

A KNOWLEDGE-BASED EDGE / OBJECT MEASUREMENT TECHNIQUE

Sabry F. El-Hakim

Institute for Information Technology
National Research Council of Canada
Ottawa, Ontario, Canada

David B. Westmore

Info Magnetics Technologies Inc.
Winnipeg, Manitoba, Canada

ISPRS Commission V

ABSTRACT

Detecting edges using generic edge detection techniques is often unsuccessful in detecting all the edges of interest under practical lighting conditions. Even if the edges are detected successfully, the problem of determining the 3-d coordinates remains difficult without a priori knowledge of the object. A procedure for edge extraction under less than ideal lighting conditions and feature-based matching technique using at least four cameras are described. A method for creating a wire-frame model of the object using either CAD data or an interactive teaching procedure, and the run-time-mode operations in applications such as inspection and reverse engineering are also presented.

KEY WORDS: Machine Vision, Knowledge Base, Feature Extraction, CAD, Image Matching.

1. INTRODUCTION

One of the challenges that had to be faced, in order to facilitate automated inspection of a variety of manufactured parts, was the location and recognition of the part features to be inspected. Typically, edge images are obtained using information from the gray scale images provided by the cameras. Several techniques can be used to produce these edge images. Standard convolution type edge operators do not always provide an adequate edge image for the localization and recognition of all the edges in the image. Edges that are close together require further processing to separate them into distinct edge features and to relate them to the actual edges on the part. Furthermore, these types of operators produce extraneous edge information that may be difficult to recognize and reject. To alleviate this problem, an edge following algorithm was developed. This algorithm makes use of constraints, determined by the nature of the edge being sought, to follow an edge. The use of this kind of technique produces an edge image which includes only the edges of interest. Typically, this technique is used in an interactive mode when an operator is teaching the system about a new part. Once a part has been measured, a three dimensional model for the part can be created. This model can be used to great advantage when inspecting other parts of the same design. If a CAD data for the part is available, this interactive teaching procedure will not be necessary to determine the 3-D coordinates of edges. By using the more prominent features on the part to locate the part in three-dimensional space, the remaining edge features, obtained from the model, can be projected onto the edge images for any camera, using its calibration parameters, in the inspection system. This makes the automatic location of specific edge features on the part simple and straight forward since the selection of edges in each image can be controlled using the three dimensional part model.

Model or CAD-based recognition techniques have been recently under development for intensity and range images (Besl, 1988, Hoffman, et al, 1989, Fua and Leclerc, 1990, El-Hakim, 1990, Niemann, et al, 1990, and Flynn and Jain, 1991). A model of an object is built by, preferably, using existing CAD data, otherwise a

training procedure must be used if no such data exists, and then matched with its two-dimensional projection in the image (in case of intensity image) or with the three-dimensional image data (in case of range images). Using reference points on the object, such as holes or corners, its location and orientation in space coordinate system can be determined. Applying such procedure is essential for successful edge detection and object recognition. Edge detection methods based only on the image data are only successful in ideal situations, such as high contrast and nearly noise free images. Also having a model for the object helps tremendously in solving the correspondence problem in stereo vision (El-Hakim, 1990). For example, the model of a part is decomposed into more primitive objects thus restricting the search for matched stereo pairs to only those corresponding decomposed objects.

The work presented in this paper addresses the above problem of edge measurement for applications such as inspection, where the object is known, and reverse engineering, where creating a model, or CAD data, is the objective. The approach is implemented in the VCM (Vision-based Coordinate Measurement) system developed at the National Research Council of Canada.

2. THE VCM SYSTEM SOFTWARE

2.1. General Description of Software Options

Since VCM has been under development for several years, components have been constantly changing (El-Hakim, 1989a, 1989b, 1990, El-Hakim and Westmore, 1991, El-Hakim, et al, 1992). It has become obvious that in order for the system to be portable and easily adaptable to the rapid technology changes, the software must be hardware independent. Of course this may result in software that is not optimum for a specific computer hardware and also too slow since it does not rely on special image processing hardware. To avoid this problem, the VCM system now comes in two main forms (figure 1) :

1- An *application program*, which is hardware dependent and comes with an easy to use user interface and can be used without modifications in various applications. The present hardware consists of the following components that may be configured depending on the application (figure 2):

Computer hardware: PC-type computer (386 or 486), at least 2Mb RAM, and sufficient hard disk space;

Image processing hardware: Matrox IM-1280 with ASD (monochrome) or CLD (color) digitizer and RTP board (all fit inside the PC). Additional digitizer is added for 8-camera applications;

Cameras: Up to 8 CCD cameras (four stereo sets), each with 512 x 480 pixels, can be used;

Lenses: High quality F-mount lenses with c-mount adaptor are recommended rather than the regular c-mount video-camera lenses. Since the former are designed for the larger format 35mm film, they are expected to perform much better on the smaller image sensor format (usually about 9 mm x 7 mm) particularly regarding distortion and field curvature;

Camera mounts, and lights, including front and back lighting options.

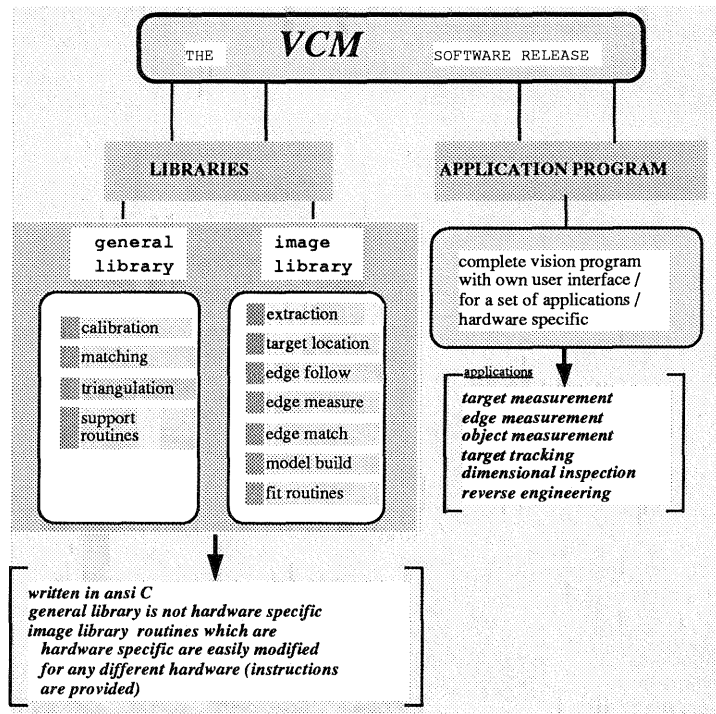


Figure 1: The VCM Software packages

Two variations of the system also exist:

- a. For application where very high accuracy measurement is required, two high resolution digital cameras such as the Videk Megaplus may be used. These cameras have 1320 (H) x 1035 (V) pixels arrays. The pixel size is 6.8 micrometers square with the same distance between pixel centers. This means that 100% of the sensor area is light sensitive (no measurable spacing between sensor elements), therefore aliasing is drastically reduced and sub-pixel accuracy is also improved. The camera features a built-in 8-bit A/D converter to take maximum advantage of the high signal-to-noise ratio available directly at the image sensor output. This eliminates signal degradation that normally occurs in amplifiers and transmission cables. The line jitter problem is also eliminated (Dahler, 1987). All other hardware components are as above.
- b. For a lower cost two-camera system, the Matrox IM-640 image processing board may be used.

2. Library routines in two main libraries.

a- The General Library: The Vision-based Coordinate Measurement `vcm_al` library is intended to be used as a development tool for stereo vision applications. It is not intended for specific hardware or a particular C compiler or an operating system. It provides operations on already extracted image coordinates. The main functions of the library are:

- 1- Camera calibration including distortion parameters.
- 2- Matching of corresponding points in stereo pair.
- 3- Computing 3-D coordinates by triangulation.

Several other routines, such as those required for memory allocation and point ordering and sorting, are also part of this library.

b- The Image Library: The `vcm_im` library provides functions for operations on images such as:

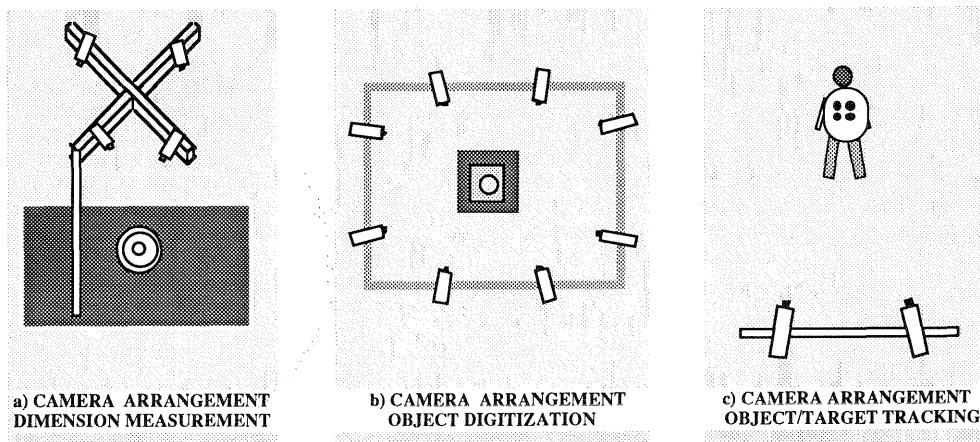


Figure 2: Examples of camera arrangements for various applications

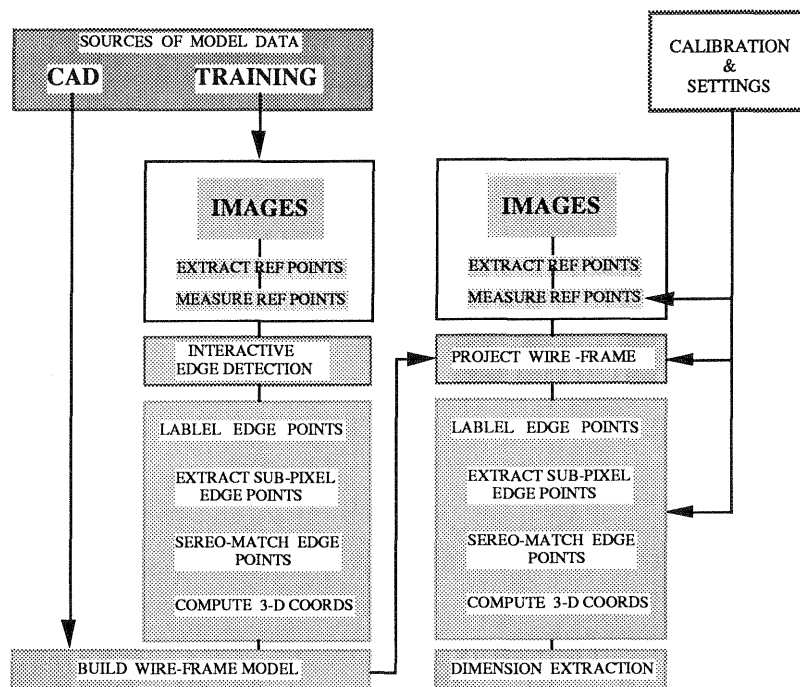


Figure 3: Edge measurement block diagram

1- Feature extraction routines for targets and edges. The edge routines include general convolutions, an interactive edge following (section 3.1.2) and edge extraction using a pre-defined model projected on the image.

2- Target and edge location to sub-pixel accuracy.

3- Stereo matching on edge points using constraints and / or models.

4- Building of object models for objects without CAD data (reverse engineering application)

5- Feature and object fitting routines, such as planes, lines, circles, cylinders, and spheres.

Some graphic routines are also included. The routines are not intended for specific hardware or a particular C compiler or an operating system. Some routines, however, such as those used to acquire pixel values or profiles, are image-board dependent and are easily modified by the user, in a given separate set of functions, without affecting the library routines.

2.2. The General Procedure for Dimensional Measurement on Edges

Figure 3 outlines the main processing steps for dimensional measurement on edges by VCM. The calibration (for camera and edge bias (El-Hakim, 1990)) and settings (calculating matching constraints) are carried out only once for a camera configuration and repeated only if the cameras have moved relative to each other or the lenses are refocused. Edge-bias calibration, is repeated if lighting conditions have drastically changed or if new types of edges are to be measured. The steps which are applied to the measurement of objects are as follows:

1) The object is placed in front of the 4-camera assembly (figure 2.a) with the side to be measured facing those cameras. Images are acquired, usually several times and averaged.

2) The target feature extraction process (El-Hakim, 1989, El-Hakim, et al, 1992) is applied to extract a minimum of three pre-defined reference points from all images. The target matching process is

then applied to determine the corresponding points followed by the computation of the 3-D coordinates by triangulation using the extended mathematical model for bundle adjustment.

3) The coordinates of the reference points and the camera parameters, determined earlier in calibration, are used to project the wire-frame model of the object on each image (section 3.1.3). If the process is successful, this should coincide with the edges of the image of the part (figure 7). The image of the wire-frame model is drawn in an overlay buffer on top of the image frame buffer so that the original image is not altered.

4) The wire-frame model will now guide the edge extraction routine (section 3.3) to extract edges in all four images. It is important to mention here that from each pair of images, only edges at angles larger than ± 45 degrees to the epipolar line are extracted, hence the requirement for four cameras to cover all edges. The reason for this is that matching of corresponding points on edges is not possible for edges parallel to the epipolar line and is unreliable for those at small angles to it.

5) For the left image in each pair, edge profiles are taken at preset intervals to reduce the edge to a set of points. The precise coordinates of these points are determined using sub-pixel pointing methods (Mitchell, et al, 1986). Edge-bias correction, as described in detail in an earlier publication (El-Hakim, 1990), is also applied at this step. For the right image, profiles are taken every pixel and the sub-pixel image coordinates of these edge points are computed. We now have a set of points for the left image and a denser set for the right image. The matching procedure will now be applied (section 3.4) and 3-D coordinates of the edge points are determined and separated, or labeled, according to each isolated edge.

6) Depending on each part, the three-dimensional coordinates are converted into appropriate dimensions, such as the radius of a circle or a cylinder, the distance between two parallel lines,

the distance between the center of a hole and a line, and so on.

3. Model-Based Edge Measurement

Several techniques can be used to produce these edge images. VCM user has a choice from three different techniques to locate and isolate edges:

- convolution with a gradient operator followed by image thresholding,
- an edge following algorithm,
- edge projection using an object model, a set of reference points to position the object, and camera calibration parameters.

Standard convolution type edge operators (the first option above) do not always provide an adequate edge image for the localization and recognition of all the edges in the image. To alleviate this problem, the edge following algorithm was developed (second option). This algorithm makes use of constraints determined by the nature of the edge being sought. The use of this kind of technique produces an edge image which includes only the edges of interest. Typically, this technique is used in an interactive mode when an operator is teaching the system about a new part.

Once a part has been measured, a model for the part can be created using the VCM. This model can be used to great advantage when inspecting other parts of the same design (the third option above). By using the more prominent features on the part to locate the part in three-dimensional space, the remaining edge features (obtained from the model) can be projected onto the edge images for any camera in the inspection system. This makes the automatic location of specific edge features on the part simple and straightforward since the selection of edges in each image can be controlled using the three dimensional part model. If CAD data exists, they can be use instead of the edge following to build the model of the part with VCM in the same manner.

3.1. Edge Extraction Procedure

The three alternatives for edge extractions using the VCM system are described in the following sections.

3.1.1. Edge Extraction Using Convolution Operators

A commonly used edge extraction technique involves the use of image gradients to identify prominent image gray scale transitions (edges). This technique is well known and will not be described in detail here. The basic procedure involves the convolution of the image with a kernel, representing a gradient operator, in order to create an image representing the image gradient. The resulting image is thresholded to produce a binary image in which the "on" pixels represent areas where the image gradient is strong. These types of edge operators work best when image contrast is good. However, these operators produce extraneous edge information that may be difficult to recognize and reject. Edges that are close together require further processing to separate them into distinct edge features and to relate them to the actual edges on the part. Weak edges may be difficult to separate from image noise.

3.1.2. The Edge Following Algorithm

The edge following technique developed for VCM makes use of a recursive path finding algorithm. Moving averages and constraints are employed to

help guide the search for an edge feature and detect and recover from errors during the search. The basic search mechanism will be described first and then the use of constraints to regulate the searching process will be discussed. The algorithm begins by selecting a starting edge point and goal point. The pixels neighboring the starting point are then examined and the best candidate (based on criteria to be described shortly) is selected as the next edge point. The process is repeated at the new edge point and subsequent points until either the vicinity of the goal point is reached or a point is reached where none of the candidates satisfy the minimum criteria. As edge points are found they are assigned a gray scale value representing the object feature of which they are a part. When a point is reached where none of the candidates satisfy the minimum criteria, the edge is followed backwards one pixel at a time until one of the previous losing candidate points can be used to move forward again. Points that are re-visited are marked with a gray scale value chosen to represent points that have been erased while searching for the edge in question. In order to help guide the search, certain parameters are recorded and used in the selection or rejection of edge point candidates. The basic algorithm is:

- 1) Obtain starting and ending points for each edge in the feature to be found.
- 2) Select a unique gray scale to represent the feature.
- 3) Select a unique gray scale to represent false edge segments that have been visited during the search and subsequently erased.
- 4) Find the starting direction "d₀" for the edge. This is done by calculating the image gradient at neighboring pixels, selecting the pixel with the largest gradient, and assigning an edge direction value according to its location relative to the starting point. The direction labelling is shown in figure 4. Depending on the type of edge feature being searched for, some of the neighboring pixels may not be included as candidates.

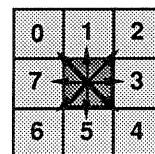


Figure 4: Direction Labelling

- 5) Use the following recursive procedure to follow the edge until the complete edge is found.
 - a) Check to see if the search is nearly complete. If the search has come within a certain distance of the goal point then the line can be finished simply by connecting the pixels in between the current point and the goal point. The reason this is done rather than letting the line finish itself is that there is no guarantee that the line will pass exactly through the goal point. The number of points in the edge is checked to make sure that the search is not terminated when first leaving the starting point.
 - b) Check to see if any of the constraints have been violated. If any constraint is violated then the current edge pixel should be rejected as an edge pixel and the search should back up to the previous edge pixel.

c) If none of the constraints have been violated, then the neighboring pixels are examined in order to select the next edge pixel. The next pixel is selected by following these steps.

i) Calculate the "current average" value for the edge direction " d_{avg} " according to the formula,

$$d_{avg} = \frac{\sum_{i=0}^{n-1} d_i k_d^{n-i-1}}{\sum_{i=0}^{n-1} k_d^i}$$

where, n is the number of edge points in the sequence,

i is an index value representing the i th edge point in the sequence,

d_i is the direction from the i th edge point to the next edge point,

d_{avg} is a weighted average of the d_i values,

k_d is a gain factor used for weighting the d_i values.

This calculation can best be performed using the following recursive equations to perform an update of the weighted average when each new edge pixel is found,

$$k_d^n = k_d \times k_d^{n-1}$$

$$\sum_{i=0}^n k_d^i = k_d^n + \sum_{i=0}^{n-1} k_d^i$$

$$\sum_{i=0}^n d_i k_d^{i-1} = d_n + k_d \times \sum_{i=0}^{n-1} d_i k_d^{i-1}$$

When the gain factor k_d is less than one, the more recent points contribute most strongly to the average value.

ii) The value of " d_{avg} " is used to select candidates for the next edge pixel. The pixel whose direction is closest to " d_{avg} ", and the two pixels next to it on the direction diagram are the only three pixels that will ever be considered (even on retraces). This is done to prevent sharp turns from being taken. Pixels that have already been considered as edge pixels, whether accepted or later determined to belong to false edge segments, are eliminated as candidates. Figure 5 shows an example of selecting edge pixel candidates based on the value of " d_{avg} ".

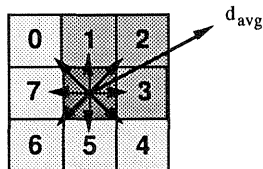


Figure 5: Edge Pixel Candidates

iii) Calculate the magnitude of the image gradient at the current pixel location and select a threshold value based upon the value obtained (a threshold value around 40% works well). Setting the threshold in this matter allows it to be adapted according to the relative strength of the edge in the area being searched at the time, thus allowing the traversal of weak segments of the contour.

iv) Calculate the magnitude of the image gradient at each of the candidate pixel locations, and select the pixel having the largest gradient magnitude value. If this value is not larger than the threshold then none of the candidate pixels are considered to be appropriate and the search should back up to the previous pixel.

v) If the next pixel is found then its value in the output image is set to the value chosen to represent the feature.

Steps (a)–(c) are repeated as long as the search does not have to back up. When the search must back up the following steps are followed:

d) The last pixel to be added to the edge is removed. Its value in the output image is reset to the value chosen to represent false edge segments that were found while searching for the feature. Pixels marked as erased are eliminated as edge candidates.

e) The effect of the removed pixel upon the value of " d_{avg} " is eliminated by performing the inverse calculations corresponding to the recursive equations presented earlier. The inverse equations are presented below.

$$\sum_{i=0}^n d_i k_d^{n-1} = \frac{\left(\sum_{i=0}^{n+1} d_i k_d^{n+1-i} \right) - d_{n+1}}{k_d}$$

$$\sum_{i=0}^n k_d^i = \left(\sum_{i=0}^{n+1} k_d^i \right) - k_d^{n+1}$$

$$k_d^n = \frac{k_d^{n+1}}{k_d}$$

f) Go back to step (a). The next best candidate for the next edge pixel will be chosen since the previous choices which led to false edge segments have been removed as candidates. If necessary, the third choice will be selected if the search fails again. Of course if there are no suitable candidates the search will continue to back up until a successful forward search is completed or until the search backs up all the way to the starting point (at which point the attempt to find the edge has failed completely).

The search process can be guided and controlled by maintaining selected parameter values during the search and ensuring that the parameter values do not exceed selected constraints. When a constraint is violated during the search for an edge, the search must back up one pixel and try to go forwards again. Methods for applying constraints when searching for straight lines and circles have been developed and will now be described.

When searching for a fairly straight line, the constraint that the next pixel should bring the search at least a certain distance closer to the goal point can be used. At each potential edge point, the distance between the point and the goal point is calculated. The difference between this distance and the distance calculated for the previously selected point is calculated and compared to a threshold value. It was found that a threshold value of 0.85 pixels works quite well in most cases. When searching for edges where there is little chance of interference from other nearby edges the use of this constraint condition was usually not necessary. The application of this constraint was found to be very useful when the

correct edge became very weak in close proximity to stronger edges that actually came into contact with the edge of interest. Sometimes the search may follow the wrong edge and not recover immediately, but eventually the constraint that the search proceed towards the goal point will be violated and the search will eventually back up to a point where it can proceed along the correct edge.

When searching for a roughly circular edge, VCM makes use of two constraints. The first constraint is that the search remain inside a ring. The other constraint is that the distance from potential edge points to the estimated center of the circle not vary too much from the current average value for this distance. In order to apply these two constraints the algorithm must be supplied with a rough estimate of where the circle's center is. In the following discussion, the distance from the i^{th} edge point to the estimated circle center will be referred to as r_i . The ring to which the edge search is confined is obtained by calculating minimum and maximum allowable r values based on the value for r_0 . This allows for the fact that a circular edge may appear as an ellipse in an image. Figure 6 illustrates this concept.

The "current average" value for "r" is calculated according to the following formula.

$$r_{\text{avg}} = \frac{\sum_{i=0}^n r_i k_r^{n-1}}{\sum_{i=0}^n k_r^i}$$

This equation is of the same form as the calculation used to maintain d_{avg} as part of the basic algorithm. Therefore it can be updated using the same form of recursive formulas. These calculations can also be performed at the same time as the update calculations for d_{avg} are being made. When the search has to back up, the effect of the removed pixel upon the value for r_{avg} must be undone in the same manner as is used for d_{avg} . At each potential edge point the value for r_i is calculated. If the difference between r_i and r_{avg} is greater than a threshold value then the search will back up. The application of this constraint helps to prevent the search from leaving the circular edge by ensuring that the r values do not change too quickly. The r values are compared to the current average value rather than the previous value because it was found that better performance was achieved when the average value is used (the average value represents the recent trend rather than just an isolated occurrence).

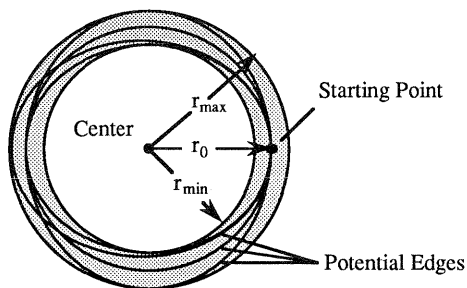


Figure 6: Circular Edge "Ring" Constraint

The edge following as described above is also used as the main tool for reverse engineering applications. Since the process is interactive, it can be applied to parts which have no model or CAD data.

3.1.3. Edge Projection Using an Object Model

One of the main application areas for which VCM was developed is that of part inspection. In most part inspection applications, the part being inspected is known. This fact can be exploited in order to greatly simplify the edge extraction process. When a model exists for the part and the system can easily locate the part in three-dimensional space, then all edges of interest can be projected onto the camera image plane for any camera in the system.

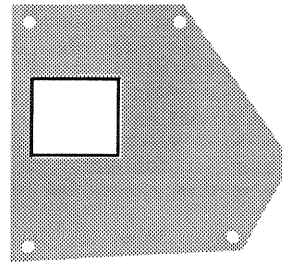


Figure 7: Example of a sheet-metal part with reference points (holes) and edges of interest (the square), the wire frame is drawn with dark lines.

The basic technique is as follows (see figure 7). First the part is located by finding and measuring the position of the most prominent features on the part. Features for locating the part should be chosen based on the ability of the system to use very general methods to find these features. Features like outside edges or holes are ideal. In the current system, only hole-type features have been used to locate parts. These features are used to locate the part reference frame with respect to the vision system reference frame. The part reference frame is a reference frame which can be thought of as being fixed to the part, and moves when the part moves. The part model is also described with respect to this fixed reference frame. The expected position of any feature defined by the model with respect to the vision system coordinate system can thus be calculated once the location of the part is known. Points in space can be projected onto the camera image planes using the system model. The expected location, in any image, of any feature described by the model can thus be found. An edge image can be created by projecting points belonging to individual features on to the image and connecting the resulting image points to form a contour. Since actual parts being inspected will differ slightly from their models, the use of this technique produces only an estimate of what the edge image will look like. It has been found in practice that this is sufficient to locate and measure the real edge in the image. One of the main advantages of this technique is that it greatly simplifies the task of finding the more difficult features. It does not matter if the contrast of the feature with its surroundings is poor or if noise is present. It also simplifies the task of automating the measurement process. The model provides all the information the system needs to find all the features to be measured.

VCM accepts object models consisting of a file of ordered edge points labelled according to the object feature to which they belong. Other types of models such as CAD models or models created from measurements provided by other systems could be incorporated by creating suitable files from the model data and using the model building routine in the VCM.

3.2. Stereo Matching and Computing 3-D Coordinates of Edge Points

Although the constraints are similar, the matching procedure for edge points differs somewhat from that used for targets (El-Hakim, 1989). Since no isolated points exist, some must be extracted from the edges. For the left image in each pair, edge profiles are taken at preset intervals to reduce the edge to an appropriate set of points and the sub-pixel coordinates of these points are determined by convolving a step-edge function with the Gaussian image function and equating it to the edge profile followed by a least-squares adjustment to find the unknown parameters of the step edge. For the right image, since the corresponding points are not yet determined, profiles are taken every pixel and the sub-pixel image coordinates of these edge points are computed. We now have a set of points for the left image and a denser set for the right image. The matching procedure is then applied as follows:

1. For each edge point from the left image, the epipolar line on the right image is computed;
2. The epipolar line will intersect several edges in the right image, out of which only one is correct;
3. Constraints, such as the disparity and depth constraint, plus the added knowledge of the shape of the complete edge, are applied to find the correct match.

Some edges are determined from one pair, while the others are determined from the other pair to avoid those parallel or at small angles to the epipolar line. The 3-D coordinates of the edge points are determined and separated, or labeled, according to each isolated edge.

3.3. Automated Part Measurement

The current VCM system is capable of automatically measuring a finite set of parts. Some parts can be measured automatically with very little operator intervention while the automated measurement of some other parts require that a setting up process be performed by an operator familiar with the measurement processes involved. In order to fully automate the inspection of a variety of parts, it was necessary to develop a strategy for dealing with parts in a very general way. Since it would be difficult to foresee all the possible requirements associated with newly encountered parts, it was recognized that the design strategy should allow for straightforward system expansion.

The following measurement strategy has been designed to meet the requirements just outlined (see also figure 3). The automatic inspection process has been divided into two main steps. In the first step, the system acquires all the measurement data, using a wire-frame model as a guide (see section 3.1.3). After this step, the measurement data will exist as sets of labelled points. In the second step, the system uses the data to perform the inspection. Dividing the inspection process into these two main steps has the benefit of allowing for the possible future integration of the system with other measurement devices. Since the second part of the measurement process receives its input as sets of labelled points, these points would not have to necessarily be provided by the machine vision system. It is possible that the points come from several different sources such as a coordinate measuring machine or laser scanner. Thus, the possibility for a multi-sensor inspection system exists.

The measurement step consists of two tasks. The first task is to locate the part. In order to perform part projection, the part must be located by the system.

This can be done using some of the more prominent features of the part. Using some of the features to locate the part and then using the model to find the rest of the part features is preferable to having the system find and identify all of the features without any guidance. At the current time, target type features have been successfully used to locate parts so that part models can be applied. Primary edge features could also be used to locate parts. Once the part has been located, the second task, acquiring data points, can be performed. Each distinct feature, as defined in the model, is measured separately. Using the edge projection technique to mark features uniquely or in isolation removes the problem of having to deal with multiple edges when performing matching of edge points, thus virtually removing the possibility for error. Another benefit of measuring features separately is that the data points belonging to specific features are easily maintained as sets of data points labelled according to the feature in the model to which they belong, thus alleviating the need to perform feature recognition on the data sets.

The second part of the measurement process consists of dealing with the measured data. This step can also be divided into two parts. In the first part, the system is provided with the information it needs to derive further feature information from the measured data. Such features might include corner points, surfaces, or even invisible features such as circle centers and imaginary lines and boundaries. Once these features are obtained they may be used in calculations in the same manner as the directly measurable features. Inspection calculations can then be driven by a list of inspection orders. Each inspection order will give all the information necessary to calculate one parameter value. The first field in the inspection order will not only identify the parameter to be calculated but will also uniquely identify the subroutine used to calculate it. Subsequent fields will identify the data sets involved in calculating the parameter and the tolerance specifications for the parameter. For example, the entry "distance_point_to_line 63 27 19.99 20.01" would be a request to calculate the distance from "point 63" to "line 27" and make sure the value is between 19.99 cms and 20.01 cms. As new types of inspection tasks are encountered, new routines can easily be added to the system to handle them without having to redesign the rest of the measurement system.

4. CONCLUDING REMARKS

The edge measurement procedure described is useful for applications where the object is known (dimensional inspection and object tracking) as well as those where the object has no available CAD data or model and it is required to create such model (reverse engineering). In the inspection and tracking applications, the procedure is fully automated, except for the set-up, while in reverse engineering the procedure is interactive. The VCM system which applies these procedures is available in two forms: an application program, targeted for the above mentioned applications, and is hardware specific; and library routines which are not hardware specific and not targeted for a specific application. The later is intended for users expected to develop their own application oriented software and user interface.

5. REFERENCES

- Besl, P.J., 1988. Geometric modeling and computer vision. *Proceedings of IEEE*, 76(8), pp. 936-958.
- Dahler, J., 1987. Problems in digital image acquisition with CCD cameras. In: Proceedings of ISPRS Intercommission conference on fast processing of photogrammetric data, pp. 48-59.
- El-Hakim, S.F., 1989a. A hierarchical approach to stereo vision. *Photogrammetric Engineering and Remote Sensing*, 55(4), pp. 443-448.
- El-Hakim, S.F., 1989b. A stereo vision system for on-machine dimensional metrology. In: Proceedings of SPIE Vol.1095, *Applications of Artificial Intelligence VII*, pp. 310-320.
- El-Hakim, S.F., 1990. Some solutions to vision dimensional metrology problems. In: Proceedings of SPIE Vol. 1395, *Close-Range Photogrammetry Meets Machine Vision*, pp. 480-487.
- El-Hakim, S.F. and D.Westmore, 1991. Performance evaluation of a vision dimension metrology system. In: Proceedings of SPIE Vol. 1526, *Industrial Vision Metrology*, pp.56-67.
- El-Hakim, S.F., N.J.Pizzi, and D.B.Westmore, 1992. The VCM automated 3-D measurement system - theory, application, and performance evaluation. In: Proceedings of SPIE Vol.1708, *Applications of Artificial Intelligence X: Machine Vision and Robotics*, pp. 460-482.
- Flynn, P.J. and A. K. Jain, 1991. CAD-based computer vision: from CAD models to relational graphics. *IEEE Trans. on PAMI*, 13(2) , pp. 114-132.
- Fua P. and Y. G. Leclerc, 1990. Model driven edge detection", *Machine Vision and Applications*, Vol. 3, pp. 45-56.
- Hoffman, P., H. R. Keshavan, and F. Tawfiq, 1989. CAD-driven machine vision. *IEEE Trans. on Systems, Man, and Cybernetics*, 19(6), pp. 1477-1488.
- Mitchell, R.O., E. P. Lyvers, K. A. Dunkelberger and M. L. Akey, 1986. Recent results in precision measurements of edges, angles, areas and perimeters. In: Proceedings of SPIE Vol. 730, *Automated Inspection and Measurement*, pp. 123-134.
- Niemann, H., H. Buenig, R. Salzburnn, and S. Schoeder, 1990. A knowledge-based system for industrial applications. *Machine Vision and Applications*, Vol. 3, pp. 201-229.