CVGIP: IMAGE UNDERSTANDING
Vol. 60, No. 2, September, pp. 245–249, 1994

DIALOGUE

Performance Characterization in Computer Vision

ROBERT M. HARALICK

University of Washington, Seattle, Washington 98195

1. INTRODUCTION

Computer vision algorithms are composed of different subalgorithms often applied in sequence. Determination of the performance of a total computer vision algorithm is possible if the performance of each of the subalgorithm 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.

This suggests that there has been a cultural deficiency in the computer vision community: computer vision algorithms have been published more on the merit of an experimental or theoretical demonstration suggesting that some task can be done, rather than on an engineering basis. Such a situation was tolerated because the interesting question was whether it was possible at all to accomplish a computer vision task. Performance was a secondary issue.

Now, however, a major interesting question 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 error criterion, and what is the performance of the algorithm under various kinds of random degradations of the input data. To accomplish this for adaptive algorithms requires being able to do a closed loop engineering analysis. To perform a closed loop engineering analysis requires being able to first do an open loop engineering analysis.

The purpose of this discussion is to raise our sensitivity to these issues so that our field can more rapidly transfer the research technology to a factory floor technology. To initiate this dialogue, we will first expand on the meaning of performance characterization in general and then discuss the 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 world in which only perfect data exist. Then we must give a random perturbation model which specifies how the imperfect perturbed data arises from the perfect data. Finally, we need a criterion function which quantitatively measures the difference between the ideal output arising from the perfect ideal input and the calculated output arising from the corresponding randomly perturbed input.

Now we are faced with an immediate problem relative to the criterion function. 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

1049-9660/94 $6.00

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. Hence, the criterion function may change from stage to stage in the analysis process.

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 output 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 engienering methodology. This methodology can be encapsulated in a protocol which has a modeling 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 modeling component of the protocol consists of a description of the world of ideal images, a description of a random perturbation model by which non-ideal images arise, and a specification of the criterion function by which the difference between the ideal output and the computed output arising from the imperfect input can be quantified. The experimental component describes the experiments performed under which the data relative to the performance characterization can be gathered. The analysis component describes what analysis must be done on the experimentally observed data to determine the performance characterization.

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 from the population of ideals is to be acquired or generated to constitute the sampled set of images. This acquisition 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 generation 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, distortion, 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. Large perturbation noise on more complex data units can be modeled by fractionating the unit into pieces and giving values to most of the pieces which would follow from the values the parent data unit had and giving values to the remaining pieces which have nothing to do with the values the original data unit had.

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. *Performance Characterization*

Some of the variables used in the image generation are those whose values are to be estimated by the machine vision algorithm. 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 may be other variables governing parameter constants that must be set in the image analysis algorithm. The values of these variables may to a large or small extent change the performance of the algorithm.

The variables governing the extent of random perturbations and the variables which are the algorithm parameter constants constitute the set of variables in terms of which the performance characterization must be measured. Suppose there are I algorithm parameters $x_1, \ldots, x_I$, which can be set, J different variables $y_1, \ldots, y_J$ governing the extent of random perturbations, and K different measurements $\hat{z}_1, \ldots, \hat{z}_K$ to be made on each image. There will be a difference between the true ideal values $z_1, \ldots, \hat{z}_K$ of the measured quantities and the measured values $\hat{z}_1, \ldots, \hat{z}_K$ themselves. The error criterion, $e(z_1, \ldots, z_K, \hat{z}_1, \ldots, \hat{z}_K)$, must state how the comparison between the ideal values and the measured values will be evaluated. Its value will be a function of the I algorithm parameters and the J random perturbation parameters.

An algorithm can have two different dimensions to the error criterion. To explain these dimensions, consider algorithms which estimate some parameter such as position and orientation of an object. One dimension the error criterion can have 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 variance, 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 variance 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 this dimension 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 produces is reliable or not.

The second dimension of the error criterion would be related to the difference between the true value of the quantity of interest and the estimated value. This criterion would be evaluated only for those cases where the algorithm indicates that it produces a reliable enough result.

Each estimated quantity $\hat{z}_k$ is a function of the values of the algorithm constants $x_1, \ldots, x_I$ and the random perturbation induced on the image by the values of the variables $y_1, \ldots, y_J$ and each z_k is a function only of the algorithm constants $x_1, \ldots, x_I$. The expected value E of $e(z_1, \ldots, z_K, \hat{z}_1, \ldots, \hat{z}_K)$ is, therefore, a function of $x_1, \ldots, x_I$ and $y_1, \ldots, y_J$. Performance characterization of the estimated quantity then amounts to expressing in graph, table, or analytic form $E[e(z_1, \ldots, z_K, \hat{z}_1, \ldots, \hat{z}_K)]$ as a function of $x_1, \ldots, x_I$ and $y_1, \ldots, y_J$.

3.4. *Experiments*

In a complete design, the values for the algorithm constants $x_1, \ldots, x_I$ and the values governing the random perturbations $y_1, \ldots, y_J$ will be selected in a systematic and regular way. The values for $z_1, \ldots, z_K$ and the values for the nuisance variables $w_1, \ldots, w_M$ will be sampled from a uniform distribution over the range of their permissible values.

The values for $z_1, \ldots, z_K$ uniquely specify an ideal image. The values for $y_1, \ldots, y_J$ 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 is generated. The values for $x_1, \ldots, x_I$ specify how to set the parameter constants required by the algorithm. The algorithm is then run over the trial image producing estimated values $\hat{z}_1, \ldots, \hat{z}_K$ for $z_1, \ldots, z_K$. Applying the error criterion then produces the values $e(z_1, \ldots, z_K, \hat{z}_1, \ldots, \hat{z}_K)$. The data produced by each trial then consists of a record

$$x_1, \ldots, x_I, y_1, \ldots, y_J, \quad e(z_1, \ldots, z_K, \hat{z}_1, \ldots, \hat{z}_K).$$

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 $y_1, \ldots, y_J$ may be defined and an hypothesis may be specified stating that all combinations of values of $y_1, \ldots, y_J$ 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 is 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 y_{J-1}/y_J has a ratio of k, then the expected performance is identical. In this case, the performance characterization can be compactly given in terms of k and $y_1, \ldots, y_{J-2}$.

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 $x_1, \ldots, x_I, y_1, \ldots, y_J$. As well, it would specify how the coefficients of the regression equation could be calculated from the observed data.

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 flucation 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 a 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 $\frac{1}{1000}$, 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 nine or fewer times, then we can assert that with 85% probability, the machine vision algorithm meets specification [1].

4. CONCLUSION

We have discussed the problem of the lack of performance evaluation in the published literature on computer vision algorithms. This situation is causing great difficulties for researchers who are trying to build upon existing algorithms and for engineers who are designing operational systems. To remedy the situation, we suggested the establishment of a well-defined protocol for determining the performance characterization of an algorithm. Use of this kind of protocol will make using engineering system methodology possible as well as making possible well-founded comparisons between machine vision algorithms that perform the same tasks. We hope that our discussion will encourage a thorough and overdue dialogue in the field so that a complete engineering methodology for performance evaluation of machine vision algorithms can finally result.

REFERENCE

1. R.M. Haralick, Performance assessment of near perfect machines, *Mach. Vision Appl.* **2**(1), 1989, 1–16.

CVGIP: IMAGE UNDERSTANDING
Vol. 60, No. 2, September, pp. 250–252, 1994

REPLY

On the Paper by R. M. Haralick

L. CINQUE, C. GUERRA, AND S. LEVIALDI

Rome, Italy

1. GENERAL COMMENTS

The article by R. M. Haralick [1] underlines the importance of performance evaluation studies at a formal level in a vision system where noise plays a crucial role in the quantitative study of the results. He also suggests considering a number of steps for the overall evaluation of an artificial vision system: (1) a noise model to apply on input images; (2) a closed loop engineering analysis; (3) a parameter set for measuring both algorithmic performance and robustness with respect to corrupted images.

With respect to the first point, we should note that most studies use images with superimposed artificial noise which may not reproduce real scenes. The nature of the image perturbation may be local (as in the occlusion case mentioned in the note) or global with some peculiar aspects due to the physical nature of the scene (dirty background, irregular illumination, uneven shading, etc.).

As for the second point we agree on the importance of a feedback loop for evaluating the algorithmic performance as a function of the obtained results (implicitly considering the existence of an automatic learning mechanism), although an open loop analysis may also provide a significant insight.

As a last point, it seems difficult to extract or define general parameters for the evaluation of a full system since different data representations are generally used at different processing levels (sometimes having specific computer architectures exploiting parallelism).

Consider, for instance, the evaluation of an optical character reader where the input device may differ from system to system (various signal-to-noise ratios and transfer functions), the application may include only printed fonts or handwritten ones, the use of temporal information (as in sequential recognizers) and/or context, the use of custom chips or of a general purpose computer with special recognition programs, etc. In all these cases it would be extremely difficult to provide parameters that may help in quantifying the performance of such systems.

2. CULTURAL ORIGINS

The solution to a computer vision problem strongly depends on education, i.e., on the university degree (mathematics, engineering, computer science, psychology, etc.) and on the cultural background. For instance, consider the problem of recognizing and manipulating objects in the blocks world, with some given constraints. A mathematician would first consider the geometrical description of the scene, then model the objects, and finally, try to match the observed scene to those models for recognition; an engineer would analyze the required measurements (from range finders, light sensors, telecameras, etc.) in order to obtain values to be used for object detection, recognition, and manipulation; a computer scientist would first preprocess the scene and then segment for subsequent identification and labeling of the components/objects giving emphasis on the computational aspect and communication issues of all the implied algorithms; a cognitive scientist would analyze perceptual cues from the objects to infer a description that could be later used in a human-like recognition process.

Performance studies, originated by the engineering approach have been sometimes neglected by researchers. In the early vision systems many other technical questions had to be answered first in order to become operational and, moreover, the difficulties encountered in the overall evaluation of such systems discouraged the investigation of a performance characterization particularly when a task independency is requested.

ESPRIT is a good example of a combined European research effort that, starting from well-defined tasks, generally within industrial environments to achieve specific goals, tries to develop innovative methodologies and techniques for solving such tasks in order to design prototypical systems. This research program involves both academia and industries with well-defined roles; in one particular program one of the purposes of an applied research program considers performance evaluation explicitly [2].

1049-9660/94 $6.00

3. PERFORMANCE STUDIES IN THE PAST

Performance evaluation has always been a difficult and nonrewarding effort much like standardization and creation of large bibliographical databases; all these efforts have been postponed since they are believed to be neither gratifying nor publishable.

Let us start by describing the evolution of the approaches to performance evaluation in image processing from the beginning of the 1960s. Initially, the main problem was first to establish whether the task could be accomplished at all. This was not obvious, since the solution of many subtasks (overcoming signal-to-noise problems, preprocessing for feature extraction, choosing some significant measurements, achieving high classification rate, etc.) was necessary in order to achieve the recognition task. Each single research group, worked on his own test data for a given application area (biomedical, physical, industrial, geographical, etc.). Later on, during the 1970s, quantitative analysis was done on the time required to solve the problem on a given class of machines (in terms of clock cycles, time dependency, program length, portability, cost) and finally (during the 1980s) the interest was shifted toward the evaluation of the quality of the obtained results in a formal way.

This is indeed a challenging task since, although many efforts have been devoted to express the processes involved in image transformations and pattern recognition within artificial vision systems, it has proved difficult both to express image quality in a quantitative way as well as recognition efficiency; as a consequence no progress has been made in reaching a universal solution. Keeping in mind that the computer has always played a central role and that it has continuously evolved (see different computational models) within the area of image processing and pattern recognition applications, the borderlines between algorithm design, architecture development, and technological choice (acquisition transducers, optical computing, communication, etc.) have been gradually displaced. First, new algorithms were developed to accomplish well-defined tasks which could be integrated into full software packages, next new computer architectures were conceived so as to match the image data and processing tasks to the corresponding algorithms, and finally, specific circuits used for the processing units and interconnections were designed and built to improve reliability and overall performance.

Benchmarks were introduced in the 1980s and one significant case is the Abingdon cross [3] which contained the most typical, elementary image processing operations and, as such, was easily accepted by over 30 different groups which provided their coded programs and time/cost values to perform the task. Such task was the extraction of the skeleton of a cross embedded in Gaussian noise. In this effort the number of image pixels, the required time, and the machine cost were considered, neglecting the evaluation of the perceptual quality of the final image. As mentioned and motivated in [4], a useful benchmark for image processing is difficult to achieve, although it is "an important and worthwhile exercise."

Programs are language dependent and with the new suggested paradigms (object oriented, logical, functional, etc.) it is even more difficult to make comparisons of transparency, portability, simplicity, program length, etc.; execution times are able hardware and technology dependent while cost heavily relies on the number of systems produced/sold. For the above reasons it would be better to introduce other parameters, possibly taking into account the quality of the obtained results.

Another issue in the performance evaluation activity is the definition of the set of test data which, in some cases, is provided by an institution (as an example, a character database is stored and distributed by the NBS); in other cases it may be defined by a group of researchers (multicomputer workshops) [5–9].

4. HUMAN JUDGMENT OF RESULTS

Although it would be nice to have a quantitative evaluation of performance given by an analytical expression, or more visually by means of a table or graph, we must remember that the final evaluator is man and that his subjective criteria depend on his practical requirements. In order to do this, a better presentation of the output may help to make judgments about the obtained results (partial and final); image visualization in a controlled environment and with real time presentation greatly facilitates the observer's evaluation.

5. SYSTEM ROBUSTNESS

In order for a system to be reliable and usable in a real environment it should not collapse with minor local perturbations. A similar approach to the one used in material sciences, i.e., stress analysis, could be employed in the evaluation of the vision system robustness by establishing a perturbation scale and a threshold above which no correct response is obtained. Even if no fully formal characterization of robustness has been given in computer vision yet, this requirement is always on the forefront, particularly in fields like image motion analysis and 3D reconstruction, where small variations produce occlusion so severe that recognition is hindered. Similarly to other fields, a graceful degradation is also desirable since a poorer response is better than no response from the system.

6. DIFFERENT DATA REPRESENTATION

As mentioned in Haralick's note, since representation of data may differ according to the processing level (low

level, intermediate level, and high level in artificial vision) the introduction of a cost function may turn out to be highly difficult due to the nonhomogeneous nature of the data; this is particularly true for the higher (logical/semantic) levels reached at the end of the process with respect to the source information at the basic or pixel level. On the other hand, if a performance value is computed for each level they cannot be added to obtain a global performance value since we are working with a nonlinear system.

7. CONCLUSIONS

We strongly appreciate the attempt to characterize the quality of an image processing system independently from the task it is performing, and, as mentioned above, we realize that many difficulties in achieving such a goal may be encountered. We believe that we still have a long way to go and therefore must now principally rely on human judgement for obtaining a practical evaluation; for some specific applications we feel that this is doomed to be the only possibility.

REFERENCES

1. R. M. Haralick, Performance characterization in computer vision, 1991, manuscript for the Dialogue section of *CVGIP: Image Understanding*.
2. ESPRIT II, Applied Research Program: PEMMON: Performance Management Monitoring Open Network, Proj. N. 5371, Sect. II-3-6, 1990.
3. K. Preston, Jr., The Abingdon Cross benchmark survey, *IEEE Comput.* **22**(7) 1989, 9–21.
4. M. J. B. Duff, How not to benchmark image processors, in *Evaluation of Multicomputers for Image Processing,* (U. L. Uhr, K. Preston, Jr., S. Levialdi, and M. J. B. Duff, Eds.), pp. 3–12, Academic Press, New York/London, 1986.
5. M. J. B. Duff and S. Levialdi, *Languages and Architectures for Image Processing,* Academic Press, London, 1981.
6. *Integrated Technology for Parallel Image Processing* (S. Levialdi, Ed.), Academic Press, London, 1985.
7. M. J. B. Duff, S. Levialdi, K. Preston, Jr., and L. Uhr (Eds.), *Evaluation of Multicomputers for Image Processing,* Academic Press, New York, 1986.
8. L. Uhr, K. Preston, Jr., S. Levialdi, and M. J. B. Duff (Eds.), *Evaluation of Multicomputers for Image Processing,* Academic Press, London, 1986.
9. S. Levialdi (Ed.), *Multicomputer Vision,* Academic Press, London, 1988.

CVGIP: IMAGE UNDERSTANDING
Vol. 60, No. 2, September, pp. 253–256, 1994

REPLY

Performance of Computer Vision Algorithms

JUYANG WENG AND T. S. HUANG

Department of Computer Science, Michigan State University, East Lansing, Michigan 48824

Haralick [1] has raised a very important issue: that of performance characterization of computer vision algorithms. We argue that his goal as stated in the protocol is laudable but that it is in most cases very difficult, if not impossible, to achieve. Nonetheless, because performance evaluation is of the utmost importance, we need to try to approach this goal as closely as possible. The protocol he proposed covers three components of performance evaluation: image generation, random perturbation, and performance characterization. Each component may involve a large number of variables. Since the number of images sampled is likely to be relatively small, the applicability and proper selection of these variables is of great importance to the quality of the performance evaluation. We bring up some related issues that need attention in conducting performance evaluation.

1. WIDE VARIATION OF VISION PROBLEMS

During the 1970s and early 1980s, fascinated by the power of the computer, computer vision researchers identified various vision tasks and attempted to develop algorithms that perform these tasks, often assuming some idealized and restricted situations. During that early stage, since the main concern was about whether something can be done, the techniques that were used by those algorithms are often relatively crude and little or no performance characterization was conducted. After some attempts, which ended up with either preliminary success or failure, many researchers have realized the limitation of those techniques and started to seriously investigate computer vision on a more solid basis. Computer vision researchers became concerned with rigorous definition of the vision tasks, the solution methods, as well as other related issues such as the existence, uniqueness, completeness, stability optimality, robustness, and efficiency of the solution. Today many of us believe that computer vision is a discipline of both science and engineering which needs rigorous scientific methodology and precise engineering specifications. But as pointed out by Haralick, for most published computer vision algorithms, there is little or no performance characterization. It appears that this phenomenon is more evident among high-level AI type algorithms than among low-level image processing type algorithms.

The lack of performance characterization in computer vision is due largely to the difference in nature between computer vision and traditional engineering disciplines. The major difference lies in the complexity of the vision problems and the need for knowledge. The complexity is reflected by the fact that most vision problems have not been rigorously defined. The available definitions of most vision problems are rather descriptive and rely very much on our experience with human visual capabilities. Because the problem itself is not well defined, the performance characterization is then groundless. The use of knowledge (which can be implicitly embedded into the solution methods) implies that the performance is highly scene dependent. For example, a texture segmentation algorithm may work well on one type of image but fall apart on other types. Due to the complexity of the problems and the use of knowledge, many factors (e.g., texture type, background complexity, and occlusion) that are closely related to the performance of an algorithm cannot be easily parameterized without contaminating the performance evaluation with some subjective bias.

The wide variation of computer vision problems may require that performance characterization be conducted according to the problem category. Roughly speaking, computer vision problems fall into three categories. The first category corresponds to low level problems. A problem is considered as low level here if it can be solved, to a large extent, from images based on mathematics and physics, and little or no high-level knowledge is necessary. Edge detection, shape from shading, and motion parameter estimation are such examples. The second category contains middle level problems whose solution requires extensive use of the knowledge about the visual appearance of the objects. Generic object recognition and segmentation belong to this category. The third category

1049-9660/94 $6.00

includes high level problems, which require symbolic reasoning from sensory data. Path planning, collision avoidance, autonomous navigation, and sensor guided assembly belong to this category. Very often, although not necessarily, the higher level problems require the results of lower level problems as input. Performance characterization for these three categories should be different since their solutions and objectives are quite different in nature.

2. SUBPROBLEMS AND DATA GENERATION

Arguably, computer vision currently is still mainly an area of research rather than applications. Often, the vision tasks are broken into subproblems and each is investigated separately, under the assumption that the results from lower level modules are available. But the level specific nature of these subproblems complicates their performance characterization. For example, the random perturbation of input data might not be realistic for simulating the imperfection of the input, since the actual error in the input, which is often an output of another algorithm, is highly correlated with the input itself. For instance, the error in stereo matching is usually closely related to the depth discontinuities and occlusions. Random noise in the input depth map may be easy to deal with by imposing certain types of smoothness, but the error along depth discontinuities and near occluded regions cannot be dealt with effectively by the same smoothness constraints. An algorithm which performs well under random noise may perform poorly on real input data. Therefore, ideally the performance characterization of an algorithm should be based on actual input and its intended use, together with the precedent and subsequent algorithms.

For many engineering systems, the specification of the total system can be met by imposing specifications on each subsystem. However, in computer vision, the performance of a vision algorithm can be so scene-dependent that often there exists no proper set of parameters that can usefully characterize the performance of a subsystem. For instance, consider an edge detector whose result is to be used by some edge-based recognition systems. The success of the detector may depend very much on the image content or the objects in the scene. It may detect every edge in one context and miss most of the edges in another. Although one can come up with an average detection rate based on a set of sample images, this detection rate may tell little about how successful an edge-based recognizer can be which uses the result from the edge detector. An artist's line-drawing rendition of a natural scene is very different from the output of a Laplacian-of-Gaussian edge detector applied to the same scene. The former preserves and links most identify-informative edges (those that are informative for identifying the objects in the scene) and neglects the rest; while the latter gives just intensity edge curves that run from one object to another. Therefore, an edge detector with a smaller detection rate (the artist, as in our example, or other good edge detectors designed for recognition) may allow a better recognition than one with a larger detection rate (Laplacian of Gaussian in our example). But an identity-informative edge is not well-defined. Consequently, it is difficult to impose proper specifications on an identity-informative edge detector and evaluate its performance.

So, the performance characterization for subproblems is very complex. Imperfection of input data is often scene dependent, and random perturbation might not be a suitable model for modeling the imperfection. While the solutions to some relatively simple vision problems may involve only a few numerical estiamtes or detection flags, the solutions to many other problems are in more complex forms, such as object segmentation, uncertain recognition, scene understanding (description of the scene), prediction, knowledge representation, planning, and mission specification. The quality of the outputs, either numerical or nonnumerical, are not always characterized by some types of statistical average. Furthermore, the relationships among subsystems are both nonlinear and scene dependent, thus, propagation of even random errors through the total system is in most cases very complicated. As a result, except for some simple problems, the goal of predicting the total vision system performance from subsystem performances may be unachievable.

3. THE NEED FOR PERFORMANCE CHARACTERIZATION

Despite the complexity of the computer vision algorithms and the difficulties in their performance characterization, computer vision algorithms cannot do without some performance characterization.

Performance characterization is important in order to identify under which conditions the algorithm gives good result and under which it does not. We must deal with a wide variation of real world conditions, including variations in lighting source, lighting geometry, viewing position, surface optical property, surface geometry, and object types. An algorithm that works in one situation may fail in another. A valid algorithm must clearly identify the conditions under which the algorithm performs normally and the quality of the solution under these conditions.

Performance characterization is also useful in establishing the value of a new algorithm. We may have seen many algorithms that perform the same task. For a new algorithm, one must clearly demonstrate its performance and the advantages over existing algorithms. Without knowing which algorithm is better, it is very difficult for a practitioner to select an appropriate algorithm.

Performance characterization can sometimes provide

insight into the problem itself, not just the particular algorithm under consideration. For example, one may wish to obtain a good estimate of the motion parameter from a sequence of low resolution images, say 64 × 64. But the digitization noise has imposed a theoretical lower error bound that itself is larger than the tolerable error. We can know what is the best one can possibly do by conducting the performance characterization of the algorithm that is known to have nearly reached the bound.

Performance characterization of all the algorithms for a particular problem may indicate clearly the state of the art of the problem, not just a few ad hoc examples about something doable.

4. FACILITATION OF PERFORMANCE CHARACTERIZATION

For most low-level problems, one of the key measures of the performance is the stability under image noise, and the quality of the solution can be characterized by some numerical measurements. The protocol proposed by Haralick can be used to conduct performance evaluation for these problems. But it should be noted that the criteria used for characterization should be suitable for the intended application.

The major concern of middle and high level problems is not necessarily noise immunity. Normally, successful recognition from good real images is sufficient for most applications. The main concern could be the success rate, or other appropriate measures, under different lighting conditions, viewing positions, backgrounds, object types, etc. A large number of these factors result in a huge space from which a small number of sample images are to be generated. The selection of the sample images from this huge space is inevitably either subjective or accidental.

As discussed above, it is difficult to generate suitable input data sets for subproblems. Synthetically generated data with random noise contamination are often far from what one actually obtains in a real world situation. It is not always possible for every researcher to generate realistic input data from original images.

Therefore, though performance characterization is an extremely difficult task. To promote performance characterization, we need not only protocols, but also means that facilitate such characterizations. The heavy burden of sample image generation, imperfect input data generation, and performance criteria selection should be removed from individual researchers as much as possible so that the individual cost of conducting performance characterization is not so formidable. This can be made possible by establishing the following channels that facilitate performance characterization in the computer vision community.

1. *Image sharing*. A set of images can be collected from different groups and be entered into an image database available to the public. It is encouraged to use images from this database for various experiments. The results from different algorithms can be compared and the performances evaluated. The selection of the images in the image database should take into account the permissible range of image generation variables. The *1991 IEEE Workshop on Visual Motion* has organized a public image sequence database for image sequence analysis. Such a practice may be extended to other computer vision problems.

2. *Result sharing*. The results from different groups can be organized into a public result database. The results are indexed according to the images in the image database and the tasks to be performed. These results can be used by others as input for their subsequent algorithms. They are also useful for comparison with other new algorithms that perform the same task. Without such result sharing, the comparison requires independent implementation of others' algorithms, which may compromise the fidelity of the original algorithms. Currently, in the computer vision community, result sharing is conducted only occasionally between groups. A coordinated public result database may greatly facilitate and expand such collaborations.

3. *Program sharing*. Various programs from different groups can be collected into a program library. Those programs can be used by other researchers to generate the input data they need. The use of these programs may be restricted to academic research only and not for commercial use. Certain other limitations may also apply, e.g., the authors' permission and participation in any comparison that is intended for publication. Currently, various program giving-away is happening among various academic groups. Most of those programs are used to bridge the gaps in computer vision experiments. By doing so, one can conduct performance evaluation from real data without having to write programs to generate the input data needed.

4. *Open competition*. Every group that has published the algorithms that perform the same task is encouraged to participate in the open competitions for the task. To eliminate human intervention, every participating group is required to submit its program to an organizing committee. The committee independently selects a set of test images, runs every algorithm on these images, and evaluates the results according to some predetermined rules. Open competition is probably the most objective way of conducting performance evaluation. Computer chess has had similar competitions for quite some time, and the computer vision community can do the same.

Facilitating performance evaluation is one of the key factors for the success of promoting performance evalua-

tion. We think that the lack of performance evaluation in computer vision has not so much to do with the understanding of its importance as with the difficulties and the lack of facilities that may otherwise make the task more tractable.

5. CONCLUSIONS

Serious discussion on performance characterization in computer vision is indeed overdue. In our view, the lack of performance evaluation is mainly due to various difficulties peculiar to computer vision. An important characteristic of performance evaluation for computer vision algorithms is that there are a large number of parameterizable and nonparametrizable factors that need to be taken into account. The subproblem status of many vision algorithms make a thorough performance characterization very difficult and time consuming. In order to translate our awareness of performance characterizatiaon into everyone's action, we need to facilitate performance characterization. We propose the establishment of public database facilities which allow free sharing of data, results, and programs. Objective performance evaluation should also be conducted in the form of open competitions.

REFERENCE

1. R. M. Haralick, Performance characterization of computer vision, *CVGIP: Image Understanding* **59,** 1994.

CVGIP: IMAGE UNDERSTANDING
Vol. 60, No. 2, September, pp. 257–259, 1994

REPLY

Computer Vision: The Goal and the Means

PETER MEER

Department of Electrical and Computer Engineering, Rutgers University, P.O. Box 909, Piscataway, New Jersey 08855-0909

In his intriguing position paper Haralick proposes an engineering approach toward the development and validation of computer vision algorithms. He emphasizes that the reliability of these algorithms (and therefore the maturity of the field) cannot be achieved without using quantitative methods at every stage of design. Intuition and luck should be replaced by methodology and inference.

We can only agree with the ideas put forth in the position paper. Changes in our way of approaching computer vision problems are certainly needed. However, we must also examine the present state of the field and evaluate whether or not it is possible to advance in one "great leap forward." The complexity and diversity of the visual input is extremely challenging and we may not yet understand all the implications arising from its discrete nature. To make our point clear we discuss the problem of random perturbations. Along the presentation some of the conclusions are spelled out as questions. In our opinion satisfactory answers to these questions are a prerequisite for achieving reliable universal computer vision algorithms.

Central to any engineering procedure is the access to ground-truth, i.e., to a known input–output relation. The ground-truth allows the designer to compare the output of an algorithm with the expected correct result. Understanding the discrepancies between the obtained and desired output helps to improve the algorithm to achieve better performance. Discarding all the unrelevant components from the input–output relation often reduces the ground-truth to simplistic data. In agreement with Haralick we assume that for such ideal data an algorithm has its best possible performance. The real world, however, is almost never perfect and we must take into account the presence of random perturbations corrupting the ideal input. These perturbations (noise) are classified into two categories that are functions of their effect.

A task is an algorithm performed on a given data set. The data can be an image, an ensemble of extracted features, any quantitative description. The task is satisfyingly executed whenever the performance of the algorithm exceeds a bound. The precise definition of this bound is not relevant for the discussion; what is of importance for us is that the performance depends on the available data. Every data point in the set carries information which either aligns or diverges from the assumptions embedded (maybe only implicitly) in the algorithm. We can also include in the latter category all the points having no influence at all on the algorithm's performance. This is justified by the fact that an increased support usually reduces the spread of the estimated quantities around their correct value and thus improves the performance. A simple example: recovery of a parametric model from nonhomogeneous data. Data that can be derived from the model is considered helpful for the performance of the algorithm. Data not accounted for by the model is considered hostile for the performance of the algorithm.

The customary taxonomy of noise processes (also used by Haralick) is similar but not identical with a purely task-dependent classification. The Type I noise yields small perturbations and corrupts every data point. However, Type I noise will not change the status of a point from helpful to hostile. (A change in the other direction is not relevant for the performance of any algorithm!) If we model the small perturbations with distributions which do not exclude large deviations (e.g., zero-mean Gaussian processes) the status of a corrupted data point may change with low probability.

Performance of an algorithm decreases with the increase of Type I noise (increase measured by some parameters of the noise distribution). The worsening of the performance, however, is reflecting mainly the sensitivity of the algorithm due to the relative small data size available in computer vision. Should a very large amount of data be available the influence of Type I noise on the performance of a good algorithm (producing unbiased and efficient estimates) can be close to its upper bound. This bound is determined by the algorithm and the available data, i.e., by the performed task, and cannot be exceeded. We can ask ourselves:

> Do we know how to describe the influence of small data size on the performance of computer vision algorithms?

1049-9660/94 $6.00

The Type II noise is present only in a subset of the data but the corrupted data points change their appurtenance from helpful to hostile for the performance of the algorithm. The corrupted points are also called as outliers. At least theoretically an ideal image can become another ideal image when corrupted by Type II noise. In this case, while the algorithm performs well, execution of the task fails. To make this clear lets use an extreme example. Assume that we are looking for a circle in a binary image with a matching algorithm which can recover circles and ellipses. The ideal image contains a circle but in the corrupted image due to the outliers we have a shape close to an ellipse. The algorithm alone will not be able to recover from the error. Additional information is needed, either available a priori or through a top–bottom component incorporating more general knowledge. What is the conclusion of this toy-example? Open-loop analysis of computer vision algorithms may not tell the whole story when Type II noise is present. This type of perturbation can alter significantly the semantic content of the data and a "narrow-minded" algorithm may not be able to discriminate the original data any more. We can ask ourselves:

> Do we know how to separate the "low-level" and "high-level" component of a task in an optimal way?

It was recognized long ago that the hardness of many computer vision tasks stems from Type II noise corrupting the data. Whenever the data contains more than one class (as defined by some homogeneity criterion) and is analyzed by an algorithm not specifically designed to handle several classes, we can regard all data points belonging to the nonmajority classes as outliers, i.e., the majority class corrupted by Type II noise. The influence of outliers on the performance of an algorithm not able to deal with Type II noise can be disastrous. Recall the result of any least squares based procedure applied to data containing a step-discontinuity.

Robust algorithms which can handle data corrupted by Type II noise were recently developed in statistics and adopted in computer vision. These algorithms recover parametric models by dichotomizing the data into a majority (the "good" part) and a minority (outliers). The philosophy behind all outlier-resistant robust algorithms is similar. First, randomly select a subset of the data points large enough to compute the values of the sought estimates. These estimates are then assumed to be valid for the entire data set, i.e., to represent the underlying model. A quality measure is computed to characterize the deviation between this model and the "good" data points. The whole procedure is repeated several times and the estimates corresponding to the extreme value of the quality measure are retained as the output of the algorithm. The extreme quality measure value corresponds to a subset containing only "good" data points and therefore carrying the correct model. Note that the size of the data is not an important factor in the performance of the outlier-resistant robust algorithms.

So if we have enough data points and Type I noise we can expect good performance; if we have Type II noise we can borrow techniques from statistics. Unfortunately in real images we have both types of noise! Type II noise appears whenever the data is nonhomogeneous, while Type I noise is generated by measurement inaccuracies. The influence of Type I and Type II noise cannot be separated when the performance of an algorithm is studied. The robust procedure overcoming Type II noise works only if the "good" data is close to ideal. Indeed, when Type I noise is also present, none of the randomly selected subsets can carry the correct model. The employed quality measure is no longer reliable since now it also incorporates a random component. The performance of the robust algorithm decreases drastically even for small Type I noise since the algorithm loses its resistance to Type II noise. We can ask ourselves:

> Do we know how to build algorithms which have acceptable performance in the presence of both noise types?

Haralick proposes sampling the high-dimensional space of all the possible images related to a general task. From the ensemble of the outputs the receiver operator characteristics or the average deviations from the ground-truth should be computed. The performance is then characterized by the dependence of these quantities on the input parameters. There is no doubt that the proposed performance measures are the desired ones; however, the amount of data required for their computation appears to be prohibitive for real images. Can we obtain less rigorous measures with less pain? Maybe. Computer-based error analysis techniques like cross-validation, bootstrapping, jackknifing are very popular in statistics. They use only a few samples of the input data to compute confidence intervals for the estimated quantities, that is, a measure of accuracy for the output of the algorithm.

The data in statistics is much less complex than images and therefore it is not clear if the abovementioned techniques can be successfully applied to measure the uncertainty about the output of a general computer vision algorithm. Should this be possible the performance of an algorithm for a given input can be characterized by the confidence we have in the output. If the confidence is high enough, i.e., the range of probable output values allows a unique interpretation, the algorithm can be considered reliable.

The proposed performance measuring protocol assesses reliability for a specific input or for a very narrowly defined set of inputs. To validate an algorithm over a larger family of inputs (say, recovery with the same algorithm of different objects laying on different backgrounds) we must

first define what the output uncertainties are that we can tolerate when correct classification is reuired. Then, if the computed confidence intervals are within these limits, the algorithm passes the test. The protocol recalls pattern recognition. At the dawn of computer vision history the official paradigm was indeed that of pattern recognition. Later its importance diminished. However, maybe it is time to reexamine the usefulness of pattern recognition principles for computer vision. The arguably most successful computer vision algorithm, the Hough transform, is also pointing toward this.

In conclusion we are in agreement with the majority of ideas in Haralick's position paper. Nevertheless, we wanted to attract attention on two issues. First, the discrimination of the signal from noise in visual data is a delicate and difficult problem. Second, restricting performance measurement to the given input may make algorithm validation easier.

CVGIP: IMAGE UNDERSTANDING
Vol. 60, No. 2, September, pp. 260–261, 1994

REPLY

Performance Characterization in Computer Vision

YOSHIAKI SHIRAI

Department of Mechanical Engineering for Computer-Controlled Machinery, Osaka University, Suita, Osaka 565, Japan

In his paper, Haralick points out that computer vision algorithms have been published more on the merit of an experimental or theoretical demonstration suggesting that some task can be done and that performance characterization is necessary for algorithms in order to be used in practical engineering. He also proposes a general method of performance characterization.

I agree with the necessity of performance characterization of an algorithm with perturbation of the input images and that of the algorithm parameter in order to show the robustness of the algorithm. His suggestion, however, might be applied only to a limited class of algorithms. We should consider the evaluation of a computer vision system separately from the algorithms used in computer vision systems.

EVALUATION OF THE TOTAL SYSTEM AND THE COMPONENTS

The performance of a total vision system can be evaluated rather easily because the purpose of the system is clear. For example, a surface defect detection system for steel milling is expected to work for input images of steel sheets which are characterized by some parameters and perturbation. The error criterion may be easily defined because the desired output can be determined.

However, the purpose of characterization is not for a total vision system, but for algorithms to be used for constructing a total vision system. Then the problem of characterization is quite different from the case of a total system because none of the above characteristics can be uniquely determined. Let us take an edge detector as an example. Suppose that the edge operator is designed on the basis of a certain property of a light intensity profile in an image and that the edge is detected as a ridge of the output of the edge operator. One important problem is now to determine the error criterion. Since the edge detection is performed according to a firm criterion, the detected edge itself is correct if we admit the definition of the edge. However, if the result of edge detection is used for the detection of defects, for example, then the edges may not always correspond to the defects. Is it a responsibility of the edge detector or the later algorithm which selects the defects among the detected edges.

If a noise is added for the perturbation of an image, new edges may be created by the noise. Then a question arises whether or not the edge detector should detect the edges caused by the noise or should avoid such edges because they do not correspond to a real defect?

As shown in this example, evaluation of lower level vision (or early vision) without any particular purpose is difficult by the suggested performance characterization method. In general, the expected value of the error cannot be obtained as a function of the algorithm parameter and the perturbation variables.

EVALUATION BY HUMAN

Objective evaluation of an early vision algorithm is difficult without specifying the purpose of a total system which includes the algorithm. One possible way is to compare the performance of an algorithm with that of human visions.

It is desirable for an algorithm to produce an output similar to human vision. When a total system is designed as a combination of component algorithms, the performance of each algorithm is predicted. The predicted performance of an early vision algorithm may be similar to the performance of human vision. If the performance of the algorithm is different from the predicted one, the performance of the total system may also be different (often lower) than the predicted one.

The variance of human vision characteristics is much smaller than the gap between the characteristics of human and machines. The color distance, for example, is determined by human subjective experiments. Although it cannot always be applied straightforwardly to all applica-

1049-9660/94 $6.00

tions, it is still useful as a criterion of an intermediate color segmentation.

A problem with the human subjective judgement of an early vision algorithm is the use of semantics by human beings. If an input image is a real image of a 3D scene, a human being may interpret the image before evaluating the performance, just like the case of edge detection given images of steel with defects.

To avoid the effect of semantics, we should make artificial images which do not correspond to real scenes. Computer graphics techniques might be useful for synthesizing many images by changing the parameters which specify the image. For each synthesized image we can prepare the result of human image processing. Of course, in order to make a convincing result of human image processing, experiments with human subjects must be performed which may require a great deal of labor.

CONCLUSION

Although the suggested method of performance characterization cannot be used for general computer vision algorithms, it may work for certain algorithms. One such class of algorithms are the algorithms with binary input images (such as thinning algorithms) because the variation of the input image is not very large. Another example is stereo matching algorithms because a stereo pair of images can be created from a known or synthesized 3D description of the scene. In fact, many researchers of stereo matching algorithms wish to use sample images with the correct range data to evaluate and improve their methods. We should begin with those possible and useful cases. Again, a practical method of perturbation is indispensable, as pointed out by Haralick, for evaluating robustness of an algorithm.

CVGIP: IMAGE UNDERSTANDING
Vol. 60, No. 2, September, pp. 262–263, 1994

REPLY

Response to "Performance Characterization in Computer Vision"

BRUCE A. DRAPER AND J. ROSS BEVERIDGE

Department of Computer and Information Science, University of Massachusetts, Amherst, Massachusetts 01003

We commend the author for raising again the issue of performance evaluation, and in particular for raising it in the context of complex computer vision systems constructed from multiple interacting components. For years researchers have been dividing vision into a series of (hopefully simpler) subproblems and designing algorithmic solutions to them. The better we understand how these algorithms behave, the more success we will have at assembling complex systems. Nonetheless, we caution those who would embrace Professor Haralick's strict methodology that it assumes conditions that cannot always be met.

The first assumption is that the errors and/or distortions in the input data are well understood and can thus be formally modeled. Haralick's performance characterization technique requires many carefully controlled inputs with accompanying ground-truth values. This level of control can generally only be achieved through artificial data sets (including but not limited to synthetic images) created by adding errors and/or distortions to idealized world and camera models. Although Haralick's methodology relies on such artificial data, he fails to mention that the error and distortion processes used to generate this data must be statistically validated. Even with statistically validated data models the accuracy of the performance characterization is limited by the accuracy with which the synthetic input data matches real data.

It is therefore possible, using Haralick's methodology, to specify performance models that are much more precise than accurate. It allows researchers to present detailed analyses of an algorithm's behavior on synthetic data without ever acknowledging that their input model is at best a crude approximation of the real world. The resulting performance models are precise in that they provide detailed predictions of output errors, but inaccurate in that their predictions do not match the algorithm's performance on real data. Moreover, such precise but inaccurate models can be harmful to the field, in part because they feign more accuracy than they possess and in part because their acceptance encourages researchers to adopt error models based on ease of analysis rather than fidelity to real data.

Haralick also assumes that object kind and pose have only a "minor effect" on the performance of a computer vision algorithm. Unfortunately, this assumption does not hold for many common algorithms. For example, object type is generally important to the analysis of matching algorithms, which confuse similar object types more often than dissimilar ones. In the same vein, the analysis of pose refinement algorithms depends on the discrepancy between an object's actual pose and the estimated pose supplied as input.

By disregarding object kind and pose, performance evaluation is limited to a subset of computer vision algorithms. Historically, computer vision research can be divided into two paradigms: *computational vision* tries to reconstruct the 3D geometry of a scene while *knowledge-directed vision* tries to match the contents of a scene to models in memory. Most computational vision algorithms can be analyzed without reference to object kind or pose since they attempt to recover spatial properties by reasoning about geometry or physics. Knowledge-directed algorithms, on the other hand, match image data to object models and their performance depends critically on the quality and kind of available object models.

Moreover, the only apparent reason for precluding object kind from performance characterization is that there is no consensus as to what variables should be used to describe object knowledge. Ignoring knowledge-directed algorithms avoids having to specify the relevant features of a model base (e.g., similarity between models, symmetries within models, variability between instances of an object class). Unfortunately, it also limits the scope of the methodology to computational vision.

Overall, this paper emphasizes the importance of performance evaluation in computer vision and gives a methodology by which precise performance models can be derived. We must remember, however, that this method-

1049-9660/94 $6.00

ology can only be applied when validated models of the errors and distortions in the input data are available and when there is a general consensus as to which scene variables need to be controlled and modeled. When these conditions are met, the performance evaluation methodology outlined by Professor Haralick is both appropriate and desirable. However, when these conditions are not met the best evaluation technique is still an empirical investigation of an algorithm's performance on real data.

CVGIP: IMAGE UNDERSTANDING
Vol. 60, No. 2, September, pp. 264–265, 1994

RESPONSE TO REPLIES

Comments on Performance Characterization Replies

ROBERT M. HARALICK

University of Washington, Seattle, Washington 98195

There are a few issues for which it is appropriate to make comment. One is that knowing the performance of one stage of an algorithm will not permit one to propagate this performance to the next stage. This point was raised by Cinque *et al.*, Weng and Huang, and Shirai.

They raise this issue because they do not fully understand the performance characterization position. Shirai's reply has the most detailed comments. He gives the example of knowing the performance of an edge detector and relating that to the performance of defect classification, of which edge detection may be one step. Defects, for example, may not always correspond to edges and edges may not always correspond to defects.

Shirai's question comes to asking how the misdetect and false alarm characteristics of the defect detector can be determined from the misdetect and false alarm characteristics of the edge detector. The answer is that it can be determined in a way exactly analogous to that in which the performance of the edge detector can be characterized. The step after edge detection, whatever it is, has a performance relative to the random perturbation of the idealized data that it inputs. Indeed it is the case that the output of the edge detector is not ideal. But once we can describe this random perturbation in terms which are relevant to the next processing step, then everything regarding performance characterization is analogous to what happened in characterizing the performance of the edge detector.

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 a 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 detection algorithm parameter values are 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 edges which are 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 its true direction and the edge direction distribution for each falsely detected edge.

Suppose that the next operation is a spoke filter. Then utilizing the information from 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 declares a defect if the maximal count is great enough. Now from the distributions of counts of defect and non-defect pixels, it should be possible to compute the misdetection and false alarm characteristics of the final defect detection step.

1049-9660/94 $6.00

And this characterization will be a parametric characterization with parameters consisting of the Poisson density parameters, the background brightness, the defect brightness and size, and all algorithm turning parameters.

Cinque *et al.*, Weng and Huang, and Draper and Beveridge raise a second issue: the issue of realistically modeling random perturbations. This issue is important because if the random perturbation models are not realistic, then to the degree that they are not realistic, the performance characterization will be meaningless. In the way they raise this issue, however, there is almost an implication that since whatever perturbation model one might use is certainly not realistic, there is no point in developing a performance characterization theory using it. So we should better spend our time working with heuristically developed algorithms applied in real data experiments and not spend any time on performance characterization.

This position has a fundamental flaw which can be seen by considering that it entails a commitment to developing algorithms. We understand that a commitment to developing algorithms means that we want to develop good algorithms, reliable ones, ones that work in the face of the real random perturbations to which the data are subject. Now once an algorithm is stated, there is an implied class of random perturbations on the input to which the algorithm is suited. Often this class of random perturbation models can be inferred by a sort of reverse statistical engineering of the algorithm. So committing to the development of an algorithm and then developing the algorithm implies an unconscious selection of a random perturbation model for which the algorithm produces good answers. The point raised by the performance methodology protocol is that this selection of a random perturbation model should not be an unconscious selection. It should be a conscious selection, for once the selection is in consciousness, then it becomes possible for the rational intellect to work with it and thereby develop algorithms which are optimal rather than being heuristic and suboptimal.

There is one more dimension to this issue, which Draper and Beveridge raise. They say that to make sure that the perturbation models are realistic they have to be statistically validated. Indeed that is true. Not only must they be validated, but the free parameters of the random perturbation model must be estimated. And it is the case that nothing was mentioned in the initial dialogue about parameter estimation and validation. So to correct that omission it must be asserted that the entire performance characterization methodology involves parameter estimation and validation of random perturbation models.

This of course 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 gound-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.

Finally, Shirai makes the comment that it is easy to evaluate the performance of an existing algorithm in an existing application, so why all the fuss on performance characterization. The answer is that it is important for the machine vision engineer to be able to predict the performance of a vision algorithm before it is tried on the factory floor. It is important for the machine vision engineer to be able to analyze the performance of a machine vision algorithm step by step to determine where effort should be put to improve the performance by using more optimal values of algorithm tuning parameters or a different algorithm step. It is important for the machine vision engineer to be able to set the algorithm running parameters to their optimal values based on the estimated parameters of the random perturbation model(s) without an experimental trial and error procedure.

In summary, performance characterization is not only applicable to low level vision. It is applicable throughout low level, mid level, and high level. Indeed it is the case that when it is applied to high level, the kind of control that high level needs to exert on mid and low level will become apparent—not as a heuristic, but as what optimally needs to happen. What performance characterization does is to take the subjective free play out of computer vision and to replace it with sound engineering systems analysis and synthesis. It replaces the fancy buzz words and buzz techniques with the kind of soundness which characterizes all the successful areas of engineering. One must remember here that engineering systems can be quite complex. Perhaps the most complex engineering system designed and built and which is in operation is more complex than the most complex computer vision system built up to today. Perhaps the success in having such a complex engineering system working is due to each module in it having a performance characterization which was utilized in the design analysis and synthesis process.