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.