Recognition of Welding Panels in Subassembly Process of Shipbuilding

Kyoung Sig Roh*,Mun Ho Jung*,Jin Oh Kim**,and In So Kweon^

*Ph.D. Student, ^Assistant Professor
Dept. of Automation and Design Eng., KAIST
207-43, Cheongryangri-Dong, Dongdaemun-Ku, Seoul, Korea
**Samsung Elec. Co.,LTD.,Suwon,Kyungki-Do,Korea
Internet Address: kweon@ design.kaist.ac.kr

ABSTRACT

We present a model-based vision system to recognize planar objects which can be represented by a set of lines. Our system uses the projective invariants of five lines. However, a set of five lines can produce up to 120 different invariant values. As the number of models in a model database becomes larger, the size of search space to find a corresponding model in the database may increase exponentially. In order to solve this problems, we introduce a line convex hull(LCH) and an indexing logic filter(ILF). The line convex hull classifies a set of five lines into one of twenty-one different types of convex hulls and also provides an unique ordering to five lines. Using the types of line convex hulls, ILF computes an integer indexing value to represent a set of six lines.

By combining the LCH and the ILF, the proposed scheme greatly improves search speed to find candidate model features in the model database that match scene features. We have performed a series of experiments on real images of the fifteen synthesized models of welding panels. The system has successfully recognized models in the database as well as the corresponding features.

KEYWORDS : recognition, indexing, invariant, line

1. INTRODUCTION

Most model-based recognition methods precompile the desciptions of objects using some prominent geometric features such as points, lines, and surfaces of objects and recognize the best candidate model by matching candidate model features to the extracted scene features. In this paper, we present a model-based recognition system based on line features. We assume that objects can be modeled as a collection of lines. Many of the previous object recognition systems have used line feature to represent and recognize objects. Ayache and Faugeras[1] proposed a 2D object recognition method called HYPER. In the HYPER system, both model and scene are represented in the same way by approximating the boundary with polygons and the geometric relations such as the angle between lines were used for recognition. Lowe[2] emphasized the importance of viewpoint consistency constraint and demonstrated a 3D object recognition algorithm, which used line features.

Recently, the projective invariant-based approaches have been proposed to consider a scene with a strong perspective effect. It is possible to index for recognition because the projective invariant value is constant under projective transformation. Geometric hashing method introduced by Schwartz et.al.[3] has been proved as a very efficient recognition method[4][6]. Tsai[5] proposed a probabilistic approach to the geometric hashing method using line invariant. Since most of the traditional indexing methods have considered all possible permutations, the index table and the generated hypotheses are very large.

In this paper, we also use line invariant to find model features in the model base which have consistent matches with scene features. Specifically, we propose two methods to reduce the size of an indexing table. One is "Line Convex Hull (LCH)" and the other is "Indexing Logic Filter(ILF)". The line convex hull is a method to obtain a permutation invariant for five lines, thus we can obtain one invariant by using the LCH. And ILF is a method for fast indexing by assigning an integer to a set of six LCHs.

This paper is organized as follows. In section 2, we present the concept of the LCH and ILF. In section 3, the object recognition algorithm is described. In section 4, experimental results for real images of the fifteen synthetic models of welding panels are given.

2. LINE CONVEX HULL and INDEXING LOGIC FILTER

2.1 Five Line Invariant

Using 5 lines, $L_1 \sim L_5$, in a model and the corresponding 5 lines, $l_1 \sim l_5$, on the image of the scene, we can define the following 2-D invariant:

$$\mathbf{I} = \frac{\det(l_5\ l_1\ l_4)\ \det(l_5\ l_2\ l_3)}{\det(l_5\ l_1\ l_3)\ \det(l_5\ l_2\ l_4)} = \frac{\det(L_5\ L_1\ L_4)\ \det(L_5\ L_2\ L_3)}{\det(L_5\ L_1\ L_3)\ \det(L_5\ L_2\ L_4)} \tag{1}$$

where $\det(l_i l_j l_k) = \begin{vmatrix} a_i & a_j & a_k \\ b_i & b_j & b_k \\ c_i & c_j & c_k \end{vmatrix}$ and $a = (a_i\ b_i\ c_i)$ represents the parameter vector of line l_i.

2.2. Line Convex Hull and Permutation Invariant

A line invariant defined in Eq.(1) is not invariant for permutation of five lines. If all these permutations are included in an index table for recognizing objects, the size of search space to find a corresponding model in the database may be exponentially increasing. In this paper, we propose a line convex hull of a set of five lines, that is, a permutation invariant. Line convex hulls classify a set of five lines to twenty-one types using the relative positions of intersection points between lines(see Fig. 1 for the twenty-one types of line convex hulls). A detailed algorithm for constructing line convex hulls is as follows.

Given all the intersection points between lines extracted, we

1. choose a line as a reference,
2. compare the number of intersection points on each side of the reference line,
3. remove all the points on the side with a smaller number of intersection points;
4. if each side has the same number of intersection points;
   (a) remove the intersection points on the side which are not collinear;
   (b) if points on both sides are not collinear, remove the points on the side having a concave shape for boundary lines;
5. construct a line convex hull from the remaining intersection points.

The twenty-one types are determined from the relative position of a LCH and the arrangements of removed points. These types are shown in Fig. 1, where blank circles indicate the remaining points to form a LCH.

To obtain a single invariant for five lines, we must give an unique line number to each of five lines. Thus, we determine the first line using the type of a LCH and the configuration of intersection points. The first line is determined as follows:

1. For a line convex hull consisting of four intersection points, the first line is
   a. a line with four removed intersection points (Type 0,1,2,3,6,7,10,11, 14);
   b. a non-parallel line (Type 17,18);
   c. a line which has two and three intersection points in each side of the first line (Type8, 9).
2. For a line convex hull consisting of five intersection points, the first line is
   a. a non-parallel line (Type16);
   b. a line which has the same number of intersection points in its left-hand and right-hand side(Type 5, 12, 13).
   c. a line which does not have intersection points in its left-hand or right-hand side (Type 4, 15).
3. Any line can be the first line if the configuration of lines is symmetric with respect to any line (Type 19);
4. For a line convex hull consisting of three intersection points, a line which intersects two parallel lines through two mid-points of intersection points on two parallel lines becomes the first line (Type 20).

Once the first line is determined, the ordering of the remaining lines can be easily determined. First, a LCH is assumed to be in the left-hand side of the first line. The order of the remaining lines are then determined by the order that intersect the first line in this direction.

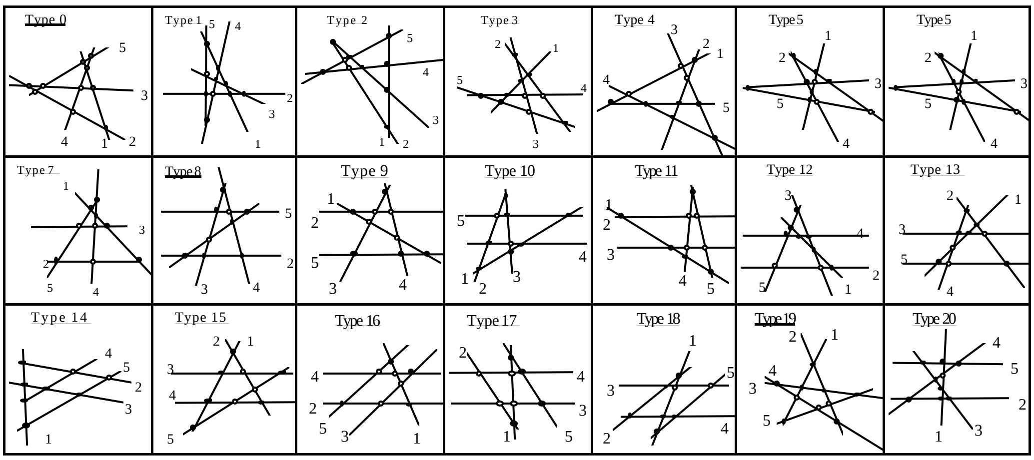


Fig.1 Line Convex Hulls and the configurations of lines.

2.3 Indexing by ILF(Indexing Logic Filter)

In this section, we present an indexing logic filter (ILF) which assigns an integer indexing value for a set of six lines. Conventional logical filtering has been applied to binary images to produce an integer value corresponding to 512 (2^9) types in a 3x3 region of the image as shown in Fig.2. For example, if 1 and 7 are selected in this 3x3 region, then the corresponding value of the logic filter becomes 130($=2^1+2^7$).

We extend this simple concept to the indexing logic filter which indexes line features efficiently. The ILF is defined as follows:

> Given a set of types with N elements, **T**={0, 1, 2,..., N-1}, we assume that **p** elements from the set **T** are selected to form a subset of types. Then, an

indexing logic value for representing the subset is defined by:

$$ILV = \sum_{k=1}^{m} a_{i_k} n^{i_k} \quad , \quad \sum_{k=1}^{m} a_{i_k} \leq p \tag{2}$$

where a_{i_k} is the number of selections of i_k which represents the type of an element of **T** and m indicates the total number of different types selected.

Note that indexing logic values are unique if and only if an integer n satisfies $n > p$. Now, it is applied to form an indexing logic value for a set of six lines. Each set of five lines defines a line convex hull as one of 21 types described in the previous section. Therefore, we can set the number of elements to N = 21. Since we obtain six line convex hulls from a set of six lines, we can also set n and p to 7 and 6, respectively. For example, let an ordered set of line convex be **T** = {3 0 0 6 8 S}, where S indicates a singular type. From Eq. (2), we obtain an integer number such that ILV = 5,882,795 = 1 x 7^8 + 1 x 7^6 + 1 x 7^3 + 2 x 7^0 for this set of line convex hulls.

In addition, the index selectivity power of ILV can be further extended by a permutation of p elements. Among all the possible permutations, the set **T**, for example, is arranged at 198th (= 5!+5!/2!+4!/2!+3!) if we assume a negative number for S and an arrangement starting from a smaller number.

8	7	6	5	4	3	2	1	0

Fig.2. An example of logic filter

3. OBJECT RECOGNITION ALGORITHM

In this section, we present a model-based object recognition algorithm. First, we explain a structure of a model database that will used for indexing. Then we describe details of the recognition algorithm.

3.1 Database Structure

For each model and for every feasible set of five lines, we
(i) compute an invariant and a line convex hull of the five lines;
(ii) by adding a new line to the five lines, form a set of six lines and compute an indexing logic value and an arrangement of six line types.

Fig.3 shows an example of the data structure of the model base. The model base consists of two indexing keys: one includes ILV and an ordering information, and the other includes invariant values of each set of five lines, their variances, a line type and an arrangement of line types. For all the given models, this data structure is created by off-line.

Indexing Key 1

ILV	Order
5883137	129

Indexing Key 2

Invariant	**0.3893**	**0.4345**	**0.8912**	**0.5681**	**0.5444**	**0.9925**
Variance	**0.0032**	**0.0033**	**0.0059**	**0.0042**	**0.0043**	**0.0071**
Line ID	**10**	**5**	**0**	**4**	**6**	**9**
Order		**04312**	**12340**	**41203**	**34021**	**04321**

Fig.3 Database Structure

3.2 Object Recognition Algorithm

For a set of six lines extracted in an input image, it computes an indexing logic value and an order as indexing keys to retrieve model features from the model base to match scene features. If there are multiple matching candidates, one correct match with model features is obtained by the verification stage. Once a correct model which matches the selected line features in the scene found, all the remaining lines in the scene are matched with model features by the verification stage.

Verification starts with multiple matching candidates which have the correct correspondence of model and six scene features. Verification eliminates many false matches by checking the correspondence of a nearby predicted model feature and a newly added scene line feature. Correspondence is confirmed when two indexing keys for a new line successfully retrieve a model feature with a correct invariant value. This procedure is repeated until a single match is obtained. In addition, verification of the entire line feature is then performed to reduce the false alarm rate.

4. EXPERIMENTS

We have done a series of experiments using real images of fifteen models which are modeled by line features. Fifteen models are designed to synthesize welding panels as shown in Fig. 4 wehre the lines inside objects denote welding seams.

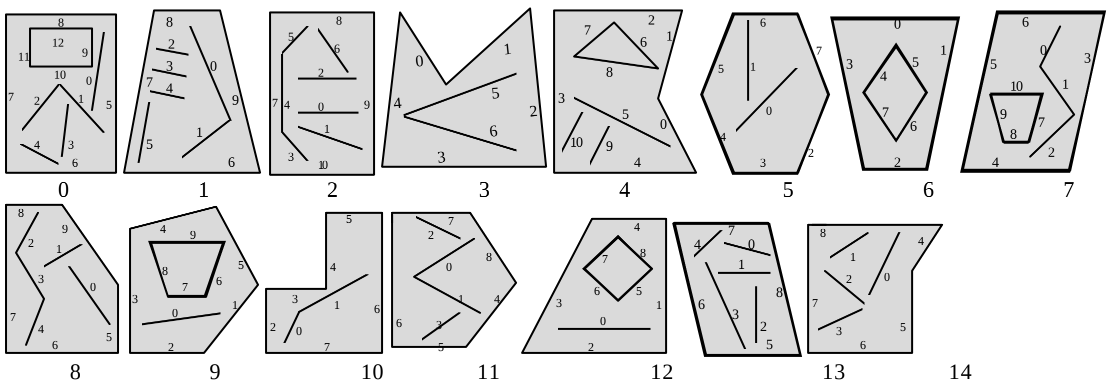
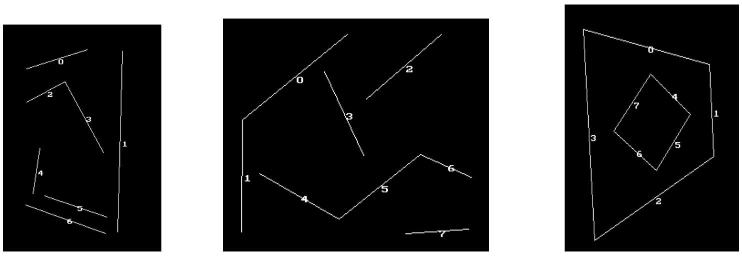


Fig. 4. The fifteen synthetic models of welding panels used for our experiments.

Fig. 5 shows the result of edge detection using FDG edge detector for three input images. To create a strong perspective effect intentionally, we set a distance between camera and an object to be approximately 50 cm and an angle between the optical axis and the surface normal of the object plane as 60 degrees. Table 1 shows the results of the recognition and the elapsed time for each step. Total elapsed time is within 1.14 sec on a PC with c30-based vision board for the fifteen models. Table 2 shows the indexing of the object and the corresponding model.



(a) Input image 1 (b) Input image 2 (c) Input image 3

Fig. 5. The results of line extraction for each input image

Table 1. The result of the recognition.

Initial Matching	INPUT 1(1)	INPUT 2(8)	INPUT 3(5)
Input Feature	0 1 2 3 4 5	0 1 2 3 4 6	0 1 2 3 4 5
Matched Feature	9 6 0 1 4 5	9 8 0 1 2 4	1 2 3 0 6 7
Elapsed Time	0.66 sec	0.98 sec	0.93 sec
Propagating			
Input Feature	0 1 2 3 4 5 6	0 1 2 3 4 5 6 7	0 1 2 3 4 5 6 7
Matched Feature	9 8 0 1 4 5 7	9 8 0 1 2 3 4 7	1 2 3 0 6 7 4 5
Elapsed Time	0.05 sec	0.16 sec	0.11 sec

Table2. The correspondence of object and model for INPUT 2.

	MODEL	OBJECT
Line ID	1 2 4 0 8 9	3 4 6 2 1 0
Convex Hull Type	2 12 10 15 5 10	2 12 10 15 5 10
ILV - Order	4761967747691000-43	4761967747691000-43
Invariants	0.0988, 0.7874,0.4333, 0.4771,0.7024,0.1861.	0.0565,0.7408,0.4744 0.4804,0.75 47,0.1109

5. CONCLUSIONS and FUTURE WORKS

In this paper, we use the invariant of five lines and their geometric configuration to recognize planar objects under the projective transformation. We introduce new concepts, such as the Line Convex Hull (LCH) and the Indexing Logic Filter (ILF), for an indexing method to reduce the searching space. Although the search complexity order remains to $O(n^6)$, our algorithm has reduced the searching space remarkably.

In the real experiments, our system has successfully recognized fifteen models with a much lower computational cost compared with some previous methods.

Work is currently underway to apply the algorithm to real welding panels. We will extend these concepts to three dimensional object recognition which needs to construct a systematic and efficient model base.

6. REFERENCE

[1] N.Ayache and O.D.Faugeras,"A New Approach for the Recognition and Positioning of Two-Dimensional Objects", IEEE Trans. PAMI, Vol.8, No.1 January 1986
[2] D.G.Lowe,"Three-Dimensional Object Recognition from Single Two-Dimensional Images", Artificial Intelligence, 31,pp355-395, 1987.
[3] Y.Lamdan, J.T.Schwartz ,and H.J.Wolfson,"Object Recognition by Affine Invariant Matching",In Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition, pp335-334,1988.
[4] C.A.Rothwell,A.Zisserman,D.A.Forsyth and J.L.Mundy, "Fast Recognition Using Algebraic Invariants", in Book of Geometric Invariance in Computer Vision, ed. J.L.Mundy, A.Zisserma, pp.398 -407, MIT Press,1992.
[5] F.C.D.Tsai," Geometric Hashing with Line Features",Pattern Recognition", Vol.27, No.3 , pp377-389, 1994.
[6] Haim J. Wolfson, "Model-Based Object Recognition by Geometric Hashing", Proc. 1st European Conf. on Computer Vision,1990.