MVA '92 IAPR Workshop on Machine Vision Applications Dec. 7-9,1992, Tokyo

CORRESPONDENCE OF SURFACES IN A SEQUENCE OF RANGE IMAGES FOR MOTION ESTIMATION AND TRACKING

Bikash Sabata and J. K. Aggarwal

Computer and Vision Research Center
The University of Texas at Austin
Austin, Texas 78712, USA

ABSTRACT

The key issue in motion estimation and tracking an object over a sequence of images is establishing correspondence between the features of the object in the different images of the sequence. For range image sequences, this problem translates into finding a match between the surface segments in a pair of range images of the scene. This paper considers the problem of establishing correspondences between surfaces in a sequence of range images. We present a novel procedure for finding correspondence and show the results on real range image sequences. A graph search procedure forms the basis for the algorithm that computes the correspondence between surfaces. The solution uses geometrical and topological information derived from the scenes to direct the search procedure. Fundamental to our strategy to match features over a sequence of range images is a hypergraph representation of the scenes. Two scenes are modeled as hypergraphs and the hyperedges are matched using a sub-graph isomorphism algorithm. The hierarchical representation of hypergraphs not only reduces the search space significantly, but also facilitates the encoding of the topological and geometrical information. Further, we present a sub-hypergraph isomorphism procedure to establish the correspondences between the surface patches and demonstrate the algorithm on different types of real range image sequences. We present results that show that the algorithm is robust and performs well in presence of occlusions and incorrect segmentations.

INTRODUCTION

The key issue in motion estimation and tracking an object over a sequence of images is establishing correspondence between the features of the object in the different images of the sequence. In this paper we deal with the tracking of objects in a sequence of range images to estimate the motion of the camera (range sensor) in the environment. Range images sense the surface of the objects, so it is natural to use surface segments as the features of interest; this translates the tracking of objects into finding a match between the surface segments in a pair of range images of the scene. This paper considers the finding of correspondences between surfaces in a sequence of range images. Finding correspondence or a match between features is not isolated to object tracking, but is also central to other computer vision tasks including navigation, object recognition, target tracking, and map building. We present a novel procedure for establishing correspondence and show the results on real range image sequences.

A graph search procedure forms the basis for the algorithm that computes the correspondence between surfaces. The solution uses geometrical and topological information derived from the scenes to direct the search procedure. In general, the input to the matching algorithm is the output from a segmentation algorithm that partitions the image into surface segments. The performance of the matching depends greatly on the results of the segmentation algorithms. Incorrect segmentation causes poor estimation of the surface parameters and affects the performance of the matching algorithm. We address this issue and obtain a solution that is robust and able to handle occlusions of surfaces, noise in data, and incorrect segmentation from a segmentation algorithm. In the present implementation, we assume that the images have planar, cylindrical and conical surfaces; however, the procedure is general enough to be extended to other surface classes.

The question of finding correspondences between features has been studied extensively (see [1, 3, 4, 5]) but, most of these approaches deal with matching a scene to a model of the object. The fundamental difference between model-to-scene matching and scene-to-scene matching is that in the former, the model description of the object is complete, and to that we match the incomplete description of the object obtained from the scene. However, in the case of scene-to-scene matching, both descriptions of the object are incomplete and we must find a match between two incomplete descriptions. By incomplete, we mean that all the features are not present in the description of the object because of occlusions and sensor errors. This difference makes it impossible to use the strategies obtained for object recognition in the domain of object tracking; new strategies based on the constraints of the problem have to be designed.

Fundamental to our strategy to match features over a sequence of range images is a hypergraph representation of the scenes. The two scenes are modeled as hypergraphs and the hyperedges are matched using a sub-graph isomorphism algorithm. To reduce the complexity of the matching task, heuristics derived from the topological and the geometrical information available from the scene are used to direct the search. The hierarchical representation of hypergraphs not only reduces the search space significantly, but also facilitates the encoding of the topological and geometrical information. Hyperedges are formed by grouping the surface features, which reduces the search space. Using a priori knowledge arising out of the physical constraints of laser scanning, a fast matching algorithm is designed.

HYPERGRAPH REPRESENTATION

Hypergraphs are generalizations of graphs. The edge is generalized as a hyperedge, where a set of vertices forms the hyperedge, instead of just two vertices forming the edge. The group of vertices forming the hyperedge may share some common property. Hypergraphs have been used earlier in vision and robotics applications [11, 12], but have not found widespread usefulness. We present a new definition of the hyperedge and a novel method for constructing the hypergraphs that makes it a powerful tool for vision applications.

Attributed hypergraphs are a concise way of representing objects such that both quantitative and qualitative information are encoded in the representation. Formally;

Definition 1 *The <u>Hypergraph</u> [2] is defined as an ordered pair $H = (X, E)$ where $X = \{x_1, x_2, \cdots, x_n\}$ is a finite set of attributed vertices and $E = \{e_1, e_2, \cdots, e_m\}$ are the <u>hyperedges</u> of the hypergraph. The set E is a family of subsets of X (i.e. each e_i is a subset of X) such that*

1. $e_i \neq \emptyset, i = 1, \cdots, m$
2. $\bigcup_{i=1}^{m} e_i = X$.

A graph is a hypergraph whose hyperedges have cardinality of two. To each hyperedge, we associate an attribute set that maps the vertices (belonging to the hyperedge) to an attributed graph.

Each surface patch in the range image forms an attributed vertex. The attribute values are the surface property values. For each pair of surfaces that are connected, an attributed arc is formed. The attributes of the arc describe the interfacing edge and the relative geometrical information between the two surfaces. Groups of the attributed vertices (surface patches) form an hyperedge, and with each hyperedge we associate an attributed graph that describes the topology of the component attributed vertices (surface patches).

The set of vertices that form the hyperedge should represent a topologically significant feature in the graph so that the matching task is guided by the topology of the scene. Cliques in the graph are significant features that are rich in information. Physically, the cliques represent groups of surfaces that are adjacent to each other. Since a clique provides a larger attribute set and many geometrical properties, the probability of a false positive match (between two cliques) is reduced significantly. Each clique forms a hyperedge in the hypergraph and the attributed graph describing the clique is the associated attribute of the hyperedge. Figure 1 illustrates the formation of a hypergraph from a scene.

The complexity of computing the cliques in a graph is exponential, so the formation the hypergraph will be exponential. However, the physics of the range imaging process restricts the size of the cliques in the scenes that we observe. It can be shown that the size of the clique is restricted to be four [8]. Once the upper bound on the size of the cliques is known, the complexity of computing the cliques becomes O(n).

THE MATCHING PROCEDURE

This section presents the matching procedure used to derive the surface correspondences in a sequence of range images. The heart of the procedure is a *directed tree search*

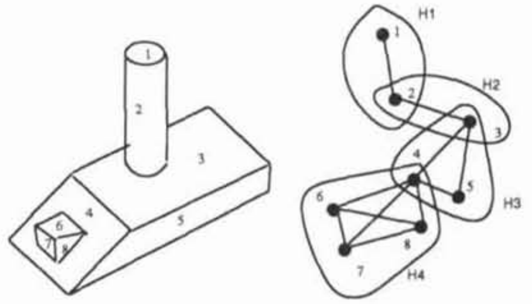

Figure 1: *An object and its corresponding Hypergraph representation*

algorithm that tests various hypotheses and rejects the impossible ones. Finally the interpretation that gives the largest match is selected as the solution. Constrained tree search algorithms have been used in many applications [4, 5, 7]. Data pairings are formed by a depth first search of an *interpretation tree*. Each node of the tree represents a possible pairing. The first data (surface patch) is taken from the first scene and paired with each of the data in the second scene. These form the nodes in the first level of the tree. To account for missing surface segments due to occlusions, the data is also paired with a *wild card* $\star$. Subsequent levels of the tree correspond to pairings of other vertices. Each branch of the tree represents a partial matching of the scenes. The constraints are used to prune the search tree and thus reduce the search space.

We present a variation to the constrained tree search, in which the search is directed based on the current hypothesis. The directed search, coupled with the termination conditions, further reduces the search space. The key idea is to use the topological constraints of the scene to determine the next most likely match, and to accept or reject the matches based on the geometrical constraints.

The features used in the matching process are surface segments. We assume that a segmentation algorithm [9, 10] segments the range image into surface patches and the surface parameters are computed. The interfacing edge between the surface patches are detected and their properties are computed. The properties of the edge segments used are (1) the edge type (straight line or curved), (2) the edge length, and (3) the depth discontinuity. The depth discontinuity across the edge implies that one surface may be occluding (partially or completely) another surface. The information about occlusion is also incorporated in the attribute list of the surface patches.

The constraints used are similar to the *unary* and *binary* constraints developed by Grimson and Lozano-Perez [6]. The only unary constraint we use is the surface type classification (planar, cylindrical, conical, etc.). Other properties, used in model based object recognition, such as area, perimeter, compactness, etc., are very sensitive to occlusion, and since occlusion may occur in either of the range images, these properties cannot be used as constraints. The binary constraints describe the relative properties between pairs of surface segments. The properties we use are (1) connectivity, (2) the angle between the surface patches, (3) the range of distances between the two surface patches, (4) the range of the components of the vector spanning the two surface patches, and (5) the properties of the interfacing edge. Each constraint is measured and tested against a predetermined threshold. For surface segments that have an occluding edge, the neighbors information is not complete (a neighbor may be hidden) and the connectivity information may be inaccurate. Therefore, for such cases only a *weak* match is hypothesized which is subject to conformation or rejection based on further evidence.

Matching between the two hypergraphs representing the scenes is achieved by computing the match between the component hyperedges. A match between the two hyperedges is hypothesized. The two hyperedges are matched by matching the attributed graphs representing the hyperedges. An order of vertices is established at each stage of the match. The order determines the branches taken in the search tree. The order is determined by listing the hyperedges connected to the vertices that have been matched in current hypothesis. The matching procedure starts by selecting the largest hyperedge H_1 and H_1' in the two scenes. The vertex with the largest degree is selected as the first node n_1 and it is matched with the corresponding vertex in the second hyperedge. The unary and the binary constraints are checked to evaluate the match between the hyperedges. Once the hyperedge-match has been established, the second set of hyperedges are selected. The next hyperedge H_2 is the hyperedge connected to H_1 at n_1. A match for each of the hyperedges connected to H_1 at n_1 is found. The search then proceeds to find matches of hyperedges connected to H_1 at other vertices belonging to H_1. The procedure goes down the list of all the vertices in the hypergraphs in the order evaluated earlier. Once a match for a hyperedge is found that hyperedge is marked as *matched*. The marked hyperedges are not considered in the future hypotheses.

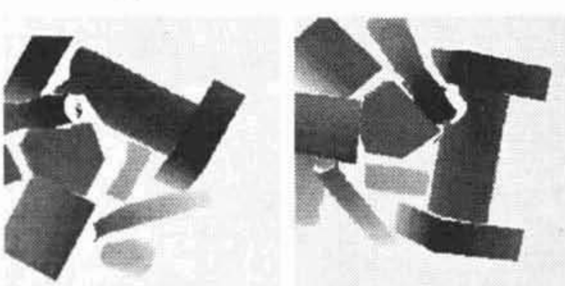

Figure 2: *The depth maps of a sequence of range images.*

Termination of the matching procedure occurs if the fraction of surface patches matched exceeds a threshold. Once a match has been determined (i.e., the search procedure has reached the leaf node of the tree), the number of positive pairings (i.e., non-wild card pairings) is computed. If this number is less than the threshold fraction then the procedure backtracks and searches other branches. At every stage the best possible match is compared with the current best match. If the best possible match is smaller than the current match, then the search along that branch is abandoned and the next branch is investigated.

RESULTS

In this section we present an example of a range image sequence and describe how the matching algorithm computes the surface correspondences. The algorithm was tried successfully on different types of range image sequences.

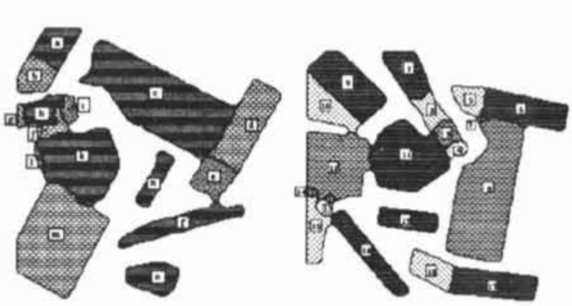

Figure 3: *The segmented range images.*

Figures 2-4 illustrate the algorithm on an example. Figure 2 shows the depth maps of two frames in the sequence of range images. The scenes consist of a jumble of different kinds of objects. The camera is moved to obtain the second frame of the sequence. The segmentation algorithm of [10] was applied on the images and the results input to the matching algorithm. The segmented results are shown in figure 3. The first step of the algorithm generates the attributed graph of the scene and computes the cliques in the graph. Each clique forms a hyperedge in the generated hypergraph. The hypergraphs generated are shown in the figure 4. For each hyperedge the component vertices form an attributed graph. In the figure the arcs of the attributed graph are shown in the hyperedges. Using the properties of the edge, interfacing two surface segments, it is determined if two surfaces are connected. If there exists an occluding edge between two surfaces then the arc in the attributed graph is *weak* (shown in the figure 4 with dotted lines). A match based on a *weak* arc is a *weak* match and further evidence is required to confirm the hypothesis.

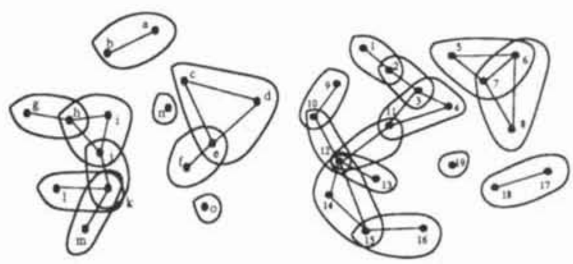

Figure 4: *The generated hypergraphs of the range images.*

The first hyperedge pair hypothesized to match is $\{h, i, j\}$ in the first scene matches $\{3, 4, 11\}$. The vertex with the highest degree h is considered as the first vertex. The unary constraints leave only one option i.e., $(h, 3)$ as the first node in the interpretation tree. However, the next two vertices i and j do not match any vertex so they are matched with the wild card $\star$. Note that in the final match that is obtained the pairing $(h, 3)$ is an incorrect pairing. The algorithm backtracks and finds the correct match even though we start with an incorrect match. We present the first few steps to illustrate how the algorithm works. The second hyperedge considered for match is $\{h, g\}$ because it is connected to the first hyperedge at h. Since the current hypothesis is $(h, 3)$, the next hyperedge match considered is between $\{h, g\}$ and $\{3, 2\}$. The unary constraints are satisfied between the pair $(g, 2)$ so the binary constraints of angle, distance and the spanning vector are tested. All the constraints are satisfied so the match pair is accepted in the current hypothesis. The next hyperedge considered now is $\{j, k\}$ as it is connected to the first hyperedge. The match between $\{j, k\}$ and $\{11, 12\}$ is tried and the pairing $(k, 11)$ satisfies all the constraints, but the connectivity is not satisfied (k is not connected to h while 11 is connected to 3). At this point we use the fact that the arc between 3 and 11 is a *weak* one so it can be broken and all the constraints are satisfied.

The procedure continues till a complete match (i.e., all the vertices are accounted for) is obtained. The match size is evaluated and if a better match can be obtained, the procedure backtracks to improve the results. The final matching results are:

I	a	b	c	d	e	f	g	h	i	j	k
II	6	5	8	17	18	16	1	2	3	⋆	11
I	l	m	n	o	⋆	⋆	⋆	⋆	⋆	⋆	
II	⋆	12	19	15	4	7	9	10	13	14	

It may be observed that in the example shown there are many errors in segmentation (for eg. surfaces j, 7, 14, 13, etc.) and there are surfaces that get occluded in one of the scenes (for eg. 4 and l); notwithstanding, the algorithm performs well and the correspondences are evaluated.

CONCLUSION

Computing motion and tracking an object over a sequence of range images involves establishing correspondence between the features of the object in different images in the sequence. The question of finding correspondence in a sequence of range images is very different from finding correspondence between a model and an object description. The fundamental difference lies in the fact that the model description of the object is complete, while in case of a sequence of range images, both descriptions of the scene are incomplete. The lack of information forces one to impose only weak constraints and allow for larger tolerances.

We presented a new framework and procedure to compute the correspondences between surface segments in a sequence of range images. Fundamental to our framework is the hypergraph representation of the range images. The hierarchical representation of hypergraphs not only reduces the search space significantly, but also facilitates the encoding of the topological and geometrical information. In addition to the topological and geometrical information obtained from the scene we also use a priori knowledge of the scene obtained from the physics of the laser scanning process used to produce the range images. Each piece of information used reduces the complexity of the matching procedure by pruning the search space. The solution is robust and accounts for errors in segmentation, occlusions of surfaces, and noise in the data. By using the topological information to guide the search procedure, the average case complexity of the algorithm is reduced significantly.

Acknowledgments

We would like to thank Debi Paxton for proof reading the paper. This research was supported in part by Army Research Office, Contract *DAAL-03-91-G-0050* and Air Force Office of Sponsored Research, Contract *F49620-92-C-0027*.

References

[1] F. Arman and J.K. Aggarwal. Model-based object recognition in dense range images - a review. *To Appear in ACM Computing Surveys*, 1992.

[2] C. Berge. *Graphs and Hypergraphs*. North-Holland Publishing Company, Amsterdam, 1983.

[3] W.E.L. Grimson. The combinatorics of object recognition in cluttered environments using constrained search. *Artificial Intelligence*, 44:121–165, 1990.

[4] W.E.L. Grimson. The combinatorics of heuristic search termination for object recognition in cluttered environments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(9):920–935, Sept. 1991.

[5] W.E.L. Grimson and D.P. Huttenlocher. On the verification of hypothesized matches in model-based recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(12):1201–1213, Dec. 1991.

[6] W.E.L. Grimson and T. Lozano-Perez. Localizing overlapping parts by searching the interpretation tree. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9(4):469–482, 1984.

[7] V. Kumar. Algorithms for constraint satisfaction problems: A survey. *AI Magazine*, pages 32 – 44, Spring 1992.

[8] B. Sabata and J.K. Aggarwal. Correspondence of surfaces in a sequence of range images using hypergraphs. Submitted for Publication.

[9] B. Sabata, F. Arman, and J. K. Aggarwal. Segmentation of 3-d range images using pyramid data structures. Accepted for publication in Computer Vision, Graphics, and Image Processing: Image Understanding.

[10] B. Sabata, F. Arman, and J. K. Aggarwal. Segmentation of range images using pyramidal data structures. In *Proceedings of International Conference in Computer Vision*, pages 662–666, Osaka, Japan, December 1990.

[11] A.K.C Wong. Knowledge representation for robot vision and path planning using attributed graphs and hypergraphs. In A.K.C. Wong and A. Pugh, editors, *Machine intelligence and knowledge engineering for robotic applications*, pages 113–143. Springer Verlag, Berlin Heidelberg, 1987.

[12] A.K.C. Wong, S.W. Lu, and M. Rioux. Recognition and shape synthesis of 3-d objects based on attributed hypergraphs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(3):279 – 290, March 1989.