IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661,p-ISSN: 2278-8727, Volume 18, Issue 4, Ver. III (Jul.-Aug. 2016), PP 53-60
www.iosrjournals.org

Pose Estimation in AVI from Unknown Correspondences Using Genetic Algorithm

A. Mukhopadhyay[1], S. Hati[2]
[1]*(Computer Science and Engineering, Gargi Memorial Institute of Technology, India)*
[2]*(Computer Science and Engineering, Budge Budge Institute of Technology, India)*

Abstract: *In this paper we propose a model based approach to determine correspondences and pose of objects in automated visual inspection applications. The method does not consider correspondence and pose estimation problems as two distinct processes. Rather they are viewed as two cooperative processes and implemented through a proposed new composite chromosome structure of Genetic Algorithm (GA). GA evolves sets of poses and matched vertices sequences and the evolution process continues until a good accuracy in the values of pose parameters are obtained. The strength and robustness of our method are demonstrated by thorough testing on both simulated and real data in an Automated Visual Inspection (AVI) system.*

Keywords: *Pose Estimation, Correspondences, AVI, Genetic Algorithm.*

I. Introduction

Pose Estimation is one of the fundamental problems in Robotics and Computer Vision. It is used to determine 3-D orientation and position of an object with respect to a model reference frame. Pose estimation find applications in inspection problems [1]. In manufacturing, the visual inspection is one of the steps in the cycle of production of parts and the inspection is automated at relatively slow pace in the industry. Several inspection tasks need a substantial quantity of reasoning capability to accept or reject choices, and to classify the type of defects. Although Automated Inspection might seem to be a panacea for improving quality and reducing cost, it may not be always possible. For automated inspection to be feasible, it must run in real time, be consistent, reliable, robust and cost effective. Furthermore, for automated inspection to compete with the human inspectors, the inspection procedure must be able to handle objects that are at unknown positions and orientations. *Flexibility* of the object or part whose position and orientation is less constrained under inspection would increase the number of automated visual inspection applications. Flexibility is thus one of the important research issues in the automated visual inspection arena [2]. Registration of a sensed object model to a reference object model is termed as pose estimation since it is the pose transformation that must be estimated in order to perform the registration. It is to be noted that in computer vision, pose estimation is usually required to perform object recognition. A robust object recognition method provides good accuracy even when the pose estimation of the object is not accurate. Hence, object recognition algorithms do not give much emphasis on the high accuracy of the estimation of pose transformation. On the other hand, the main objective in inspection is to compute the differences between the stored reference model and the sensed object model. That is why the registration of the sensed model with the stored reference model should be highly accurate and the difference between these two models is not due to the improper registration process. The objective should therefore be to design a more robust pose estimation process in which the matching between the image and the object model will not be assumed a priori. Ideally, matching and pose estimation processes should be integrated in a common framework and should work cooperatively, so that any hypothesized matching can be validated with geometrical constraints and estimated poses. We have used genetic algorithm to solve pose estimation from unknown correspondences and have proposed a composite chromosome to tackle both the problems of pose estimation and matching. The composite structures of the chromosomes used by genetic algorithm help us to implement both matching and pose estimation as integrated and cooperative processes.

We discuss the previous work on pose estimation and matching in Section 2. Section 3 presents the mathematical formulation of pose estimation problem. In this section we have also discussed the pose estimation and matching using GA. In Section 4 the results of extensive computer experiments are reported. The final section presents the conclusions.

II. Related Work

Both correspondence and pose estimation problems are found in model-assisted object recognition. In model-assisted matching, models are brought into correspondence with an image of the object to be recognized. There are three classes of matching problems found in the literature viz., (i) Pruned search matching, (ii) Non-correspondence matching, and (iii) Search in model-to-image transformation space. Pruned search technique requires that the maximum subset of paired model and image features under a single transformation. But maximum subset selection is of the order of exponential complexity. In non-correspondence matching, a transformation is computed from the global features of a model and an image. In this method, no correspondence is assumed between individual parts or features of the model and the image. Cyganski and Orr [3] propose a non-correspondence approach where a tensor is computed from the moment of inertia of the image. The non-correspondence matching requires that every part of the object should be visible. The method fails if certain parts of the object are not visible and the scene contains multiple objects. Methods based on the searching for solution in transformation space hypothesizes matching and hence determine the transformation there from. David et al. [4] determine the poses by combining two techniques - for matching they have used Soft Assign algorithm[5], and Dementhons iterative POSIT algorithm [6] for determining object pose under a full-perspective camera model. Hati and Sengupta [7] have demonstrated that Artificial Neural Network based computation of pose transformation matrix is robust with respect to noise. They assume a matched sequence of images and 3-D object model vertices. Hence the objective is to synthesize a more powerful and robust pose estimation method where matching between image model and object model will not be assumed a priori. Matching and pose estimation procedures ought to be coordinated in a typical structure and ought to work agreeably, with the goal that any assumed matching can be accepted with geometrical constraints and evaluated poses. We have utilized genetic algorithm to unravel pose estimation from unknown correspondences and have proposed a composite chromosome to handle both the issues of pose estimation and matching. The composite structures of the chromosomes utilized by genetic algorithm help us to realize matching and pose estimation as a coordinated and agreeable methodology. Schweighofer and Pinz [8] determine the position and orientation of a camera from the knowledge of given four co-planar points in real time. They have shown that pose determination from planar targets suffers from pose ambiguities. In their work they find two local minima of the error function that exist for wide angle lenses and closed range planar targets. They give a thorough explanation of the two minima and develop an algorithm for unique and robust pose estimation from a planar target. Lu, Shao and Xiao [9] treat pose estimation and segmentation as inter-wined problem that provide cues for each other. They put object segmentation and pose estimation into a joint optimization. Pose estimation like image segmentation is formulated as a binary optimization problem. The top-down pose shaped cues, bottom-up visual cues, and the consistency constrained are used to derive the final ob-jective function. They have verified their results on Ramanan benchmark data-set. Assa and Janabi [10] propose multi-camera sensor fusion technique to pose estimation. They have employed Kalman filter based sensor fusion approach for pose estimation. The method is robust to camera motion and image occlusion.

Our approach based on GA search simultaneously both in matching space as well as in transformation space. Hence, it consists of two objective functions, one objective function evaluating how good the model to image matching is and another objective function looks for model to image transformation. Classical optimization techniques transform multiple objective functions into a scalar function and find a single pareto - optimal solution at a time. Genetic algorithm is well suited to multi-objective optimization. Parameters can search for multiple solutions, eventually taking advantage of any similarities available in the family of possible solutions to the problem [11].

III. Pose Estimation Problem

Let the n points in object space be denoted by $x_i = (x, y, z)^T$, i = 1, …, n. Let the corresponding image points in 2D image space after perspective projection be (p_{i1}, p_{i2}), i = 1, …, n. The object space to image space relationship is given by [12],

$$p_{i1} = f\frac{r_1 x_i + t_1}{r_3 x_i + t_3} \qquad (1)$$

$$p_{i2} = f\frac{r_2 x_i + t_2}{r_3 x_i + t_3} \qquad (2)$$

$$\boldsymbol{t} = (t_1, t_2, t_3)'$$

$$R = (r_1, r_2, r_3)'$$

Where f is the focal length of the camera, ***t*** is the 3-D translation vector, and **R** is the 3 × 3 rotational matrix. Let us assume that the object under inspection is on a plain surface and has three degrees of freedom – t_1 and t_2, two translation parameters along X and Y axes and θ, the rotational parameter about Z-axis. Our problem is to determine the translation vector ***t*** and the rotation matrix R from the unknown correspondences of x_i and p_i, *i = 1, …, n*. Fig. 1 demonstrates the our model-based AVI system. Here we consider a flexible automated visual inspection system where the object under inspection may get displaced from its pre-defined orientation and position. Let the test object has a pose ($\Delta x, \Delta y, \Delta\theta$), where Δx and Δy denote the translation of the object along *X* and *Y* axes and θ denote the rotation of the object about Z-axis with respect to the pre-defined model orientation and position. Eq (1) and Eq (2) can be expanded to Eq (3) and Eq (4) as given below [13].

$$p_{i1} = f\frac{K_1}{K} \qquad (3)$$

$$p_{i2} = f\frac{K_2}{K} \qquad (4)$$

Where

$K_1 = (x_i + \Delta x - x_0)\cos(\theta + \Delta\theta) + (y_i + \Delta y - y_0)\sin(\theta + \Delta\theta);$

$K_2 = -(x_i + \Delta x - x_0)\sin(\theta + \Delta\theta)\cos\gamma + (y_i + \Delta y - y_0)\cos(\theta + \Delta\theta)\cos\gamma + (z_i - z_0)\sin\gamma;$

$K = -(x_i + \Delta x - x_0)\sin(\theta + \Delta\theta)\sin\gamma + (y_i + \Delta y - y_0)\cos(\theta + \Delta\theta)\sin\gamma + (z_i - z_0)\cos\gamma + f.$

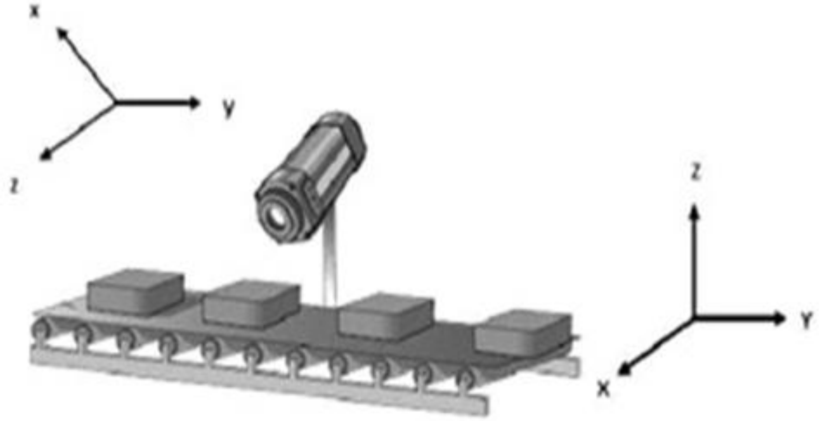


Figure 1: Model diagram of our AVI system

In Fig. 2 we have shown the schematic diagram of our pose estimation system using genetic algorithm. The computer representation of the object to be inspected is done using solid modeling technique [14]. Edge based data structure – Winged-edge data structure [15] is used to store and represent the vertices, edges and faces of the solid object to be inspected. For a given camera position and orientation, we extract the visible view of the object to be inspected using hidden edge removal algorithm. The visible view consists of visible edges and

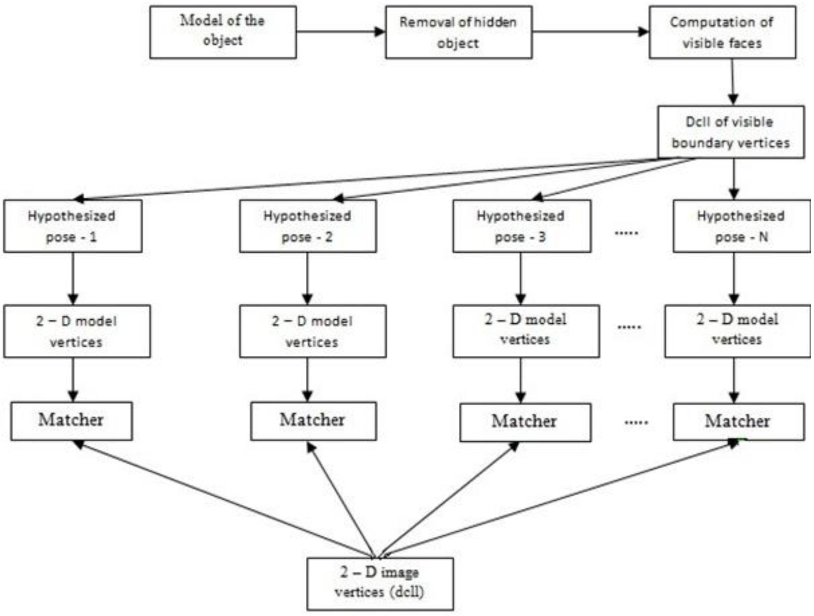


Figure 2: Block diagram of our model based Pose Estimation System

faces. Output of the hidden edge removal algorithm is the set of visible edges. We construct the visible faces from these visible edge segments using Franklin's algorithm [5]. The visible edge segments are kept in a doubly circular linked list. This link list is referred to as "list of 3D model vertices". Now, the edges from the boundary of the image of the object are extracted. These edges are listed in a doubly connected circular linked list. This link list is referred to as “list of 2D image vertices”. From Fig. 2, it is clear that for each of the estimated poses in a set of chromosomes, the list of 3-D model vertices from the doubly connected circular linked list will be changed by the comparing pose, according to Equations (3) and (4) and we might get the relating anticipated set of 2-D model vertices. For each of the conjectured representations, the matcher will assess the closeness of the 2-D list of model vertices with the 2-D list of image vertices. Let **B** = {b_1, b_2 , …, b_M} be the m image points at the boundary of the processed image corresponding to the n model points at the boundary of the model view stored in **A** = {a_1, a_2, …, a_N}. The matcher establishes the correspondences between the sets **A** and **B** and it utilizes genetic algorithm to hypothesize both pose parameters and matching at the same time.

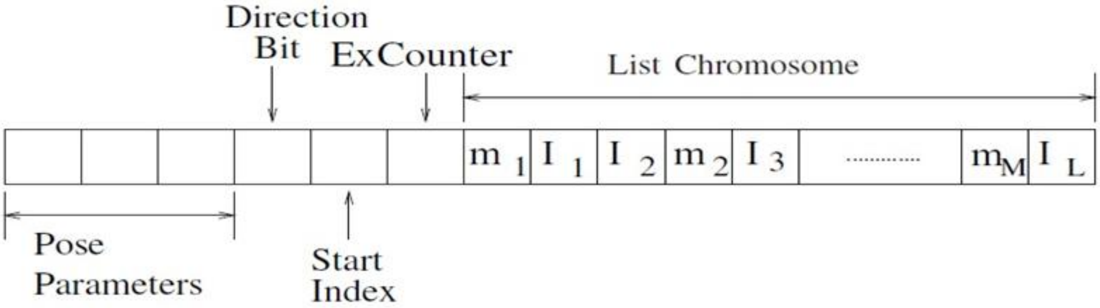


Figure 3: Reference Diagram of a Composite Chromosome

Genetic algorithm uses a composite chromosome comprising of a real chromosome of pose parameters and a list chromosome of vertices. In Fig. 3 we have demonstrated the schematic diagram of a composite chromosome. Different fields of the composite chromosome are detailed as follows.

Pose Parameters: Three pose parameters (Δx, Δy, Δθ) are represented by three fields of the real chromosome.

Direction Bit: The direction bit can assume values either 0 or 1 for either clockwise or anti-clockwise directions of traversals respectively along circular doubly connected linked list matching sequence.

Start Index: This field generates the starting vertex index from which the correspondence starts between the sets A and B.

Excess Counter: This field generates the number of spurious or missing vertices.

List Chromosome: This chromosome consists of the list of integers. These integers are both model vertices and image vertices indices. Genetic algorithm generates random orders of these model and image vertices indices.

We distinguish the image vertices indices from the model vertices indices in a list chromosome by re indexing the image vertices indices. We re-index the image vertices indices by L + i, where i = 0, 1, 2, ..., I − 1 are image vertices indices and L is a constant indicating the maximum value of the model vertices indices. The model vertices indices are indicated by j = 0, 1, 2, …, M − 1 where M < L. It is to be noted that the number of model vertices and number of image vertices are known to us from solid model and the image analysis module respectively and these two numbers are constant throughout the chromosome set. The following two cases may arise depending on the number of model and image vertices detected.

Case-1: M > I Here, it is at least (M − I) unmatched vertices from the model. Let P (<= I) be the number of matched model and image vertices. Therefore, we have (I − P) number of unmatched image vertices (referred to as “spurious”) and (M −P) number of unmatched model vertices (referred to as “missing”). The excess counter field E contains the hypothesized count of missing vertices. This will be used to pick up those many number of missing model vertices from the list chromosome. In the example illustrated in Fig. 3, the list chromosome contains image vertices indices I_j and model vertices’ indices m_i, interleaved in any random way. Here, each of the m_i’s may assume any value in the range $0 <= m_i <= L - 1$ and each of I_j ’s may assume any value in the range $L <= I_j <= L + M - 1 + SP$, where SP is the number of spurious vertices. Now the first E vertices of the list chromosomes which are model vertices, will be regarded as “missing”. We would correspondingly consult the doubly-connected linked list for the model and leave these vertices from matching. Since, genetic algorithm generates and hypothesized E, it is now possible to derive the number of spurious image vertices SP which is given by,

$$SP = I - M + E \qquad (5)$$

The above equation can be explained as follows. The number of model vertices is (M − E) which are to be matched and consequently the number of image vertices which are spurious is I − (M − E). We consider now the first SP number of vertices of the image from the list chromosome, which are labeled as $I_L I_{L+1} \ldots I_{L+SP}$, which will be labeled as ”spurious”. We will leave out these vertices from the doubly connected linked list of image vertices and exactly (M − E) number of remaining model and image vertices in the linked list will be considered for matching in the given order.

Case-2: M <=I Here, genetic algorithm generates the ExcessCounter value E as the number of hypothesized spurious vertices. The number of missing vertices (MI), as in Case-1, can be shown to be

$$\mathrm{MI} = \mathrm{I} - \mathrm{M} + \mathrm{E} \qquad (6)$$

1. Fitness Function

Genetic algorithm evolves the composite chromosomes and these chromosomes exhibit randomness with respect to pose parameters, as well as the hypothesized match sequence that is derived from the start vertex index, the excess count and the list chromosome structure. Each composite chromosome should be evaluated for its fitness with respect to combined pose estimation and matching. We derive two counts from a given chromosome – the missing vertex count *MI* and the spurious vertex count *SP*. We mark the corresponding missing and spurious vertices from the 3-D model dcll and the 2-D image dcll respectively. Then we have P number of unmarked and matchable vertices from both the dclls. Now we want to determine a matching score. For this purpose we first require to transform all the P unmatched vertices from the 3-D model dcll to the projected 2-D space as per Equations (3) and (4) using the hypothesized pose parameters given in the chromosome under consideration.

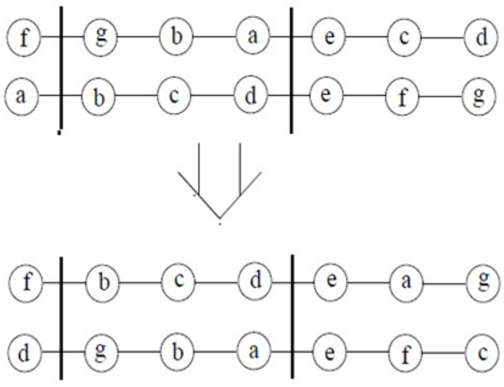


Figure 4: Block Diagram of list order Crossover

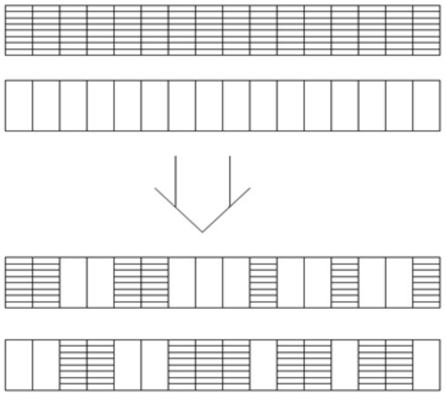


Figure 5: Block Diagram of Uniform Crossover

Let us suppose that $c_1, c_2, \ldots, c_p$ as the transformed unmarked model vertices in 2-D and $d_1, d_2, \ldots, d_p$ as the unmarked image vertices in 2D space (see Fig. 2). Let the co-ordinates of the vertices $c_1, c_2, \ldots, c_p$ be (x_1, y_1), (x_2, y_2), …, (x_p, y_p) respectively and those of the vertices $d_1, d_2, \ldots, d_p$ be (x_1', y_1'), (x_2', y_2')…, (x_p', y_p') respectively. In terms of these coordinates, the fitness function of the chromosome is given by,

$$F = \sum_{i=1}^{P} (|x_i - x_i'| + |y_i - y_i'|) \qquad (7)$$

2. Genetic Crossover and Evolution of the Chromosomes

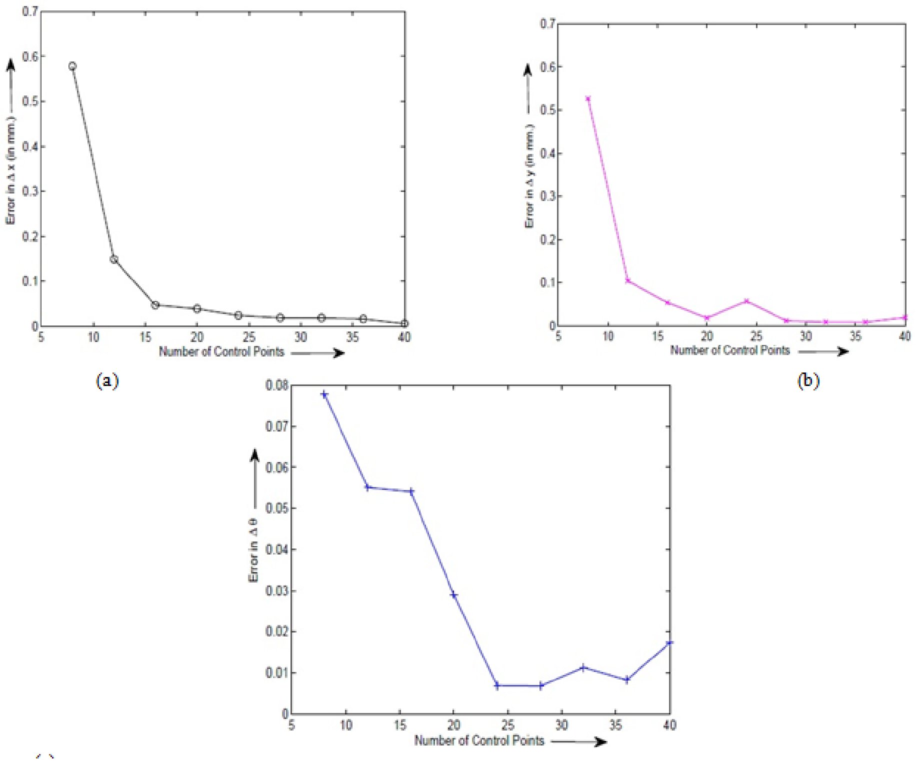
Initially we create a set of composite chromosomes by randomly varying the fields, viz., pose parameters, direction bit, start vertex index, number of missing or spurious vertices and the list chromosome indices. Matching and pose parameters are hypothesized in this way. Then genetic algorithm operators i.e., crossover and mutation are applied on these chromosomes. It is to be noted that crossover and mutation operators are invoked separately for each part of the composite chromosome. The uniform crossover operation is applied on real chromosomes and listorder crossover operation is applied on the list chromosomes. They are explained in Fig. 4 and Fig. 5 respectively. A random unit Gaussian noise is added to each field of pose parameters for the mutation of real chromosomes. For the mutation of the list chromosomes we have used swap mutators which randomly selects two positions in the list and exchanges the contents of the nodes in these two positions. Let $(c_1 c_2 \ldots c_i \ldots c_j \ldots c_n)$ be a list chromosome before mutation. After mutation the exchange of the contents in node i with that in node j, the list chromosome reduce to ($c_1 c_2 \dots c_j \dots c_i \dots c_n$). After the crossover and the mutation operations, a new generation is created. In this new generation, the chromosomes consist of parents as well as their offsprings and all these chromosomes are evaluated for their fitness function and exactly the original number of chromosomes is retained according to their fitness values. The lots of chromosomes with low fitness values are eliminated. These fit chromosomes in the new generation go through the processes of selection, crossover and mutation again and the whole process is repeated again and again, till the convergence is obtained.

IV. Results and Discussions

The genetic evolution of the proposed composite chromosome structure is tested using (1) simulated point vertices (2) real image data. Various experiments performed are described below.

1. Pose Estimation from a set of simulated vertices

In this experiment, we simulate 3D vertices in world coordinates with respect to a fixed camera position and orientation. The vertices are then subjected to a rigid body translation and rotation. We estimate the pose parameters and the results so obtained are compared with the actual values. The experiment is done in the following way. We generate a set of N vertices, starting with N=12. The co-ordinates of the vertices are generated using uniform distribution. The interval chosen for the x and y co-ordinates is [100, 400] and that of z-coordinate is [0, 150]. The pose of the camera is fixed during the experiments. The position and orientation of the camera does not affect the estimation results and hence an arbitrary viewing position and orientation of the camera is chosen. The gimbal center of the camera is chosen at the point (0, 0, 200) and the tilting and spanning angles of the camera are chosen at 120^0 and 135^0 respectively.



(a)

(b)

(c)

Figure 6: Error in Pose Estimation Parameters against the Number of Control Points

We demonstrate the robustness of GA-based integrated correspondences and pose estimation by conducting a set of 100 trials. Estimation error in x, y and θ are plotted in Fig. 6 for one of these trials. As no conclusion can be made from the results of one experiment and the variability indicated in the graph are random in nature, a set of definite conclusions can be made from the results of 100 trials.

- The maximum error in x and y is less than 1 unit of world coordinate and is well below 1 degree for the parameter θ.
- Even at lower values of N, the accuracy in pose estimation is excellent and similar accuracy is not possible

from classical methods, such as in Haralick et. al. [16].

- The approach is robust as at 20 dB SNR with 10% missing and 10% spurious vertices and the results are well accepted.

1.1 Correctness of Correspondences from the Set of Simulated Points

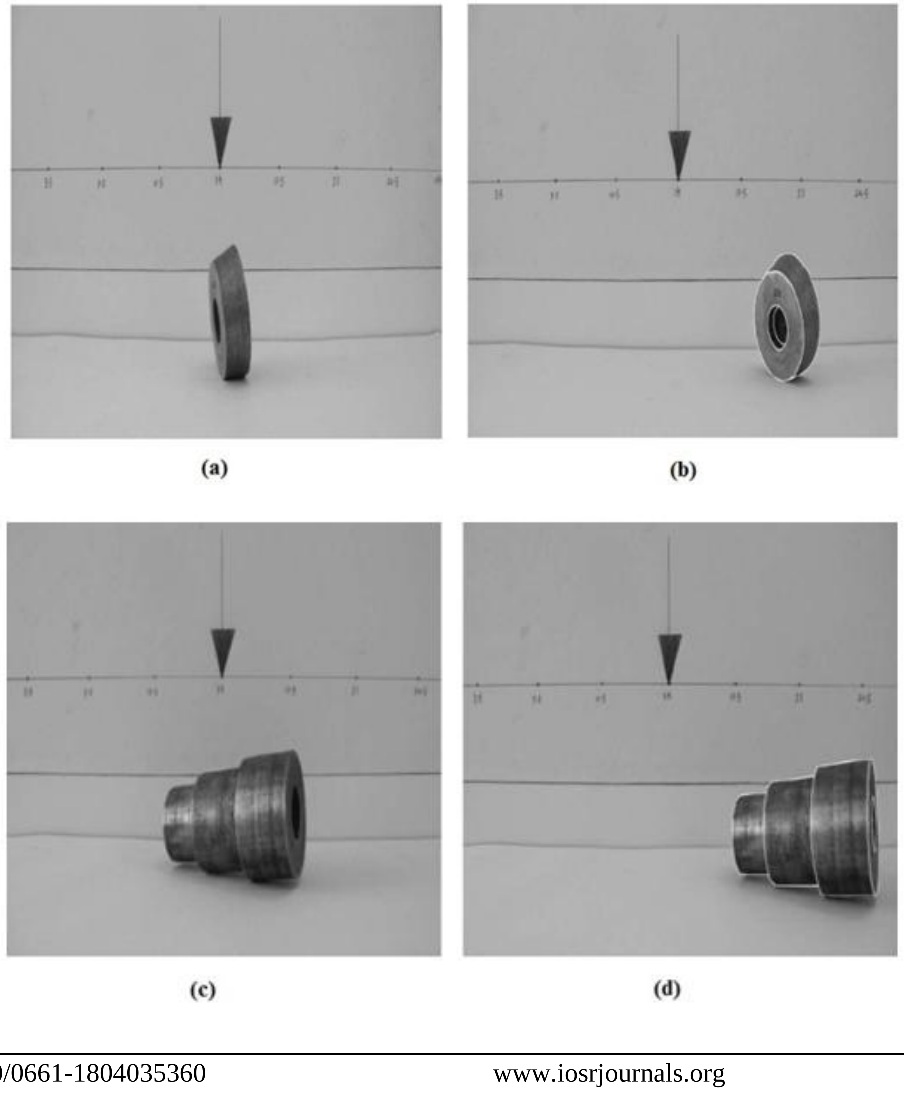
We have considered the number of points, N=28 in this experiment. We have added noise in a similar way as described in the last experiment. The number of correctly matched points continues to be 100% at 40 dB SNR with 30% of the vertices labeled as missing and 30% as spurious. At 20 dB SNR, when the percentage of missing and spurious vertices increases to 30%, the correctness of correspondences starts degrading. This is shown in Table 1.

Table 1: Effect of missing and spurious vertices on matching process

SNR	% of missing and spurious vertices			
	0%	10%	20%	30%
40 dB	100%	100%	100%	100%
20 dB	100%	100%	100%	85%

2. Pose Estimation using Real Data

In our work we have used QICAM FAST 1934 digital CCD camera. Its specification is compatible with IEEE 1394 Firewire digital CCD camera. In Fig. 7 we have shown three images of objects grabbed from different viewpoints. Fig. 7 (a), 7 (c), 7 (e) define three standard positions and orientations of the objects. In Fig. 7 (b), Fig. 7 (d) and Fig. 7 (f) we have shown translated and rotated version of objects mentioned before. We compute the translation and rotation of the objects using our method. Wireframes of the models of the objects are constructed using winged-edge data structure [15]. Now we align the wireframe of the models of the objects with the images using the knowledge of computed translation and rotation. The results are shown in Fig. 7 (b), Fig. 7 (d) and Fig. 7 (f).



(a)

(b)

(c)

(d)

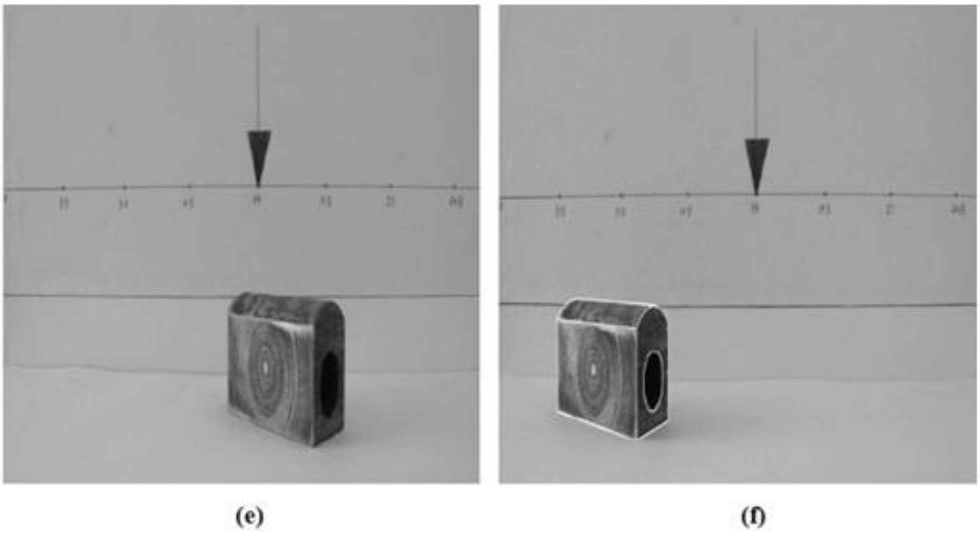


Figure 7: Alignment of wireframe models with the objects in real images.

V. Conclusion

Pose Estimation is one of the fundamental problems in Robotics and Computer Vision. The purpose of this work is to help to design a flexible automated visual inspection system. In this regard we have discussed a genetic algorithm based solution to pose estimation problem when the correspondences between object space and image space points are not known. In this method of solution, matching and pose estimation have been viewed as a single monolithic integrated process. We have proposed a composite chromosome structure by fusing two chromosome hypothesizes the process of correspondence. Most of the work on pose estimation found in the literature is from known correspondence. They avoid the overhead of matching while estimating the pose accurately. Our genetic algorithm based approach shows its robustness with respect to noise, and missing and spurious vertices and hence it is suitable for designing a *flexible* inspection system.

References

[1] T.G. H., S.N. Gottschlich, Ivis: An Integrated Volumetric Inspection System. Computer Vision and Image Understanding, 1995, 61(3): 430-444.

[2] N.T. S., A.K. Jain, A Survey of Automated Visual Inspection, Computer Vision and Image Understanding , 1995, 61(2): 231-262.

[3] D. Cyganski, J.A. Orr, Application of Tensor Theory to Object Recognition and Orientation Determination, IEEE Trans. on Pattern Analysis and Machine Intelligence, 1985, 662–673.

[4] P.David, D. DeMenthon, R. Duraiswami, H. Samet. Softposit: Simultaneous Pose and Correspondence Determination, Lecture Notes on Computer Science, 2002, 424–431.

[5] W.R. Franklin, V. Akman, Reconstructing Visible Regions from Visible Segments, 1986, BIT, 26: 430 – 441.

[6] D. DeMenthon, L.S. Davis, Model Based Object Pose in 25 liness of code, International Journal of Computer Vision, 1995, 15: 123-141.

[7] S. Hati, S. Sengupta, Pose Estimation in Automated Visual System using ANN, International Journal of Neural Systems, 2002, 12(6): 483-496.

[8] G. Schweighofer, A. pinz, Robust Pose Estimation from a Planar Target, IEEE Trans. on Pattern Analysis and Machine Intelligence, 2006, 28(12): 2024-2030.

[9] H. Lu, X. Shao, Y. Xiao, Pose Estimation with Segmentation Consistency, IEEE Trans. on Image Processing, 2013, 22(10): 4040-4048.

[10] A. Assa, F. Janabi-Sharifi, A Robust Vision – Based Sensor Fusion Approach Real Time Pose Estimation, IEEE Trans. on Cybernetics, 2014, 44(2): 217 – 227.

[11] C.M. Fonseca, P.J. Fleming, Genetic Algorithm for Multiobjective Optimization: Formulation, Discussion and Generalization, In Genetic Algorithms: Proceeding of the Fifth International Conference San Mateo, CA:Morgan Kaufmann, 1993, 416–423 .

[12] R.M. Haralick, Computer and Robot Vision, Addision-Wesley Publishing Company, 1993.

[13] R.C. Gonzalez, R.E. Woods, Digital Image Processing, 1st edition, Addision-Wesley Publishing Company, 1993.

[14] M¨antyl¨a, An Introduction to Solid Modeling, Computer Science Press, 1988.

[15] K. Weiler, Edge Based Data Structure for Solid Modeling in Curved Surface Environments, IEEE Computer Graphics and Applications, 1985, 28(12): 21–40.

[16] R.M. Haralick, H. Joo, C. Lee, X. Zhuang, V. Vaidya, M.B. Kim, Pose Estimation from Corresponding Data, IEEE Trans on SMC, 1989, 19(6): 1426 – 1446.