Adaptive Image Processing Techniques

James S. J. Lee, Paul V. Budak, Charlotte R. Lin, Robert M. Haralick*

Boeing High Tech Center, P.O.Box 24969, MS 7J-24, Seattle, WA 98124-6269
*EE Dept, FT-10, University of Washington, Seattle, WA 98195

ABSTRACT

Adaptive image processing schemes can be classified as open-loop, input sensing, invariant-expectation, and model reference systems. Two major adaptive image processing system mechanisms, processing status measurement and parameter adjustment, are described and a multi-resolution approach is developed. The multi-resolution schemes allow efficient adaptive image processing implementation, by enabling coarse-to-fine parameter (operation flow) adjustment in both image and parameter domains.

The adaptability and robustness of these techniques is demonstrated on morphologically segmented objects from actual laser radar (range) data.

1. INTRODUCTION

A successful computer vision system for unconstrained environments must be able to perform algorithms in a robust and adaptive fashion, relying on a balance between a high-level intelligent rule-based scheme and low-level adaptive image processing.

Many rule-based intelligent vision schemes appear in the literature (Hanson & Riseman (1978, 1984), McKeown, et. al. (1984), Brooks (1981), and Davis et. al. (1985)), while adaptive low-level schemes are not fully explored. We introduce adaptive image processing schemes which can support dynamic operation flow and processing parameter determination. A typical operational stage in an adaptive image processing system has an expectation of its output and of the initial values of its dynamic adjustable parameters. During processing, an image status measurement unit is employed to monitor the result of the current process. The difference between the measured and expected results is used by the adaptive mechanisms. A significant difference may cause reapplication of the current operation with different parameters or abandonment of the current process and diversion of operation flow. Current processing status feedback is used to dynamically determine the expectations for other operation stages.

The adaptive image processing approach is intended to complement rather than compete with rule-based high-level vision systems. In a vision expert system a request for low-level processing can be represented in a backward chaining fashion by posting a goal (see Kohl et. al., 1987) which results in the instantiation of an adaptive image processing process. The processing expectations, initial parameters, and selected status measurement method can be expressed in the specifications of the goal representation generated by the high-level process. The adaptive image processing stage takes the goal data and dynamically approaches the desired goal. In this way we permit the low-level processes to achieve high-level goals without explicit high-level control. Hence, the burden of adapting low-level processes in response to changes in input data will not be solely placed on slower high-level processing.

In this paper different adaptive image processing schemes are introduced. Image status measurements and automatic parameter adjusting mechanism are reviewed. Moreover, a multi-resolution approach is developed to achieve real-time performance.

The adaptability and robustness of these techniques is demonstrated on morphologically segmented objects from actual laser radar (range) data.

2. ADAPTIVE CONTROL FOR IMAGE PROCESSING

Adaptive control is a mechanism to automatically compensate for unforeseen changes in system parameters and input signals (Ng (1969), Eveleigh (1967)). Special techniques are applied to ensure optimum performance or maintain invariant system response under changing conditions. Applied initially in aircraft and missile control systems, adaptive control can also be applied to image processing and machine vision systems. Similar to conventional adaptive control systems, adaptive image processing systems can be classified into four categories:

Open-loop adaptive systems

Input sensing adaptive systems

Invariant-expectation adaptive systems

Model reference adaptive systems

An automatic image processing system normally includes several adaptive image processing subsystems and certain intermediate processing status test stages. The subsystems may fall into different adaptive processing categories. A global control is used to interface with high-level vision controls coordinate subsystems.

2.1. Open-loop adaptive systems

The underlying philosophy of these systems is similar to the top-down control philosophy of knowledge-based high-level vision systems. The control flow of a typical open-loop adaptive controller is shown in Figure 1. An operating condition knowledge base is set up at the design stage, including a set of parameters (operations) grouped to provide the best image processing under different conditions. In operation, a specific parameter (operation) is selected corresponding to prevailing operating conditions. This technique suffers from the following disadvantages.

a. Prior knowledge of operating conditions must be known accurately.

b. Scene (target, background, etc) characteristics at each of these operating conditions are required.

c. A means of detecting when conditions have changed has to be incorporated in the controller.

d. The design problem is exacerbated by the need to design for a large number of operating conditions.

e. The finished controller is at best only optimized for the design conditions studied.

Condition (c) requires a measurement from which to make an estimate of operating conditions. There is no feedback to modify this estimate, even if actual conditions differ so that the control is unacceptable. (a), (b), and (d) involve additional time and expense in production of the controller. These systems can generally adapt to changes in one parameter only and may be extended to multi-parameter control only with considerable increase in analytical effort and system complexity. Moreover, if a current processing result is used to hypothesize the operating conditions for other operations under similar control, one erroroneous processing result may cause side effects (e.g. falsely fire new inapplicable rules and cause further confusion, or falsely poison applicable rules) from which recovery, even if possible, may take a tremendous effort. Nevertheless, an open-loop scheme may be very effective in simple, well controlled industrial machine vision applications.

2.2. Input sensing adaptive systems

Input sensing adaptive systems are designed to compensate for changes in characteristics of the input image data, as shown in Figure 2. An input image characteristic is measured and the result used to adjust operational parameters of the image processing operation. These systems invariably have open-loop structures and cannot therefore compensate for features other than the ones for which they have been designed. To implement this scheme, the characteristic analysis mechanism is designed and incorporated as part of the image processing operation design. Examples of this scheme include automatic image thresholding (using grayscale histogram as input characteristic measurement) and automatic background normalization (to be discussed in section 6).

2.3. Invariant-expectation adaptive systems

In the invariant-expectation adaptive image processing system, parameters are adjusted to achieve a pre-specified processing expectation despite the lack of knowledge about the input data. In one scheme, shown in Figure 3, an initial operational parameter is applied to the input image data and the output is measured. The output image is fed into a performance evaluator which in turn adjusts the operational parameters until the specified expectation is obtained. If the specified expectation can not be obtained by the current image processing operation, the current stage is abandoned and operation flow may be diverted. Current processing status feedback can also be used to dynamically determine the expectation and initial parameters for other operation stages. Such a parameter adjusting mechanism is described in more detail in section 4. The morphologic iterative noise filtering scheme described in section 6 is also an example of this system. Since the control scheme involves iterative application of the same operations, the computational cost can be reduced by applying a coarse-to-fine multi-resolution scheme. The ideal is to estimate a good initial operational parameter by performing the scheme at coarse resolution. A good initial parameter allows the operational parameter to converge rapidly (see section 5).

2.4. Model reference adaptive systems

In model reference systems, a model of the ideal image data or a simulation noise free data is subjected to the same operation as the input data. Processing results are compared for the system and the model. Operational parameters are adjusted if the results significantly differ from the model. A more detailed description of this kind of adaptive control is given in section 4.

A typical model reference adaptive system is shown in Figure 4. One possible implementation is the pattern recognition training-set method of using a known part of the image as a model and adjusting the operational parameters by comparing results from the input and the model. We can then apply the derived parameters to the whole image in order to extract the unknown image features. This scheme is especially effective in industrial automatic machine vision applications, for repetitive or regular images as in electronics inspection.

3. PROCESSING STATUS MEASUREMENT MECHANISM

Adaptive processing systems make image status measurements to evaluate current image status for use as current processing status feedback. A good image measurement should reflect clearly the current processing status; availability of image status measurements affects processing design. Conversely, image processing operations also motivate the provision of certain useful image status measurements. In the following we give a brief review of existing image status measurement schemes (see Lee, 1986) which are useful for processing status measurements.

Status measurements for an image include Euler number (genus), dominant feature covariance directions, connected components labeling, homotopy tree extractions, image projections in different directions, image histogram, shape or size distribution, edge image figure of merit, and global feature image area/ centroid/ orientation/ tonal properties (mean, skewness, kurtosis, etc). Features can be extracted for each connected region of an image, such as area, centroid (center of mass), bounding rectangle or convex hull, extreme points, fitting ellipse axis lengths/ orientation/ elongation, grayvalue maximum/ minimum/ mean/ variance, perimeter, color eccentricity, corners, angle statistics, angular variability, shape signature, region compactness, circularity, genus, region histogram, and region texture measurements.

The measurements provide an abundance of information about the current image status which in turn either discloses the effectiveness of the current process or suggests a new processing scheme for the image data. Euler number provides a simple binary image complexity measure; dominant feature covariance directions disclose the dominant orientations of the image structure; connected components labeling provides information about both the image as a whole and each individual connected region; image projections provide image status sensing in any given direction; image histogram tells the grayscale distribution of the image, shape and size distributions give an image structure measure. Combining measurement information and relating it to image processing status is a crucial part of any good adaptive image processing scheme.

4. PARAMETER ADJUSTING MECHANISM

As mentioned in section 2, an efficient parameter adjusting mechanism is essential to both invariant-expectation and model reference adaptive image processing systems. In this section, we discuss these kinds of parameter adjusting mechanisms.

Corrective parameter adjustments normally require a measure for the parameter index $f(e)$ to change in the parameter k, $\frac{\partial f(e)}{\partial k}$ where $f(e)$ is a function of the difference e between the expectation and the image processing result. One technique, called hill-climbing, adjusts the parameter by a trial-and-error technique to the optimum setting corresponding to an extremum of $f(e)$.

A perturbation technique can be employed to determine $\frac{\partial f(e)}{\partial k}$. The response difference e is the difference between the fixed expectation and the image processing result, which is a function of parameter k. Therefore a performance index $f(e)$ can be specified as a function of k, $f(k)$. Assume $P = f(k)$. When positive-negative perturbations $+\delta, -\delta$ are applied, we have

$$P^+ = f(k+\delta) = f(k) + \frac{\partial f(k)}{\partial k}\delta + \frac{1}{2}\frac{\partial^2 f(k)}{\partial k^2}\delta^2 + \ldots$$
$$P^- = f(k-\delta) = f(k) - \frac{\partial f(k)}{\partial k}\delta + \frac{1}{2}\frac{\partial^2 f(k)}{\partial k^2}\delta^2 - \ldots$$

Since

$$P^+\delta - P^-\delta \approx 2\delta^2\frac{\partial f(k)}{\partial k}$$

these results,

$$\frac{\partial f(k)}{\partial k} \approx \frac{P^+\delta - P^-\delta}{2\delta^2} = \frac{P^+ - P^-}{2\delta}.$$

The performance index may be minimized by steepest descent techniques, making incremental changes in parameter error so as to maximize decrease in the performance index magnitude at any point. This may be accomplished by choosing these incremental changes proportional to the negative gradient of the performance index function at the operating point. Thus,

$$\triangle k = -g\frac{\partial f(k)}{\partial k}$$

where g is a constant.

5. MULTI-RESOLUTION APPROACH

In an adaptive image processing system, image operations and image processing parameters are determined during execution and several iterations may be required to achieve a satisfactory result. Thus, we pay for adaptability and robustness by increasing computation time and/or hardware complexity. A multi-resolution approach is used to reduce this cost and enable the implementation of a complicated real-time adaptive control scheme. Multi-resolution schemes suitable for adaptive processing include multiple resolutions in both the image and the parameter domain. In the following we show a multi-resolution coarse-to-fine parameter (operation flow) adjusting mechanism.

In any multi-resolution image generation system, a down-sampling operation must be applied to generate an image one resolution level lower than the current level. A multi-resolution image pyramid can be generated by the following iterative operations:

$$I_i = Down(I_{i-1})$$
$$I_0 = I_{input}.$$

Different schemes have been used for the '*Down*' operation. A popular method is to lowpass filter the image before down-sampling (such as Gaussian pyramid (Burt 1984)).

Normally an image is subjected to application of an operation and the operation is subjected to changes of some adjustable parameters. Suppose we repeatedly apply an operation to an image I, and during each application we change the value of an operation parameter k. After we apply the operation with all possible values of k, a family of resulting images O_k is produced, where $O_k = Operation(I, k)$ and image O_{k_i} corresponds to the operation output when k is set to k_i (see figure 5). Allowing change to other free parameters increases the dimensions of the operation output space, so the multi-resolution concept is applied to the operation parameters to constrain the data to a managable size. Let K be the finest resolution in a parameter space. A multi-resolution parameter pyramid can be generated by the following iterative operations:

$$C_i = Reduce(C_{i-1})$$

$$C_0 = K,$$

where *Reduce* is a down sampling scheme in the parameter space, possibly as simple as doubling the interval between adjacent parameter values. Let *Autoparameter(operation,I,C,k)* be an automatic parameter determination process for '*operation*' when the input data is '*I*', the parameter space is '*C*', and the initial parameter is '*k*'. A coarse-to-fine automatic parameter adjustment mechanism can be defined by the following iterative operations:

$$k_i = Autoparameter(operation, I_i, C_i, k_{i+1}),$$

where an initial parameter value k_p (best initial guess) evolves to the final parameter value k_0 (see figure 6). At each intermediate step the automatic parameter adjusting mechanism, *Autoparameter(operation,*I_i*,*D_i*,*k_{i+1}*)*, takes the initial parameter k_{i+1} from the coarser resolution and produces a new parameter value k_i which will in turn be used by a parameter adjusting mechanism, *Autoparameter(operation,*I_{i-1}, D_{i-1}, k_i*)*, operating on finer resolution data. To be more efficient, we can constrain the range of the parameter adjusting space for each resolution level to a fixed size. Thus, the coarse resolution parameter adjusting mechanism provides a coarse representation of the optimal parameter and the finer resolution parameter adjusting mechanism fine tunes (within a range) the parameter to a higher resolution. At each stage of the multi-resolution parameter adjusting mechanism, a checking mechanism can be provided. The checking mechanism determines from the performance index $f(k_i)$ from the current stage, whether the parameter adjusting process should continue to finer resolution or should exit the current level and either divert operation flow or return control to the high level vision control invoking this operation.

6. APPLICATION EXAMPLES

In this section we show adaptive image processing concepts in an automatic image segmentation system. Input to the automatic image segmentation system are laser radar 3-D range images (Hersman et. al., 1987) of man-made objects: cone, cylinder, box, basketball, roll of tape. (see Figure 7). The processing system takes the data and automatically segments the objects from the background, using morphologic processing (Serra, 1982). A fixed set of operations with certain associated intermediate image status test stages and adjustable operation parameters are used. The processing sequence, shown in Figure 8, is fixed, only the optional branches and operation parameters are variable. The concept of predefining a sequence of operations is analogous to the way image processing is set up in most of application systems. However, the adaptive system we use allows operation flow and parameters to be adjusted dynamically. We believe that by identifying the adjustable parameters and operation flow of an image processing application scheme, it is not difficult to convert it into an adaptive processing system using the predefined parameter values as the initial values for parameter adjusting mechanisms.

In the example, the operation sequence has 15 steps. Step (1) acquires the range data which is first geometrically transformed by step (2) to map the range to a plane instead of distance to a point. Noise is then removed by an iterative morphologic filtering (step (3)) operation. The table on which the objects sit is closer to radar laser source, so it shows up when a simple thresholding scheme is applied. A morphologic background normalization operation (step (4)) is applied to remove the table background and provides background status measurement which is true when the background is insignificant. A test step (step (5)) will determine whether the background normalization operation is actually needed. We mask out the background regions to preserve object shapes and retain only the focus of attention area of the image (steps (6)-(8)). Step (9) thresholds to pick seed regions of objects; Step (10) grows the seed regions by dilating the seed regions, conditioned by edge boundaries. We use a multi-resolution approach for these two steps. Thus, step (9) provides an initial seed region for the coarsest resolution and the result of step (10) in this resolution is used as the seed region for step (10) in the next higher resolution. Step (11) smooths the detected object boundaries by a morphologic closing operation and step (12) extracts corners of the segmented objects for spatial relation extraction. Step (13) extracts object features which can be used for high-level processing. Step (14) updates the mask image and step (15) checks whether more objects are left for further processing (go back to step (8)).

The operation parameters involved in this operation sequence include the number of iterations (steps (3) and (10)), structuring element sizes (steps (4), (7), (11), (12), and (14)), and thresholding values (steps (6) and (9)). In the following we describe important steps of the operation sequence. One important aspect of this processing is that in many steps, no time need be wasted converging to "perfect" adaptively selected parameters, because they only bound later process. For example, perfect edge-link is not performed because complete edges are not required when the edges serve only to "stop" a region growing operation.

Iterative morphologic filtering

The filtering is accomplished by performing successive openings and closings with square bricks of increasing size. The adjustable parameter of this operation is the number of opening-closing iterations. To dynamically determine the number of iterations, an invariant-expectation adaptive system configuration is used. The initial structuring element size is set to 3×3 and it increments by 2 for each iteration. The processing status measurement used is the root mean square difference between two successive iterations. A typical RMS difference graph of the images is shown in Figure 9. Processing should stop when the first local minimum is reached. The terminating structuring element size provides a lower bound for the foreground object size and can be used by other processing steps to constrain their adjustable parameters. A 3-D surface plot of Figure 7(e) and its filtered image is shown in Figure 10.

Morphologic background normalization

Background normalization is accomplished by a morphologic opening residue operation (Sternberg, 1986). The structuring element for the opening operation is a long horizontal line, chosen because the uninteresting background consists of a horizontal table which is longer than the objects of interest. The opening residue has the effect of eliminating the background only if the structuring element is long enough to preserve the object and eliminate the background (table), otherwise, some of the object of interest would be eliminated. To dynamically determine the proper size of the structuring element, an input sensing adaptive system configuration is used and a multi-resolution approach is applied to identify the appropriate structuring element size efficiently. The system performs an opening residue of different structuring element sizes on an image in coarse resolution and uses horizontal projection to measure the merit of the processing. Horizontal projection of a binary image I is defined by

$$P_h(r) = \sum_{\forall i, (r,i) \in I} 1.$$

In each resolution level the structuring element size is selected for which the projection values are stable and no table background (corresponding to the count at the tail part of projection index) is in the projection. Figure 11 shows an example of the coarse-to-fine structuring element size determination process applied to the image shown in Figure 7(e). In the $I_4(16 \times 16)$ image, six different sizes (14, 12, 10, 8, 6, 4) are tried and the projection counts are listed in figure 11(a). Among the six different sizes 8 is the best choice at this level. We then perform the same measurement on $I_3(32 \times 32)$ around size 16 (corresponding to size 8 in I_4). Five (20, 18, 16, 14, 12) different sizes are tried and the projection counts are listed in figure 11(b). Only 16 counts (16-32) of the projection index are used to determine the best size at this level. The best size is chosen as 14 and one more iteration is performed on $I_2(64 \times 64)$ around size 28 (corresponding to 14 in I_3). Five different sizes (32, 30, 28, 26, 24) are tried and the projection counts are listed in figure 11(c). Note that still only 16 counts (41-56) of the projection index are used to determine the best size, chosen as 26. Since the purpose of this operation is to provide a region of interest mask, it need not be very accurate. We stop the coarse-to-fine structuring element size search process at this level and select the 26×4 size as the final structuring element size for the finest level.

The parameter determination process also provides a background status measure. If the table background is insignificant in the image (no significant count at the tail portion of its projection index), the background status value will be set to true which in turn causes jumping in the operation sequence and thus ignores the focus of attention process. The background normalized image of figure 7(e) is shown in Figure 12(b).

Focus-of-attention region extraction

The operations involved are steps (6)-(8). Again, high accuracyis not required since the focus-of-attention region only serves as the basis for further segmentation processing. A simple adaptive scheme can be used to achieve a highly efficient adaptive processing operation. Here an input sensing scheme determines the thresholding value of step (6). The intensity histogram of the background normalized image is used to dynamically determine the threshold. The background normalized image typically has a bimodal histogram from which it is not difficult to determine the threshold value dynamically. Again, the input sensing scheme can be used to determine the structuring element size of step (7). The average connected component area A_{av} of step (7) input image is measured. Two predefined values s_w and s_l are used to compute the structuring element width W and length L by the following simple formula

$$W = s_w * A_{av}$$

$$L = s_l * A_{av}$$

The masking operation of step (8) needs no dynamic parameter determinations. The initial segmentation image and the focus of attention image of figure 7(e) is shown in Figure 12(c) and (d).

Object region extraction

The seed region extraction operation provides seed regions for the conditional dilation region growing operation. The threshold value may be inaccurate since the conditional dilation operation (step (10)) obtains accurate final object segmentation. A Gaussian pyramid is generated from the region-of-interest image and its edge image is down sampled as well. We dilate using a 2 by 2 kernel before down sampling the binary edge image to avoid losing edges in the coarse resolution (see Haralick et. al., 1987). An invariant-expectation scheme and a coarse-to-fine control strategy is applied to the region extraction process. The coarsest scale data is thresholded to extract seed regions by a rough threshold value determined by its histogram. A dilation of the seed region conditioned by the edge image is performed at the coarsest level.

At each level, a merit measurement $m = add/dil$ is used to determine when to stop the conditional dilation process, where dil is the number of pixels been added to the region after an iteration of dilation is performed and add is the number of pixels actually been added to the conditional dilation result. The ratio m reflects the percentage of the newly generated data been used, and we stop the conditional dilation operation if the ratio is small to avoid leaking from the region through gaps in the edge image. The conditional dilated image is expanded and serves as seed region for the next finer level conditional dilation process. The process continues until the stop condition occurs in the finest pyramid level.

To smooth the boundary of the segmented image, a morphologic closing operation is applied. an input sensing scheme determines the structuring element size dynamically by performing a distance transform of the region and extracting the histogram of the distance image. The largest histogram value reflects the size of the maximum enclosed square, and is used to determine the structuring element size for closing operation. This value can also be used to derive the size for the opening residue corner detection operation. The segmentation results of Figure 7 are shown in Figure 13.

7. CONCLUSIONS AND FUTURE WORK

Adaptive image processing schemes have been described and tested. A multi-resolution approach has been introduced which allows operations in different image and parameter scales. The experimental results show that the adaptive scheme is able to provide dynamic compenstation for input variations.

Future work includes more experiments to better understand how to map conventional predefined image processing processes into adaptive configurations, how to select the most appropiate status measurements, and how to reach a balance between adaptive image processing and high-level vision processing.

8. REFERENCES

1 Brooks, R. A., 'Symbolic Reasoning Among 3-D Models and 2-D Images', Artificial Intelligence. Vol. 17, Nos. 1-3, Aug. 1981, pp. 159-170.

2 Brooks, R. A., 'Model-Based Computer Vision', UMI Research Press, 1981.

3 Burt, P. J., 'The Pyramid as a Structure for Efficient Computation', Multiresolution Image Processing and Analysis, Rosenfeld, eds., Springer-Verlag, 1984, pp. 6-35.

4 Davis, L. S. and S.S.V. Hwang,'The SIGMA Image Understanding System', Proceeding of the third Workshop on Computer Vision: Representation and Control, 1985.

5 Eveleigh, V. W., 'Adaptive Control and Optimization Techniques', McGraw-Hill, 1967.

6 Hanson, A. R. and E. M. Riseman, 'VISIONS: A Computer System for Interpreting Scenes', in Computer Vision Systems, Hanson & Riseman, eds., Academic Press, N. Y., 1978, pp. 303-333.

7 Haralick, R. M., C. Lin, J.S.J. Lee, and X. Zhuang,' Multiresolution Morphology', ICCV'87, June, 1987, London, pp. 516-521.

8 Hersman, M., F. Goodwin, and A. Slotwinski, 'Coherent Laser Radar Application to 3-D Vision and Metrology', Proceeding of the Vision'87, June, 1987, pp. 3.1-3.12.

9 Kohl, C. A., A. R. Hanson, and E. M. Riseman, 'Goal-Directed Control of Low-level Processes for Image Interpretation', Proceedings Image Understanding Workshop, Los Angeles, Feb. 1987, pp.538-551.

10 Lee, J.S.J., 'Real-Time Machine Vision Algorithms', Technical Report, Machine Vision International, August, 1986.

11 McKeown, D. M., Jr., W. A. Harvey, and J. McDermott, 'Rule Based Interpretation of Aerial Imagery', Proceeding of the IEEE Workshop on Principles of Knowledge-Based Systems, Dec. 1984, pp. 145-157.

12 Ng, K. C., 'Adaptive Control Systems (hill climbing)', Modern Control Theory and Computing, Bell and Griffin eds., 1969.

13 Riseman E. M. and A. R. Hanson, 'A Methodology for the Development of General Knowledge-Based Vision Systems', Proceedings of the IEEE Workshop on Principles of Knowledge-Based Systems, Dec. 1984, pp. 159-170.

14 Serra J., 'Image Analysis and Mathematical Morphology', London: Academic, 1982.

15 Sternberg S. R., 'Grayscale Morphology', Special Section on Mathematical Morphology, Computer Vision, Graphics, and Image Processing, Sep. 1986.

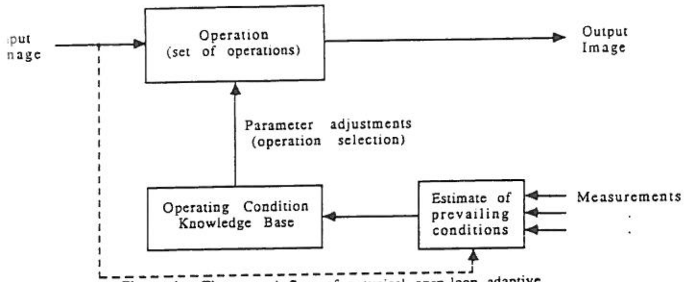


Figure 1. The control flow of a typical open-loop adaptive image processing controller.

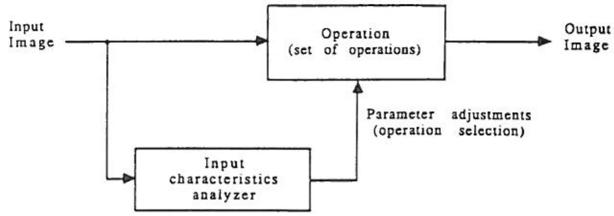


Figure 2. A general schematic of the input sensing adaptive image processing system.

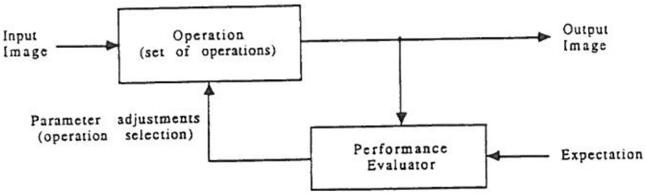


Figure 3. The control flow of a typical invariant-expectation adaptive system.

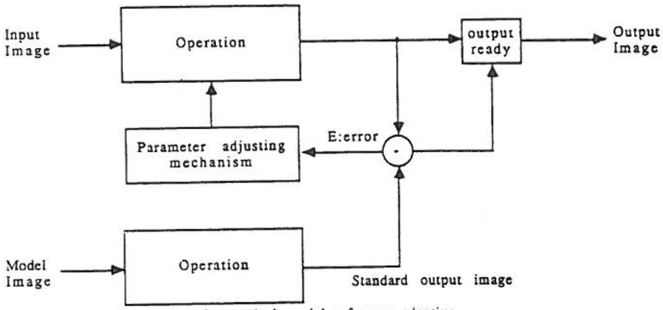


Figure 4. The schematic of a typical model reference adaptive image processing system.

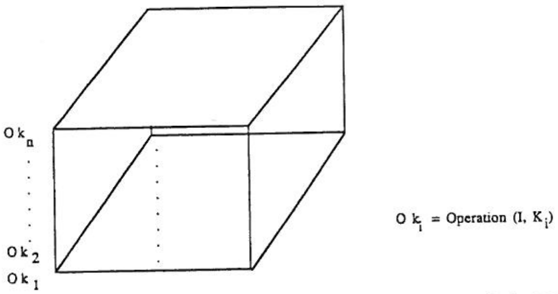


Figure5. The family of output images when the input parameter k is changed.

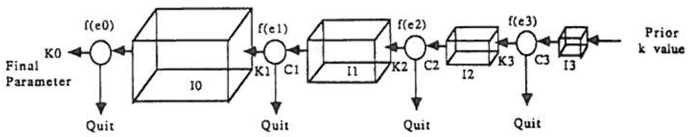


Figure6. Coarse-to-fine multi-resolution parameter (operation flow) adjustment mechanism.

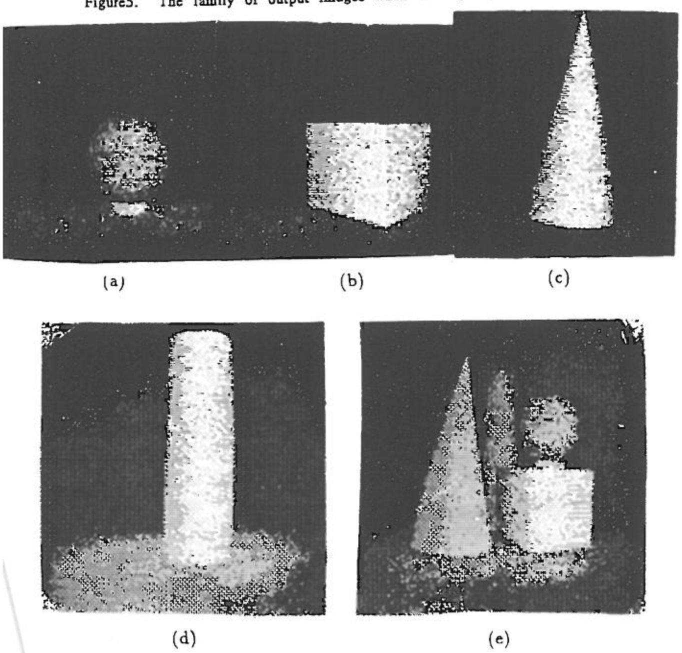


Figure 7. The laser radar 3-D range images of man-made objects.

i = 0

(1) I_{in}= Laser radar(objects)

(2) I_1 = Geometric transformation (I_{in})

(3) I_2 = Iterative morphologic filtering(I_1;# iterations)

(4) I_n+background status

= Morphologic background normalization (I_2;size)

(5) If (background status) $I_{foa} = I_2$; goto (9)

(6) I_4 = Threshold (I_n; value)

(7) I_m = Dilate(I_4;size)

(8) I_{foa} = Mask-out(I_2, I_m)

(9) I_5 = Threshold (I_{foa}; value)

(10) I_6 = Conditional dilation (I_5, I_{edge};# iterations)

(11) I_{obj} = Close(I_6;size)

(12) $I_{corner} = I_{obj}$ - Open(I_{obj};size)

(13) Object Feature(i+k) = Feature extraction(I_{obj}, I_{corner})

k = 1,...,# of object detected

(14) $I_m = I_m$ - Dilate(I_{obj};size)

(15) IF (object status(I_m)) i=i+# of object detected; goto (8)

Figure 8. Image segmentation operation sequence.

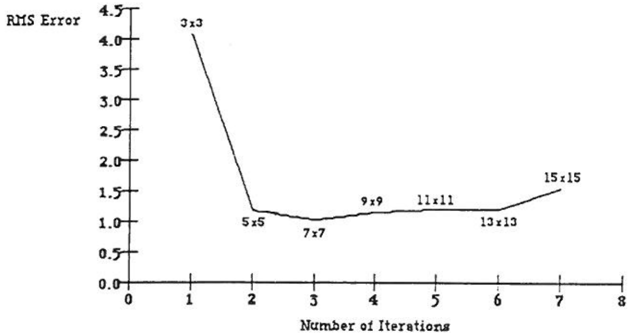
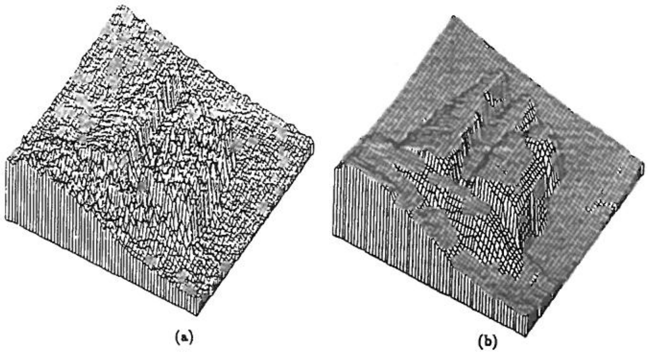


Figure 9. Processing Status Measurement for Filtering.



(a) (b)

Figure 10. (a) the 3-D surface plot of figure 7(e) and (b) the 3-D surface plot of (a) after iterative morphologic filtering is applied.

Structuring Element Size

Projection Index	14	12	10	8	6	4
0	160	166	179	187	194	207
1	4	4	3	2	1	0
2	4	3	3	2	1	0
3	3	2	2	2	1	0
4	4	4	4	4	3	1
5	4	4	4	4	4	2
6	7	7	7	8	5	4
7	6	8	6	4	4	4
8	6	6	6	6	6	4
9	9	9	9	6	6	6
10	10	10	8	8	8	6
11	10	10	8	8	8	8
12	8	8	8	8	8	8
13	7	7	7	7	5	4
14	2	2	2	2	2	2
15	6	3	0	0	0	0
16	6	5	0	0	0	0

16X16

(a)

Projection Index	20	18	16	14	12
16	11	11	11	11	11
17	17	15	15	15	15
18	19	16	16	16	16
19	16	15	15	15	15
20	16	16	16	16	16
21	17	16	16	16	16
22	18	16	16	16	16
23	18	18	18	18	18
24	18	18	18	18	17
25	13	13	13	13	9
26	5	2	2	0	0
27	0	0	0	0	0
28	0	0	0	0	0
29	0	0	0	0	0
30	0	0	0	0	0
31	2	0	0	0	0
32	6	0	0	0	0

32X32

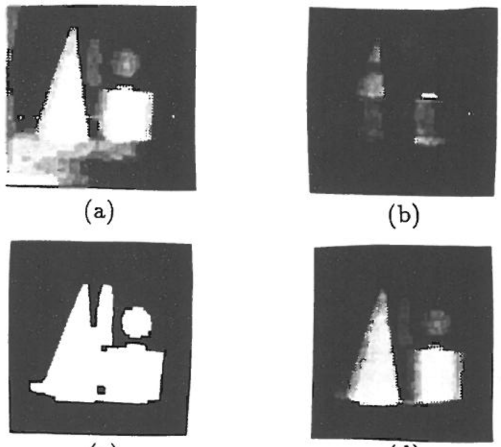
(b)

Projection Index	32	30	28	26	24
41	34	34	34	34	34
42	34	34	34	34	34
43	35	35	35	35	35
44	35	35	35	35	35
45	35	35	35	35	35
46	35	35	35	35	35
47	35	35	35	35	35
48	34	34	34	34	31
49	24	24	22	19	18
50	10	10	7	3	0
51	0	0	0	0	0
52	5	0	0	0	0
53	3	3	0	0	0
54	0	0	0	0	0
55	0	0	0	0	0
56	0	0	0	0	0

64X64

(c)

Figure 11. The coarse-to-fine structuring element size determina tion process.



(a) (b) (c) (d)

Figure 12. (a) Noise removed image of figure 7(e); (b) background normalized image of (a); (c) initial segmentation result of (b); (d) the focus-of-attention image.

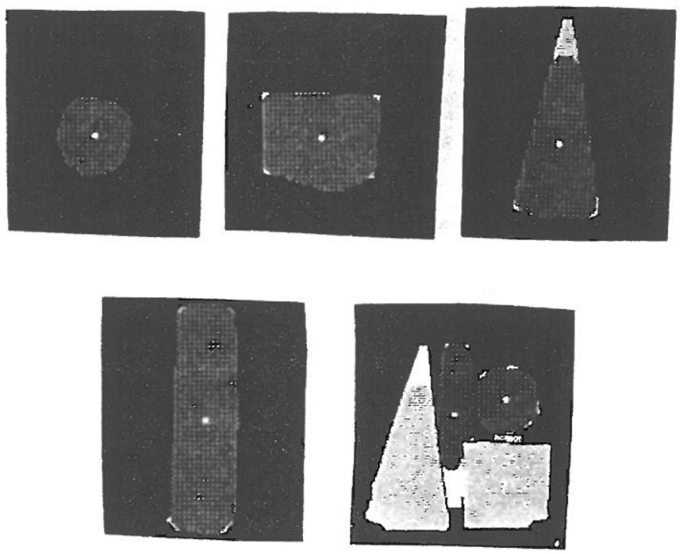


Figure 13. The segmentation results of figure 7.