

MULTIRESOLUTION MODELING AND MATCHING USING SALIENT FEATURES

Z.N. Li

*School of Computing Science
Simon Fraser University
Burnaby, B.C. V5A 1S6*

Q. Song

*Department of Electrical Engineering and Computer Science
University of Wisconsin-Milwaukee
Milwaukee, WI 53201*

ABSTRACT

Automatic recognition of industrial parts is vital to industrial automation and robotics. In this paper, we describe a program that recognizes 2D objects using salient features. The objects used to illustrate our results are ordinary room keys from a set of test images. Multiresolution images in a pyramid-like structure are used. At a coarse level the (standard) Hough transform is applied for hypothesizing orientations and locations of the objects. At lower levels in the pyramid the generalized Hough transform is applied for the matching of finer salient features. Methods for hierarchical modeling and hierarchical matching are presented. Control strategies in this hierarchical environment are also discussed.

KEYWORDS Hierarchical modeling, Hough transforms, Multiresolution, Object recognition, Pyramid, Salient features

1. Introduction

Automatic recognition of industrial parts is vital to industrial automation and robotics. In this paper, we describe a program that recognizes 2D objects using salient features. The objects used to illustrate our results are ordinary room keys from a set of test images. Multiresolution images in a pyramid-like structure are used. At a coarse level the (standard) Hough transform is applied for hypothesizing orientations and locations of the objects. At lower levels in the pyramid the generalized Hough transform is applied for the matching of finer salient features. Methods for hierarchical modeling and hierarchical matching are presented. Control strategies in this hierarchical environment are also discussed.

Model-based vision systems have been extensively studied by many researchers [1]. Among the many systems developed, Perkins [2] reported a model-based vision system for recognizing industrial parts. "Concurves" of parts are formed from "chain" data and then matched to models. Brooks [3] described model-based 3D interpretations of 2D images in the ACRONYM system. The system can recognize simple 3D objects such as electrical motors.

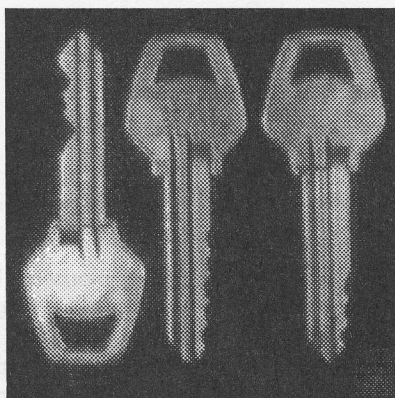
Hierarchical object recognition was pioneered by Kelly in his 1971 work of recognizing human faces [4]. Edges of faces are first located in a reduced resolution image. The edges found are then used as a "plan" for processing the original picture. In 1975, Tanimoto and Pavlidis [5] proposed a pyramid structure that supports multiresolution and hierarchical representation of image data. The basic pyramid structure has n levels, each level k ($0 \leq k < n$) has $2^k \times 2^k$ nodes. Each node at level k is linked to its 13 neighbors, i.e., 1 parent, 8 siblings and 4 children. The connectivity between layers of nodes in the pyramid structure makes the hierarchical data flow (bottom-up and/or top-down) very efficient.

Recently, Neveu et al. [6] described a program for 2D object (room keys) recognition using multiresolution models. Objects are not represented by connected "chains" of points such as ribbons, skeletons, or concurves. Instead, simple sets of unconnected edge points are adopted in modeling. The program uses generalized Hough transform for all its matching processes. "The coarsest scale boundary description of the object is stored as the root node of the model graph and is matched first with the corresponding resolution level of an input image. The result of this match are used to hypothesize a list of approximate positions and orientations of the object." The program then uses higher resolution models to match finer boundary features. It decides the detection of objects according to the accumulated weights. The hierarchical modeling and the coarse-to-fine matching have been shown to speed up the recognition of room keys significantly. One of the advantages of using the generalized Hough transform is that the matching process is less sensitive to noisy and broken contours. Neveu's program follows a simple top-down search pattern. Its success relies heavily on the high level hypotheses. Since the top level image resolution is very low (16×16 in [6]), the generalized hough transform may not be able to yield good hypotheses. Moreover, a simple breadth-first search strategy is used to traverse the model graph composed of many features of interest for the object. It seems that an application of salient features would considerably reduce the search space for the domain of their objects (keys, bolts, washers).

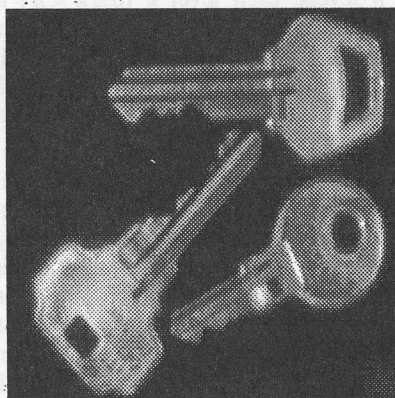
We have been studying object recognition in the pyramid environment using key (salient) features [7,8]. The application of salient features appears to be even more powerful for the recognition of 2D industrial parts which often

have more regular and rigid shapes. In this paper we illustrate how the salient features can be incorporated into multiresolution modeling and matching for the recognition of 2D objects. Section 2 describes the methodology. Section 3 presents our experimental results. Section 4 consists of conclusions and some discussions on failure recovery and control issues.

2. Hierarchical Object Recognition Using Salient Features



(a) Test Image 1



(b) Test Image 2

Fig. 1 Key images

Our test data are images of ordinary keys (Fig. 1). For ease of identification, the keys in the Test Image 1 (Fig. 1(a)) are named K1, K2, and K3 (from left to right). The small key in the Test Image 2 (Fig. 1(b)) with a round head is named K4, and the larger key with a nearly rectangular head is named K5. Apparently, K1 also appears in the Test Image 2 with a different orientation.

2.1. Salient Features

Salient features will be used for modeling and matching. For this set of keys the salient features are rather obvious. The key K4 has its unique round head and smaller size, whereas K5 has its own head shape and larger size. If the task is to distinguish K4 or K5 from other keys, then the detection of these salient features would be sufficient. The shape of the key holes (circular for K4, square for K5) could probably add some degree of certainty to the recognition program. The shape

information for the head and hole is salient for identifying K1, K2 and K3 as a group. However, it is not sufficient for separating them. To discriminate K1 from K2 and K3, one of the salient features would be the large first tooth (the one closest to the tip). Finally, for K2 and K3 which have exactly the identical first three teeth, the only salient feature will be the existence of the fourth and fifth teeth in K4.

The concept of salient features has been explored by many other researchers. For example, Bolles [9] successfully applied a local-feature-focus method for recognizing partial visible objects. The advantage of employing the salient features is that the program can focus its attention in finding the most important features for recognition.

2.2. Hierarchical Modeling Using Salient Features

In many cases the salient features have different sizes. In our examples the key heads are much larger than the key holes and the key teeth. It is often better to use multiresolution image representations in which large and extended features can be more simply detected as local features at a coarser resolution. Since multiresolution images can readily be stored in the pyramid, it is very efficient to create hierarchical models and match them at different levels in the pyramid.

As in [6] edge points (not necessarily connected) are used to construct models. Figure 2 illustrates our multi-level hierarchical modeling for the keys using salient features. The graph is much smaller than the model graph presented by Neveu et al. [6] for a similar room key. At the top level (level 5 in the pyramid) in this hierarchical model graph, entire edge maps for complete keys at a resolution of 32×32 are displayed. As one might expect, many details (e.g., teeth, holes of the keys) are lost. Only global features are preserved. One of the salient global features is the long straight line on one side of the key shafts. For other levels (level 6 and 7 in the pyramid) the complete keys are no longer displayed. Only the salient features that are most important for the recognition are used as elements of the hierarchical models. At level 6 in the pyramid, the edge points on the contours of the key heads are adopted as models. At the next level (level 7 in the pyramid), the edge points for the key holes and key teeth are used. For example, the large first tooth is shown for K1, the first three teeth and the shape of the fourth and fifth teeth are shown for K2 and K3. The choice of the best resolution levels for the salient features depends on individual applications. Some of the factors to consider are the object size and the resolution of input images.

The generalized Hough transform is employed for the most of the recognition process. R-tables are constructed for salient features at different resolutions. At a chosen resolution a set of edge points on the contour of a salient feature from a modeling key is used to construct a R-table. Their gradient angles are used as indices to the R-table. A reference point (x_c, y_c) is chosen near the center of these edge points. For each edge point a vector r is computed from the edge point to the reference point. Instead of α, r suggested by Ballard [10], $(x_i - x_c, y_i - y_c)$ is used [6] to represent the vector r . This seems to simplify the calculation. The R-table contains one such entry for each edge point (x_i, y_i) of the model.

2.3. Hierarchical Matching Using Salient Features

The hierarchical matching process is illustrated by the flow

chart in Fig. 3.

As we pointed out in Section 1, many traditional top-down, coarse-to-fine search programs rely heavily on the generation of a rough outline from low resolution images [4]. In Neveu's paper [6] the generalized Hough transform is applied to the top level zero-crossing image with the coarsest resolution 16×16 . At this level (level 4) the transform is attempted on 16 possible directions to give an estimated object orientation in increments of 22.5° . The process will then go down to the next level (level 5) with a higher resolution to search for finer features. Taking into account the possible error introduced by the loss of information at the low resolution images, transforms are attempted on 11 possible directions at multiples of 4.5° centered around its parent's estimated orientation angle at level 5. Since the generalized Hough transform takes time in proportion to the number of possible orientation angles trialed, the computation could be quite expensive. Moreover, the very low resolution images, e.g., 16×16 , usually do not preserve the needed shape information. It is likely that the initial estimate will not even guarantee the error to be less than 22.5° . In that case, the success of the search would be very difficult.

It is often feasible to find some salient features that can be preserved well in low resolution images and be used to

generate a good hypothesis regarding to the location and orientation of the object. In our key images the long straight contour lines at one side of the key shafts have such nice properties. The extraction of such features will help generating rather accurate estimates on the object orientation. Relatively few possible orientations need to be tested on for the matching. Hence, the search space at the next lower level becomes considerably smaller.

For extracting the long straight lines, the (standard) Hough transform [11] can be employed. Lines are represented by $x \cos \theta + y \sin \theta = \rho$, where ρ is the normal distance from the line to the origin, and θ is the gradient direction of the edge points on the line. Only a single two-dimensional $(\theta - \rho)$ accumulator array is needed. Because the gradient angle is used, only one element of the accumulator array is incremented for every edge point in the edge map. The computational cost is very low. It is proportional to the number of edge points in the edge map.

After generating the initial hypotheses for the possible locations and orientations at the coarse level, the multi-level models (R-tables) can be used for matching other salient features at lower levels. Our program implemented so far follows a simple top-down (coarse-to-fine) control pattern. The generalized Hough transforms are employed for matching

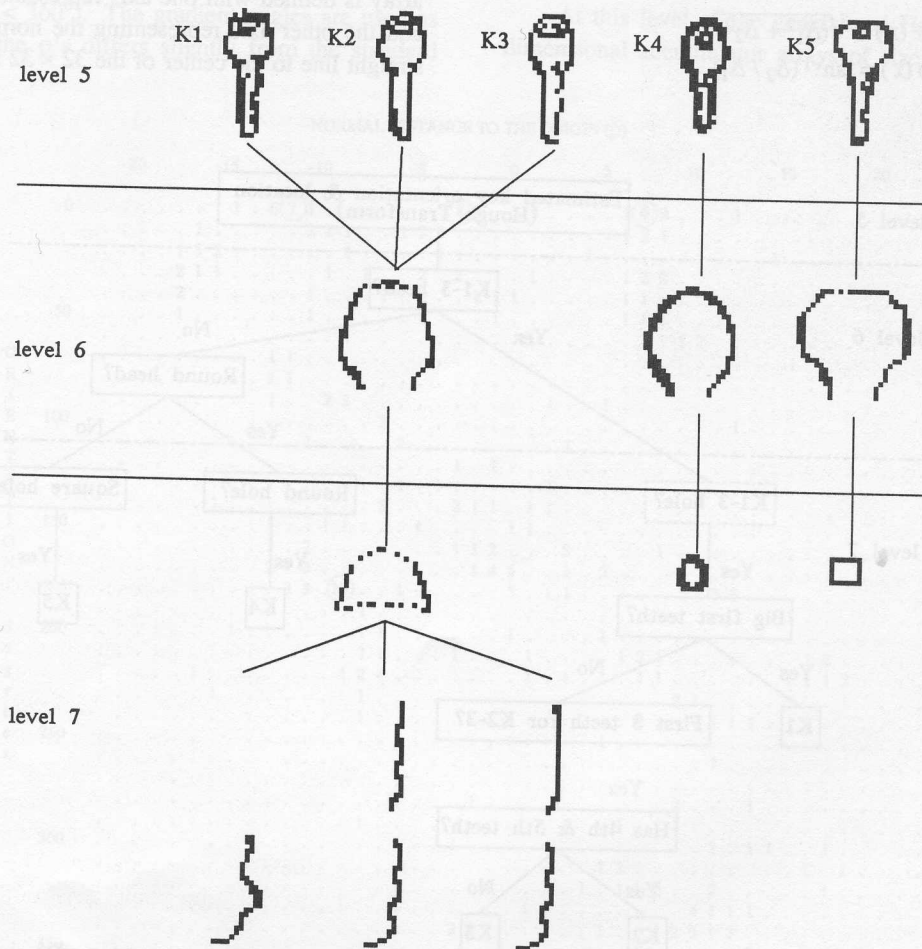


Fig. 2 Multi-level Hierarchical Modeling Using Salient Features

the finer features. The method for the hierarchical matching is similar to Neveu's program except (1) fewer angles will be examined, because better estimates on the key orientations can be applied; (2) smaller space will be searched, because attention will be focused on the salient features. The matching process for the key images will be described in detail in the next section.

3. Experimental Results

In this section we will illustrate how our program recognizes K1, K2 and K3 from the Test Images 1 and 2.

3.1. Multi-level Edge Detection

The pyramidal median filtering [7] is applied to derive multiresolution images in the pyramid. This technique is good for reducing noise while still preserving edges in the original image. An input key image (256×256) is stored at level 8. At first, each node on level 7 takes the median intensity value of its 3×3 child set as its intensity value. The result is a filtered image at level 7 with a resolution of 128×128 . The same process will then be repeated at levels 6, 5, and 4 to generate a sequence of multiresolution images.

After that, an edge operator is applied to obtain the gradient information at each level. For an image function $f(x)$, the gradient magnitude $s(x)$ and direction $\phi(x)$ can be computed as

$$s(x) = \sqrt{\Delta_1^2 + \Delta_2^2}$$

$$\phi(x) = \tan^{-1}(\Delta_2 / \Delta_1)$$

where

$$\Delta_1 = f(x+1, y) - f(x, y)$$

$$\Delta_2 = f(x, y+1) - f(x, y)$$

Our program computes $s(x)$ first. Direction $\phi(x)$'s are computed only for those edge points whose $s(x)$ exceed a certain threshold. These edge points form an edge map at this level. Also, in our experiment slightly different Δ_1 and Δ_2 are used, i.e.,

$$\Delta_1' = f(x+1, y) - f(x-1, y)$$

$$\Delta_2' = f(x, y+1) - f(x, y-1)$$

It is found that the gradient directions thus extracted are more accurate.

Fig. 4 depicts the multiresolution edge maps at levels 4, 5, 6 and 7 in the pyramid for the Test Image 1. The edge maps with different resolutions are displayed at a common size for comparison. As one might anticipate, very little shape information is preserved in the level 4 (16×16) edge map.

3.2. The (Standard) Hough Transform for Detecting Key Orientations

The long straight side of the key shafts is the salient feature that indicates the locations and orientations of the keys. The (standard) Hough transform is employed to find the long straight lines. The edge points from the low resolution image at level 5 (32×32) are used. A two-dimensional accumulator array is defined with one axis representing θ from 0° to 360° , and the other axis representing the normal distance ρ from a straight line to the center of the 32×32 image. For each edge

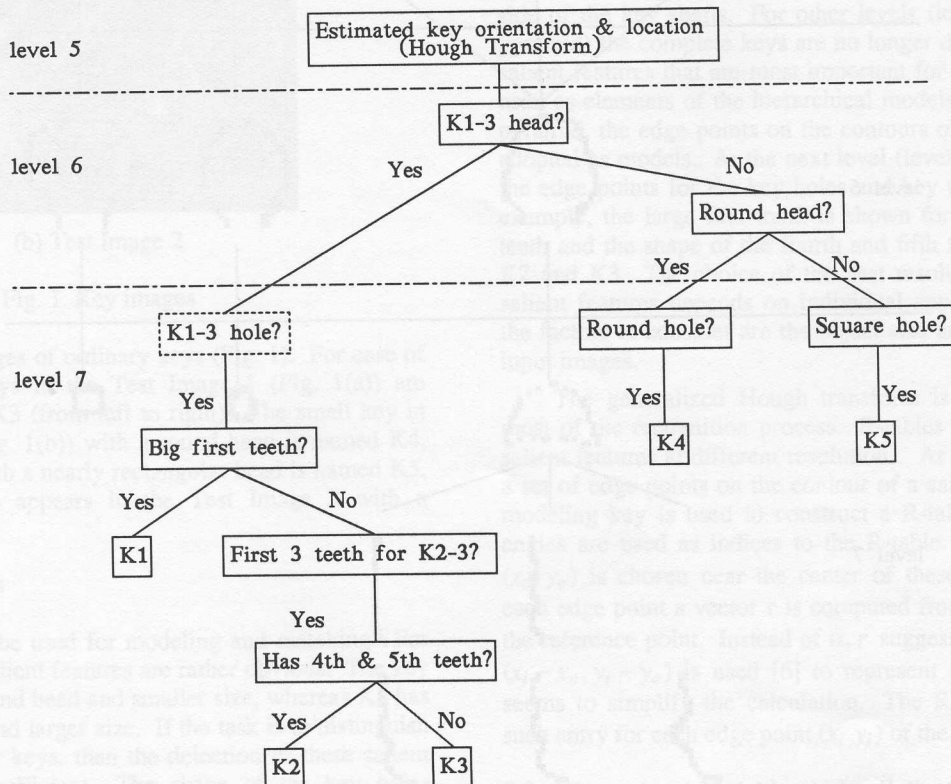


Fig. 3 Hierarchical Matching Using Salient Features

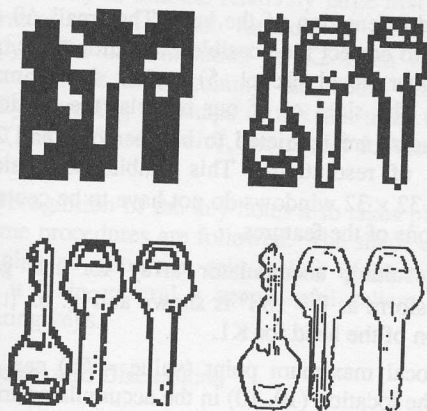


Fig. 4 Four Level Edge Maps in the Pyramid

point (x_i, y_i) with its gradient direction θ , a straight line is hypothesized, i.e.,

$$x_i \cos \theta + y_i \sin \theta = \rho$$

Consequently, the corresponding point (θ, ρ) in the accumulator array is incremented.

Fig. 5 is the resulting accumulator array for the straight line detection at level 5. The image used is the Test Image 1. For clarity of the display, all zero values have been shown as dots. The horizontal axis represents ρ and the vertical axis represents θ ($0^\circ \leq \theta < 360^\circ$). The gradient angles are used as θ 's. The range of the ρ 's differs slightly from the standard

$(\theta-\rho)$ space in the way that we do have both positive and negative ρ values even though θ ranges from 0° to 360° .

Simple thresholding method is employed to locate the local maximum points in the accumulator array. There exist at least three true local maximum values in this accumulator array indicating the detection of the straight sides of the key shafts at $(\theta=180^\circ, \rho=-9, \text{value}=10)$ for K1, $(\theta=0^\circ, \rho=-3, \text{value}=8)$ for K2, and $(\theta=0^\circ, \rho=7, \text{value}=9)$ for K3. The program might also take other "false" local maximum values, e.g., at approximately $(\theta=0^\circ, \rho=-12, \text{value}=7)$ which is caused by the teeth side of the key shaft for K1. This hypothesis will later be shown to be false.

3.3. Generalized Hough Transforms for Matching Finer Features

The detection of the key shaft sides enables the program to generate hypotheses for the possible locations and orientations of the keys. The generalized Hough transform is first applied at the next lower level (level 6) to recognize the salient features of the keys, for example, the shape of the heads. As an example, the matching of the model (R-table) for the head of K1-3 is shown below. Windows of 32×32 are placed at the hypothesized locations in the edge map. Table 1 shows the generated hypotheses for locations and orientations of possible key heads in the Test Image 1.

At this level of the generalized Hough transform, three-dimensional accumulator arrays of size $32 \times 32 \times 3$ are used,

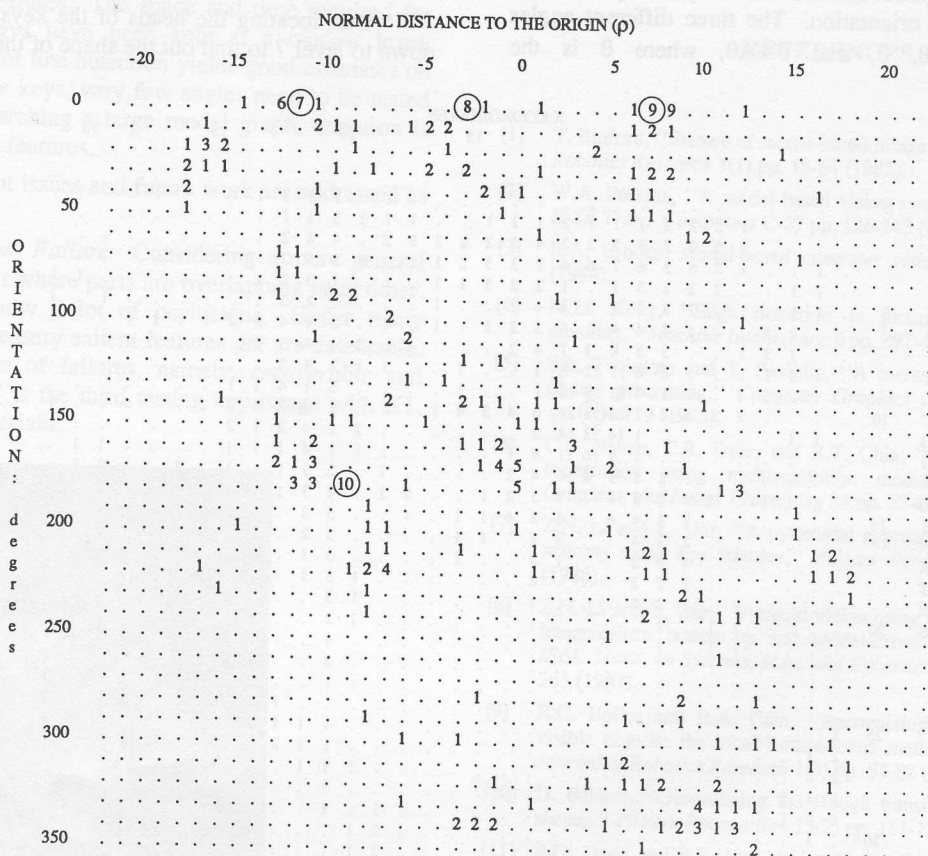


Fig. 5 An accumulator array for the Straight Line Detection at Level 5

Table 1
Hypotheses for Locations and Orientations of Key Heads

	Hyp. 1	Hyp. 2	Hyp. 3	Hyp. 4
Strength of the Hypothesized Key Side	10	8	9	7
Orientation (θ)	180°	0°	0°	0°
Location of the Key Side at Level 5	(7, 7) to (7, 16)	(13, 15) to (13, 24)	(23, 16) to (23, 24)	(3, 5) to (3, 16)
Location of the Key Side at Level 6	(14, 14) to (14, 32)	(26, 30) to (26, 48)	(46, 32) to (46, 48)	(6, 10) to (6, 32)
Center Position for the Key Head at Level 6	(12, 45)	(28, 16)	(48, 16)	(8, -2)*
Upper Left Corner of 32x32 Window at Level 6	(2, 34)	(18, 4)	(38, 4)	(0, 0)*

* At level 6 the (erroneously) hypothesized key head was centered at (8, -2) which is beyond the boundary of the image. Nevertheless, there is not even any trace of a portion of the key head around this location in the image. Hence, no significant maximum value was detected in the accumulators. The hypothesis (Hyp. 4) was refuted.

representing two degrees of freedom in position and one degree of freedom in orientation. The three different angles examined are $\theta - \Delta\theta$, θ , and $\theta + \Delta\theta$, where θ is the

hypothesized orientation of the key. The small $\Delta\theta$ (currently 10°) is used to correct the possible estimation error introduced by the higher level (level 5) nodes in estimating the orientation. The sizes of our models (the portions of the salient features) are restricted to be no more than 24x24 at their levels of resolutions. This enables an easier match, because the 32x32 windows do not have to be centered at the exact locations of the features.

The resulting accumulator array for the generalized Hough transform as $\theta = 180^\circ$ is shown as Fig. 6. It illustrates the detection of the head for K1.

One local maximum point (value = 58) can clearly be located at the location (10, 10) in the accumulator array. Since the position of the window is known, the coordinates of the maximum point in the accumulator array can simply be converted into coordinates in the original image at this level. For this example, the position for the upper left corner of the window is (2, 34). Hence, the head for K1 is actually found at the location (12, 44) at the level 6 image, which is very close to the hypothesized location (12, 45). The accumulator array for $\theta - \Delta\theta$ (170°), and $\theta + \Delta\theta$ (190°) are also examined. The maximum values are 18 and 16 respectively, which are significantly less than the value 58 obtained as $\theta = 180^\circ$. Therefore, the program is very confident of the detection of the key head at the orientation of 180°.

Additional two local maxima are found in the accumulator arrays for $\theta = 0^\circ$ representing the heads of K2 and K3 in the Test Image 1. The result for the Test Image 2 has only one maximum point in the accumulator array as $\theta = 270^\circ$ indicating that the key head for K1 is recognized.

After locating the heads of the keys, the process will go down to level 7 to find out the shape of the teeth. The program

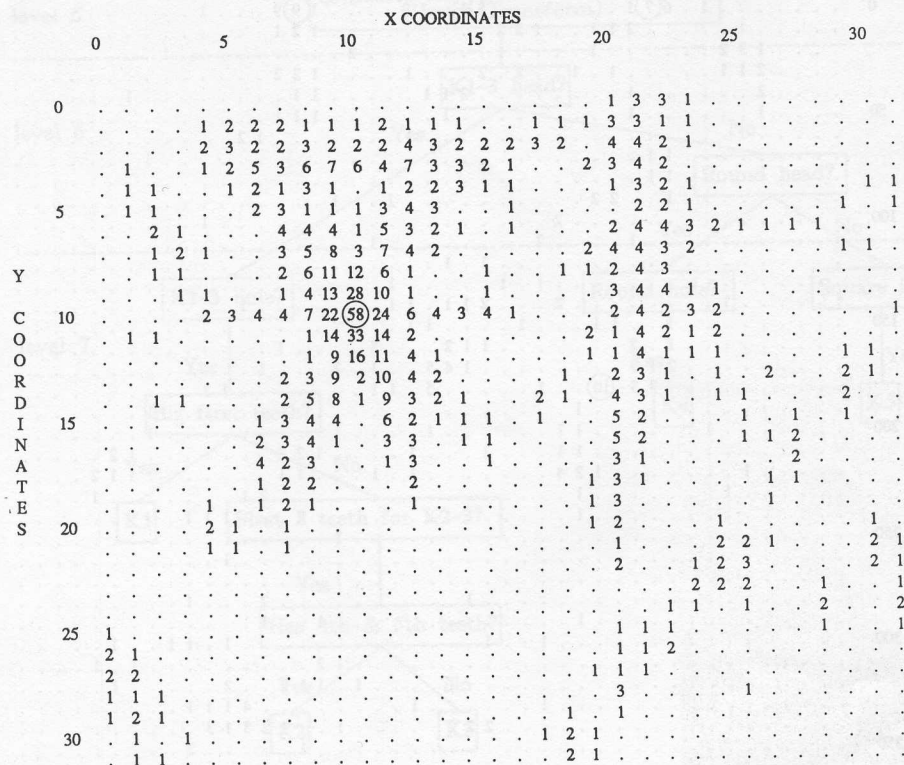


Fig. 6 An Accumulator Array ($\theta = 180^\circ$) for the Detection of a Key Head at Level 6

has little difficulty to find the relatively large first tooth of K1 in Test Images 1 and 2 with rotations of 180° and 270° accordingly. The discrimination of K2 from K3 takes the program two steps in examining the teeth. Namely, after successfully matching the shape of the first three teeth at (58, 103) and (104, 103) at level 7, the model for the fourth and fifth teeth is matched.

The recognition of the key holes also takes place at level 7. The same procedures are followed. The shape of key holes does not play an important role in this illustrated example. However, it is in general a very useful salient feature for discriminating keys.

4. Conclusion and Discussions

This paper described a program that recognizes 2D objects using salient features. The test data are the images of room keys. It has been shown that the multiresolution image representation in the pyramid structure facilitates the hierarchical modeling and matching. The recognition process starts at a relatively high level (with a resolution of 32×32) in the pyramid. At that level, straight lines are chosen as the salient features to look for, because (1) they imply strongly the locations and orientations of the keys, (2) they can be preserved better than other features at relatively low resolutions, and (3) they can be efficiently detected by the (standard) Hough transform. The current implementation of the program follows a simple top-down, coarse-to-fine search pattern. Once a hypothesis for the location and orientation of a key is generated, the search will be directed to those salient features that can be best detected at certain resolutions. The generalized Hough transform is used for modeling and matching the finer features. The space and time required for the accumulator arrays have been kept at a modest level, because (1) the straight line detection yields good estimates on the orientations of the keys, very few angles need to be tested on; (2) instead of searching a large model graph, attention is focused on the salient features.

Several important issues and future work are addressed as below:

- *Recovering from Failure.* Considering a more general industrial environment where parts are overlapping each other. A 2D image will show a lot of occlusions. Under many circumstances the necessary salient features are not detectable. There exist two types of failures, namely, *recoverable* and *unrecoverable*. Fig. 7 is the third testing key image with K1, K2 and K4 slightly overlaid.

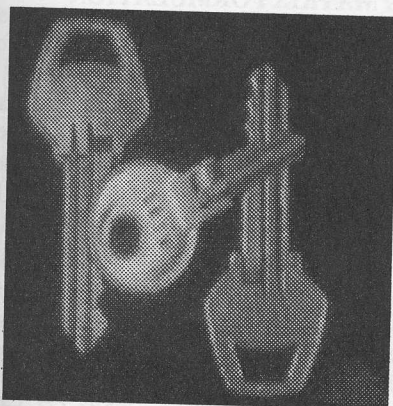


Fig. 7 Test Image 3

Our program can easily recognize K1, because all the salient features in the model graph are still present. On the other hand, it will never be able to surely recognize K2, because the 4th and 5th teeth are completely occluded by K4. This is a unrecoverable failure. (As a matter of fact, humans can not identify K2 from this 2D image either.) However, if a portion of the 4th and 5th teeth was visible, this immediately became a more interesting recoverable situation. Some common methods for recovering from a failure are: (a) *probabilistic reasoning*, i.e., assign weights to your results; (b) *secondary search*, i.e., search for other features (salient or non-salient) when your first attempt fails. The program should be enhanced to handle recoverable failures. The adoption of salient features requires its own recovery mechanism. It does not appear that the use of the salient features would necessarily hinder the analysis of complex images.

- *More Sophisticated Control Strategies.* For a pure top-down search program, its success depends on the reliable information at the top level. Moreover, a rigid serial control between levels is enforced. It is desirable to develop more powerful search patterns for this type of hierarchical object recognition. The pyramid structure is deemed to be suitable for building hardware multicomputers. The control issue is especially important if we want to bring in more parallelism for this type of multiresolution modeling and matching in a multiprocessor environment. A potential improvement is to emphasize the bottom-up search and *lateral search* [8]. At least, the multiprocessors at lower levels could start to look for the appropriate salient features at their levels when the top level hypotheses failed. Furthermore, they could communicate with other processors in various directions, upward, downward, and laterally.

REFERENCES

- [1] T. Binford, "Survey of model-based image analysis systems," *Int. J. Robotics Research* 1(1) pp. 18-64 (1982).
- [2] W.A. Perkins, "A model-based vision system for industrial parts," *IEEE Trans. Computers* C-27 pp. 126-143 (1978).
- [3] R.A. Brooks, *Model-based computer vision*, UMI Research Press (1984).
- [4] M.D. Kelly, "Edge detection in pictures by computer using planning," *Machine Intelligence* 6 pp. 397-409 (1971).
- [5] S.L. Tanimoto and T. Pavlidis, "A hierarchical data structure for picture processing," *Computer Graphics and Image Processing* 4 pp. 104-119 (1975).
- [6] C.F. Neveu, C.R. Dyer, and R.T. Chin, "Two-dimensional object recognition using multiresolution models," *Computer Vision, Graphics, and Image Processing* 34 pp. 52-65 (1986).
- [7] Z.N. Li and L. Uhr, "A pyramidal approach for the recognition of neurons using key features," *Pattern Recognition* 19(1) pp. 55-62 (1986).
- [8] Z.N. Li and L. Uhr, "Pyramid vision using key features to integrate image-driven bottom-up and model-driven top-down processes," *IEEE Trans. on Systems, Man, and Cybernetics* SMC-17(2) pp. 250-263 (1987).
- [9] R.C. Bolles and R.A. Cain, "Recognizing and locating partially visible objects: the local-feature-focus method," *The International Journal of Robotics Research* 1(3) pp. 57-82 (1982).
- [10] D. Ballard, "Generalizing the Hough transform to detect arbitrary shapes," *Pattern Recognition* 13(2) pp. 111-122 (1981).
- [11] R.O. Duda and P.E. Hart, "Use of the Hough transform to detect lines and curves in pictures," *Communications of the ACM* 15 pp. 11-15 (1972).