stage of the algorithm as well as to the total algorithm in an end-to-end manner.

The modeling component of the protocol consists of a description of the world of ideal inputs, a description of a random perturbation model by which such non-ideal inputs arise, and a description of a random perturbation process which characterizes the random perturbation of the output for the algorithm stage.

The annotating component involves gathering a representatiive sample of images and annotating the images delineating the different class of structures or clutter so that the statistics of the respective random perturbation processes can be estimated. The annotation means that for each of the different random process involved, the random process for the clutter, the random perturbation process for each of the kinds of entities of interest, the annotation identifies the units which are involved (a set of pixels, a chain of pixels, a boundary, a set of boundaries, an area, a set of areas, etc.) so that each of the units can be measured for its observed value. Then from this data, the free parameters of each random perturbation model can be estimated.

The estimation component involves determining how to estimate the values of the free parameters of the each random perturbation model from experimentally prepared and annotated data sets.

The validation component consists of statistically validating each random perturbation model. This involves using the estimated values for each of the free parameters of the input random perturbation processes as the true values and then with the random perturbation models fully specified, theoretically propagating their effects through each algorithm phase. Then the output data must be annotated so that each different output unit can be identified and assigned a label of its kind. And the free parameters for each of the output random perturbation processes must be estimated. The validation then amounts to comparing the theoretically predicated values of the free parameters of the random perturbation processes with the estimated values of the free parameters fo the random perturbation processes. If they are close enough, then the models are validated.

The propagating component of the protocol specifies the expected relationship between the values of the parameters of the input random perturbation to the values of the parameters of the output random perturbation process. This will be a function of the ideal input data. For an algorithm which has many stages this methodology would permit the theoretical determination of the parameters of the output random perturbation process given the parameters of the the input random perturbation process very much in the manner of chaining together the perturbation parameter transformations from stage to stage. The parameters of the final output random perturbation then relate to what the system user is most interested in by determining the value of the appropriate criterion function for the system and its application.

The optimizing component specifies how the optimal value of the tuning parameters can be determined once the distribution of the different classes of objects has been estimated.

The experimental component of the protocol describes the experiments performed under which the data relative to the performance characterization can be gathered. Some experiments can use synthetically generated images/data and some experiments can use real images/data. In either case the experimental component of the protocol indicates how the input data is to be gathered or generated, how it is to be perturbed, and what values will be measured. The description must be detailed enough so that another researcher can replicate the experiments.

The data analysis component of the protocol describes what statistical analysis must be done on the experimentally observed data to determine the experimental performance characterization, compare the experimental with the theoretically expected performance, and to validate the the perturbation models and/or the approximations/simplifications used in the derivations of the theoretically expected performance.

3.1 Image Acquisition

This part of the protocol describes how, in accordance with the specified model, a suitably random, independent, and representative set of images is to be acquired or generated to constitute the sampled set of images. This aquisition can be done by taking real images under the specified conditions or by generating synthetic images. If the population includes, for example, a range of sizes of the object of interest or if the object of interest can appear in a variety of situations, or if the object shape can have a range of variations, then the sampling mechanism must assure that a reasonable number of images are sampled with the object appearing in sizes, orientations, and shape variations throughout its permissible range. Similarly, if the object to be recognized or measured can appear in a variety of different lighting conditions which create a similar variety in shadowing, then the sampling must assure that images are acquired with the lighting and shadowing varying throughout its permissible range.

Some of the variables used in the image acquisition process are ones whose values will be estimated by the computer vision algorithm. We denote these variables by $z_1, \ldots, z_K$. Other of these variables are nuisance variables. Their values provide for variation. The performance characterization is averaged over their values. We denote these variables by $w_1, \ldots, w_M$. Other of variables specify the state of the controlled random perturbation and noise against which the performance is to be characterized. We denote these variables by $y_1, \ldots, y_J$. The generation of the images in the population can then be described by $N = J + K + M$ variables. If these N variables having to do with kind of lighting, light position, object position, object orientation, permissable object shape variations, undesired object occlusion, environmental clutter, distor-

tion, noise etc., have respective range sets $R_1, \ldots, R_N$, then the sampling design must assure that images are selected from the domain $R_1 \times R_2 \times \ldots \times R_N$ in a representative way. Since the number of images sampled is likely to be a relatively small fraction of the number of possibilities in $R_1 \times R_2 \times \ldots \times R_N$, the experimental design may have to make judicious use of a Latin square layout.

3.2 Random Perturbation and Noise

Specification of random perturbation and noise is not easy because the more complex the data unit, the more complex the specification of the random perturbation and noise. Each specification of randomness has two potential components. One component is a small perturbation component which affects all data units. It is often reasonable to model this by an additive Gaussian noise process on the ideal values of the data units. This can be considered to be the small variation of the ideal data values combined with observation or measurement noise. The other component is a large perturbation component which affects only a small fraction of the data units. For simple data units it is reasonable to model this by replacing its value by a value having nothing to do with its true value, or by introducing extraneous units which have nothing to do with the objects of interest but whose apearance mimics aspects of the objects of interest, (false alarms) or by just eliminating some of the interesting units (misdetections). Large perturbation noise on more complex data units can be modeled by fractionating a unit into pieces or by merging spatially connected units together. In the case of fractionating, values can be given to most of the pieces which would follow from the values the parent data unit had and values can be given to the remaining pieces which have nothing to do with the values the original data unit had. In the case of merging, values can be given to the merged unit which are consistent with the size and geometry and appearance of the merged unit.

This kind of large random perturbation affecting a small fraction of units is replacement noise. It can be considered to be due to random occlusion, linking, grouping, or segmenting errors. Algorithms which work near perfectly on small amounts of random perturbation on all data units, often fall apart with large random perturbation on a small fraction of the data units. Much of the performance characterization of a complete algorithm will be specified in terms of how much of this replacement kind of random perturbation the algorithm can tolerate and still give reasonable results. Algorithms which have good performance even with large random perturbation on a small fraction of data units can be said to be robust.

3.3 Characterization

Some of the variables used in image acquisition are those whose values are to be estimated by the machine vision algorithm. For objects which are not narrow and have distinct features on all their surfaces, object kind, location, and orientation are prime examples. The values of such variables do not make the recognition and estimation much easier or harder, although they may have some minor effect. For example, an estimate of the surface normal of a planar object viewed at a high slant angle will tend to have higher variance than an estimate produced by the planar object viewed at a near normal angle. The performance characterization of an image analysis algorithm is not with respect to this set of variables. From the point of view of what is to be calculated, this set of variables is crucial. From the point of view of performance characterization, the values for the variables in this set as well as the values in the nuisance set are the ones over which the performance is averaged.

Another set of variables characterize the extent of random perturbations which distort the ideal input data to produce the imperfect input data. These variables represent variations which degrade the information in the image, thereby increasing the uncertainty of the estimates produced by the algorithm. Such variables may characterize object contrast, noise, extent of occlusion, complexity of background clutter, and a multitude of other factors which instead of being modeled explicitly are modeled implicitly by the inclusion of random shape perturbations applied to the set of ideal model shapes.

Finally, there are the algorithm tuning constants that must be set in the algorithm. The values of these variables may to a large or small extent change the performance of the algorithm. And, for best performance, they need to be set to provide best algorithm performance for the given input random perturbation processes.

The random perturbation parameters and the algorithm tuning parameters constitute the set of variables in terms of which the performance characterization must be measured. Let us denote the setable algorithm parameters by x, the parameters of the input random perturbation processes by y_i, the ideal values of the units of the input image by z_i, the observed values made on the units of the input image by $\hat{z}_i$, the ideal values of the units on the output image by z_o, and the observed values made on the units of the output image by $\hat{z}_o$, and the values of the free parameters of the output random perturbation process by y_o. Then $\hat{z}_i$ is to be regarded as an outcome of the input random perturbation processes with free parameters y_i acting on z_i and $\hat{z}_o$ is to be regarded as an outcome of the output random perturbation processes with free parameters y_o acting on z_o. And the random perturbation models provide the way of calculating the probabilties or the probability densities $p(\hat{z}_i|z_i, y_i)$ and $p(\hat{z}_o|z_o, y_o)$.

The perturbation propagation step amounts to theroetically determining y_o, the values of the free parameters of the output random perturbation processes, as a function of z_i, the ideal input state, y_i, the values of the free parameters of the input random perturbation processes, and x the algorithm turning constants. No criterion function is involved in this step and this is the reason that it becomes possible to propagate the values of the free parameters of the random perturbation processes from one algorithm stage to another. It is not until the last algorithm stage that a criterion function is needed. The criterion function

e is a scalar function comparing the ideal output z_o with the observed output $\hat{z}_o$: $e(z_o, \hat{z}_o)$. So if $e(z_o, \hat{z}_o)$ is known and $p(\hat{z}_o|z_o, y_o)$ is known then it becomes possible to compute the expected value of e, or for that matter any of its moments or any statistics of its distribution. For example,

$$E[e(z_o, \hat{z}_o)|y_o] = \sum_{z_o} e(z_o, \hat{z}_o) p(\hat{z}_o|z_o, y_o)$$

And since y_o depends on the ideal input state z_i, the free parameters of the input random perturbation process y_i, and the algorithm tuning paramters x, we see that it becomes possible to compute $E[e(z_o, \hat{z}_o)|z_i, y_i, x]$. If the input environment can be characterized by a prior probability $p(z_i)$, then the expected value of the criterion function can be computed over the input environment population.

$$E[e(z_o, \hat{z}_o)|y_i, x] = \sum_{z_i} E[e(z_o, \hat{z}_o)|z_i, y_i, x] p(z_i)$$

Now it becomes clear what the smartest non-adaptive algorithm control should do: it must determine the value of algorithm tuning parameters x to maximize $E[e(z_o, \hat{z}_o)|\hat{y}_i, x]$, where $\hat{y}_i$ is the estimated value of y_i.

3.4 Reliability

Consider algorithms which estimate some parameter such as position and orientation of an object. One kind of criterion function e is reliability. An estimate can be said to be reliable if the algorithm is operating on data that meets certain requirements and if the difference between the estimated quantity and the true but known value is below a user specified tolerance. An algorithm can estimate whether the results it produces are reliable by making a decision on estimated quantities which relate to input data noise covariance, output data covariance, and structural stability of calculation. Output quantity covariance can be estimated by estimating the input data noise variance and propagating the error introduced by the noise covariance into the calculation of the estimated quantity. Hence the algorithm itself can provide an indication of whether the estimates it produces have an uncertainty below a given value. High uncertainties would occur if the algorithm can determine that the assumptions about the environment producing the data or the assumptions required by the method are not being met by the data on which it is operating or if the random perturbation in the quantities estimated is too high to make the estimates useful.

Characterizing reliability can be done by two means. The first is by the probability that the algorithm claims reliability as a function of algorithm parameters and parameters describing input data random perturbations. The second is by misdetection false alarm operating curves. A misdetection occurs when the algorithm indicates it has produced a reliable enough result when in fact it has not produced a reliable enough result. A false alarm occurs when the algorithm indicates that it has not produced a reliable enough result when in fact it has produced a reliable enough result. A misdetection false alarm rate operating curve results for each different noise and random perturbation specification. The curve itself can be obtained by varying the algorithm tuning constants, one of which is the threshold by which the algorithm determines whether it claims the estimate it produes is reliable or not.

3.5 Robustness

Robustness of an algorithm is the degree to which large perturbations in a small fraction of the data units affects the variance of the parameters estimated by the algorithm. To measure the robustness of an algorithm we can perturb a given fraction of the data units. Suppose of the N data units, K data units are perturbed with a large perturbation and $N - K$ data units are perturbed with small perturbations. Next we determine the expected value of the error criterion function for two cases: the case of $N - K$ data units perturbed with small perturbations, the error criterion function having expected value E_1, and the case of N data units with K data units perturbed with large perturbations, the error criteion function haveing expected value E_2. The robust efficiency can be measured as E_2/E_1. The idea behind this measure is that the large perturbations essentially create outliers and the function of the robustness of the algorithm is to throw out the outliers. In that case, the best the algorithm can do is to estimate based on the $N - K$ inlier units. But a robust algorithm is not perfect and will not completely throw out all the outliers. So by looking at the ratio we can measure the extent to which robust processing is really working.

3.6 Experiments

In a complete design, the values for the algorithm tuning parameters x, and the values of the free parameters of the random perturbations y will be selected in a systematic and regular way. The values for z, which specify the ideal input data state and the values for the nuisance variables will be sampled from a uniform distribution over the range of their permissible values.

The values for z uniquely specify an ideal image. The values for y specify the extent to which random perturbations and noise are randomly introduced into the ideal image and/or object(s) in the ideal image. In this manner, each noisy trial image may be synthetically generated. Or if the qcquisiton is to be with real images, then the real images must be annotated, the annotation essentially defining z. The values for x specify the algorithm tuning parameter values. The algorithm is then run over the trial image producing estimated values $\hat{z}$ for z. The data produced by each trial then consists of a record $x, y, z, \hat{z}$.

The data analysis plan describes how the set of records produced by the experimental trials will be processed or analyzed to compactly express the performance characterization. For example, an equivalence relation on the range space for y may be defined and an hypothesis may be specified stating that all combinations of values of y in the same equivalence class have the same expected error. The data analysis plan would specify the equivalence relation and give the statistical procedure by which the hypothesis

could be tested. Performing such tests are important because they can reduce the number of variable combinations which have to be used to express the performance characterization. For example, the hypothesis that all other variables being equal, whenever the ratio of the last two components of y have the value k, then the expected performance is identical. In this case, the performance characterization can be compactly given in terms of k and the remaining components of y.

Once all equivalence tests are complete, the data analysis plan would specify the kinds of graphs or tables employed to present the experimental data. It might specify the form of a simple regression equation by which the expected error, the probability of claimed reliability, the probability of misdetection, the probability of false alarm, and the computational complexity or execution time can be expressed in terms of the independent variables x, y. As well it would specify how the coefficients of the regression equation could be calculated from the observed data. If error propagation can be done analytically using the parameters associated with input data noise variance and the ideal noiseless input data, the data analysis plan can discuss how to make the comparison between the expected error computed analytically and the observed experimental error.

Finally, if the computer vision algorithm must meet certain performance requirements, the data analysis plan must state how the hypothesis that the algorithm meets the specified requirement will be tested. The plan must be supported by a theoretically developed statistical analysis which shows that an experiment carried out according to the experimental design and analyzed according to the data analysis plan will produce a statistical test itself having a given accuracy. That is, since the entire population of images is only sampled, the sampling variation will introduce a random fluctuation in the test results. For some fraction of experiments carried out according to the protocol, the hypothesis to be tested will be accepted but the algorithm, in fact, if it were tried on the complete population of image variations, would not meet the specified requirements; and for some fraction of experiments carried out according to the protocol, the hypothesis to be tested will be rejected but if the algorithm were tried on the complete population of image variation, it would meet the specified requirements. The specified size of these errors of false acceptance and missed acceptance will dictate the number of images to be in the sample for the test. This relation between sample size and false acceptance rate and missed acceptance rate of the test for the hypothesis must be determined on the basis of statistical theory. One would certainly expect that the sample size would be large enough so that the uncertainty caused by the sampling would be below 20%.

For example, suppose the error rate of a quantity estimated by an machine vision algorithm is defined to be the fraction of time that the estimate is further than ϵ_0 from the true value. If this error rate is to be less than $\frac{1}{1,000}$, then in order to be about 85% sure that the performance meets specification, 10,000 tests will have to be run. If the image analysis algorithm performs incorrectly 9 or fewer times, then we can assert that with 85% probability, the machine vision algorithm meets specification.

4 Protocol Summary

Any protocol must include the following items.

1. The protocol must state:
   - what is to be measured on each experimental trial
   - what is to be inferred (estimated) from the measurements or what hypothesis is to be tested (the requirement)
   - with what precision (standard deviation) is the inferred quantity to be estimated or with what misdetect and false alarm error is the hypothesis to be tested
   - over what population of scenes/images and vision algorithm tuning parameters are the experiments to be done
2. The protocol must layout an experimental design which describes
   - how a suitable random, independent, and representative set of images from the specified population is to be sampled, generated, or acquired, including the setting of all parameters relevant to the population
   - what parameters of the algorithm will be varied and how their values will be varied
   - what parameters of the algorithm will be set and what values they will be set at
   - the experiments which must carried out for each image in the population
   - the accuracy criterion which states how the comparison between the true values and the measured values will be evaluated
3. The protocol must have a data analysis plan which
   - states how to test the hypothesis that the algorithm meets the specified requirement
   - indicates how the data (the true values and the corresponding measured values) will be analyzed
   - tells explicitly what performance curves will be generated
4. The data analysis plan
   - must be supported by a theoretically developed statistical analysis
   - and show that an experiment carried out according to the experimental design and analyzed according the data analysis plan will produce results having a given accuracy (i.e., the rates of false alarms and miss detections due to the experimental sampling variation)

5 Summary

We have described a protocol for computer vision algorithm performance characterization, not just in terms of a simple criterion function, but in terms of how the random perturbation characterization on the output is a function of the random perturbation characterization of the input. Although this idea is a classic one in system's engineering, it is one which has been more neglected in the computer vision community. Indeed, there is actually considerable resistance to this methodology, a methodology which has been considered part of sound engineering practice for a long time. This resistance says something about where the field is. Statements like "It is too hard." or "It involves too much work to do this correctly and only get a single paper out of it." or "It requires having a fluency with probability and statistics that I do not have." or "The field is not yet ready for this." or "It is not something new or advanced enough for many funding agencies to be interested in," are indicative of a cultural attitude. Engineers do what engineers must to sucessfully get an engineering project done. They work hard. They learn what they have to. They experiment to get data points they do not have. They do analysis of subsystems. They do simulation of their designs. They do performance characterization.

Computer vision as an engineering or as an experimental discipline can be advanced in a deeper way by a general change in cultural paradigm. Changes in cultural paradigms are precisely the changes that have historically advanced science. They have been responsible for the scientific revolutions.

Our work will have served its purpose if it facilitates discussions and thinking which helps move the computer vision frontiers by the opening of a more comprehensive methodology essentially effecting a paradigm change.

The paradigm change puts a different look at the way that we are called upon to do our research. For it suggests that one of the first steps is to gather a suitable real data set and annotate or ground-truth it. And from this data set the parameters of the perturbation model must be estimated and then the perturbation model must be statistically validated. Then having a validated perturbation model, we should proceed to the design of the algorithm step whose input data perturbation model we have in hand. And the design will use the values of the algorithm tuning parameters which optimize the expected value of the final error criterion.