Detect and Represent Lines Based on a Set of 3-Pixel Elementary Units

WEI Hui WANG Zengjin LIU Bin

Department of Computer Science, Lab of Algorithm for Cognitive Model, Fudan University, Shanghai 200433, P. R. China

Summary

A key task of digital computer image understanding is to organize image pixels according to semantic structure, and one of the most important clues to this task are the edges and profiles of the objects. In most cases edges can be represented as lines, therefore detecting line elements is an important procedure of image understanding. However, edge detection algorithms only provide a set of discrete pixels instead of direct descriptions of lines. In this paper, we propose a new method of line detection and line description based on combinations of multiple three-pixel basic patterns. The new algorithm connects those adjacent pixels that are geometrical likely to form line segments, thus it remarkably simplify the description of an image by using line segments instead of pixels. The proposed method consists of three major components: first, patterns of elementary unit that compose saw-toothed line segment in grid mode are defined. Second, the rule of combination of different basic patterns is provided, and proved to be feasible in experiments. Third, a hierarchical and parallel network computing model to obtain the basic patterns of lines and a clustering algorithm based on basic patterns. Experimental results show that this new method can considerably improve performances in computation time, memory requirement and detection accuracy compared with other classical line detection algorithms. Another advantage of this kind of geometric-character-based clustering algorithm is that the output of this method can be taken as the input of subsequent object for recognition procedures directly. Moreover, no extra manipulations with higher algorithm complexity, such as searching line end points, filtering false lines or segmenting collinear line segments are needed.

Key words:

Line detection, Image understanding

1. Introduction

The edges of objects in an image are one of the most important features for scene pattern recognition. If we analyze the components of the boundary lines, we find that the line segments with various lengths and directions are the basic elements to describe edges. As line segments are the most basic elements or units for object recognition, therefore detecting line segments is a key step for pattern recognition.

Pixel array is a direct method to describe a scene image. This method can only describe the low-level color and brightness features, other than point-point or point-line combination, although it is simple, intuitive and flexible. So the ultimate goal of detecting lines is to describe the image in a more meaningful manner. Pixels are set cluster property, after lines of image are formed into vectors. We can compose and separate each individual features, and we can describe an image at a higher level. From a fundamental pattern representation viewpoint, this method of combining feature is extremely different from other methods of popular statistical pattern recognition, resulting in quit difference of their recognition algorithms. Pattern representation method based statistic has “over” and “lacking” disadvantages, so statistic based vision pattern recognition approach is constrained by forms of pattern representation at explaining, convergence and generalization of learning algorithm. However, semantic description based pattern recognition method has a satisfying result in terms of complexity of representation dimensions and correctness of representation.

A classical algorithm of detecting straight line is the Hough Transform[1](HT), which was proposed by Paul Hough in 1962 and patented by the IBM. HT is obviously a milestone of detecting line and regarded as a special approach[2] to analyze shape and movement of image with noise and irrelevant data. It is a standard tool to extract geometric primitives. Obviously HT is very time and space consuming: assuming the number of image pixels is n, there will be $n*(n-1)/2 \sim n^2$ kinds of combinations between two points. Then if all points are compared with the certain line, the complexity is $n*(n*(n-1))/2 \sim n^3$. In addition to some experimental applications, this method is negated by ascending computer time consuming. Therefore many interesting studies have been made to deal with that problem.

Subsequently, improved Hough Transform algorithms which make examination more robust, were developed, such as the DHT, Adaptive HT(AHT), Combinatorial HT(CHT), Curve Fitting HT(CFHT), Dynamic Combinatorial HT(DCHT), Dynamic Generalized Hough

Manuscript received September 5, 2006.
Manuscript revised September 25, 2006.

Transform(DGHT), Discrete Hough Cosine Transform(DHCT), Optimal Bayesian HT, Probabilistic HT(PHT), Progressive probabilistic HT(PPHT), Randomized HT(RHT), Dynamic RHT(DRHT), Random Window RHT(RWRHT), Window RHT(WRHT), Connective RHT(CRHT), Regularized HT. Those methods can reduce memory space[3][4], computation time[5][6][7][8] and improve accuracy[9][10][11][12]. DHT[16] was improved based on SHT(Standard Hough Transform) and presented in 1972. DHT increased calculating accuracy, and dissolved some limited functions of SHT. RHT[6] was proposed on 1990. RHT, different from standard HT and DHT, adopted a different mapping relationship. So it can handle any straight line in image, greatly reduces computation complexity O(n2) at the most bad cases, and storage space because of emptying the accumulator of the parameter space which is emptied during calculate process. Otherwise, lots of no denominated HT algorithms, aimed at the particular condition and added many constrains, resolve detecting line problem.

HT can be considered as mathematical statistics based line detection approach. For a instance, NOLD(Neighbor-orientation line detection algorithm)[16] based on HT is a new parameter space model algorithm which uses gathering property of straight line, sets speed be linear relationship with number of boundary points, and designs a smaller one dimension fixed parameter space.

It was already proved by lots of experiments that those methods, which do not process but directly discover straight line in the pixel layer, encounter difficulties in computation complexity, algorithm robustness, and description. As digital images are discrete, it is easy to describe the collinear relationship between pixels in a small region. A feasible approach can be described as follows: firstly, the image is divided to some small regions, and a shorter vector straight line is found at each region. Secondly, we analyze collinear relationship among them at short straight line level. The visual cortex of human brain has physiological reaction for straight line in image during line recognition. For instance, the visual cortex has strong nerves responses, if there are different directions straight lines in the neighboring regions[17].The results of physiological research can be used for reference to discover straight line at pixel level, for example, illusory contours, which does not have picture contrast and does not exist outline, but it can be seen by image clues(light-and-shadow contrast or compensate light grid), and sport contrast or two eyes contrast[18]. Evidently, the grouping short line segment algorithm can be designed by simulating physiological experiment process. DSCC (Directional Single- Connected Chain), which is another grouping short line segment algorithm, made up the weakness of classical chain code method in discovering the straight line segments of different width. It makes use of probability and mathematical statistics, and adopts clustering method to group long line from short line segments. The shortcoming of this type of algorithms is poor robustness, for lots of methods are aimed at some particular applications, and are added scene constrains, which result in lines in image having the additional features.

A new method of line detection and line description, based on combinations of multiple three-pixel elementary units, is presented in this paper. The difference between this method and other line detection algorithms mentioned above is that it is based on geometric features. The following paper is organized in this way: we first discuss the principal difficult of line detection. In section 2, we present details of the line detection algorithm based on combinations of multiple three-pixel elementary units, significance on syntax pattern recognition, and a hierarchy network-computing model. Experimental results are given for a performance comparison between our algorithm and classical algorithms on both synthesized images and real scene data. Finally, we summarize the strength and limitation of our new approach and discuss the potential future work.

2. The elementary units feature of the saw-toothed straight line

2.1. Design patterns of 3-pixel elementary unit

Formation of an imagine device, which is equipped in robot and focal plane staring array equipments, usually is circle or square 2D array. The digital image got by those devices is 2D array composed of luminance and color. Because sensors are always arrayed discretely in the device, whatever high resolution, line that is shown on digital camera or screen must be saw-toothed at theory, similar the line being drawn by Word or Draw software, then image understanding to computer is based on this type of approximate line segment. This is an important precondition to construct patterns of elementary unit. The key point is that the differences of pixel are positions and colors, but there is no concept of high level to the hardware. The straight line we called is the observer perception in mental, exactly the plenty of pixels are combined and considered as line by visual system of the observer. It is optimized combination issue, however, biological visual system costs few time to compute optimized combination. The line-detecting algorithm is anti-procedure of line constructing, but it is more complex than the line constructing. The line detecting algorithm has to face below key problems: (a) the rule to form straight line, (b) the sufficiency results of the line detecting, (c) the

Index	1	2	3	4	5	6	7	8	9	A	B	C
Graphics of Elementary Unit	□□□ ■■■ □□□	□□■ ■■□ □□□	□□□ □■■ ■□□	□□■ □■□ ■□□	□■□ □■□ ■□□	□□■ □■□ □■□	□■□ □■□ □■□	□■□ □■□ □□■	■□□ □■□ □■□	■□□ □■□ □□■	□□□ ■■□ □□■	■□□ □■■ □□□
Approximate Angle of elementary Unit	0 or π	$\pi/6-$	$\pi/6+$	$\pi/4$	$\pi/3-$	$\pi/3+$	$\pi/2$	$2\pi/3+$	$2\pi/3-$	$3\pi/4$	$5\pi/6-$	$5\pi/6+$
Sign of Elementary Unit	a	b	c	d	e	f	g	h	i	j	k	L

Table 1: Twelve patterns of 3-pixel elementary unit

efficiency of algorithm, (d) the adaptability for the approximate lines.

At first, we analyze constitutes of straight line, the straight line is created by combining pixels which have near distance to line equation at computer graphics[19]. This type of lines is called ideal saw-toothed line. But the boundary of the image is not ideal saw-toothed line, after the image is sharpen and extracted the boundary. We investigate the micro construction of saw-toothed straight line, and discover that the repetitions at two levels exist in a straight line, the first is in the sub-line segment level, the second is in pixel level, which give us important clue for "repeated combination of pattern". The straight line is combined by lots of pixels, so "repeated pattern" is "pattern of point combination". What combination can be designed as elementary units? We find 3-pixel combination pattern is appropriate to two sides: combination quantity and foundation property of pattern. All twelve patterns of 3-pixel elementary unit are shown in table 1. Every elementary unit is corresponding to a declarator and sign of angle.

A single pixel doesn't have direction, which means that any direction line extended to this point need be evaluated, but 3-pixel elementary unit has direction, for example, elementary unit pattern 1 and elementary unit pattern 7 are impossible in same line obviously, so decision times are decreased and collinear judgment is simplified totally. An important advantage of this design is easy to be implemented in hardware circuit directly, parallel of high granularity. At same time it can be directly mapped to nerve mechanism for visual information. We also can

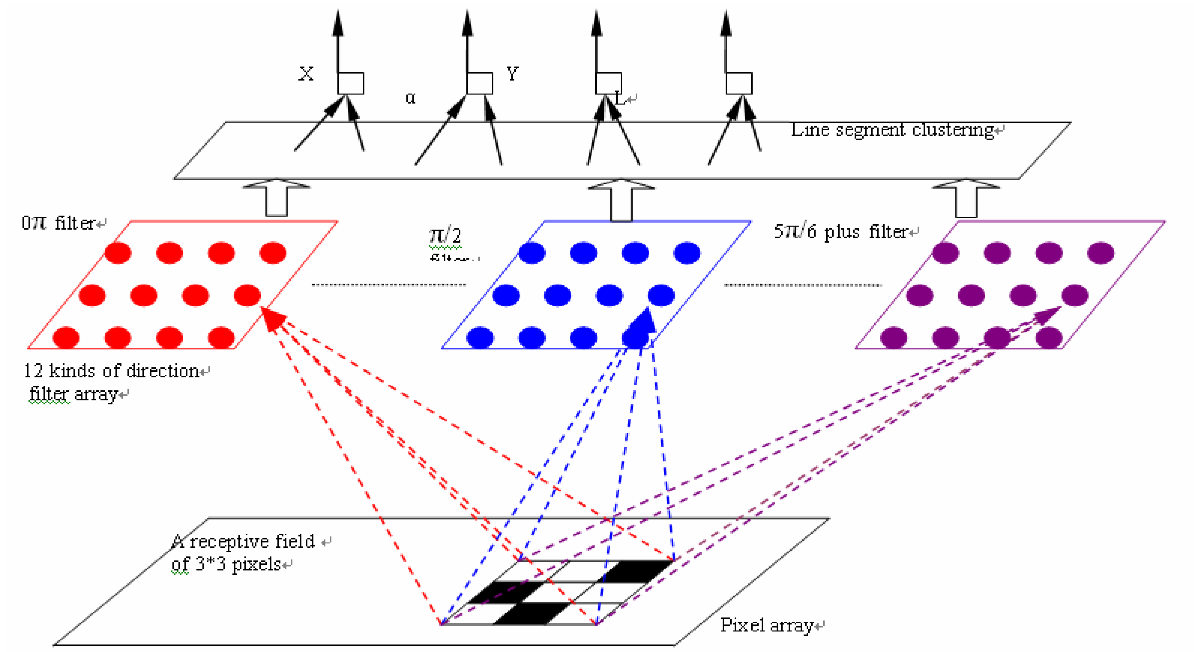


Figure 2: Hierarchy network computing model to detecting pattern of elementary unit.

increate the number of pixels in elementary unit, e.g. 4-pixel or 5-pixel.

2.2. Hierarchy compute mode and data structure fro detecting elementary unit

After defining patterns of elementary unit, we need design a parallel computing model to filter twelve kinds of patterns of elementary unit[20].Hierarchy compute mode for this target is illuminated in Fig.2. There are twelve kinds of direction filter arrays, and each of them is in change of searching matched pattern of elementary unit at given position where is overlapped with the center pixel of 3*3 pixels receptive field. The layer that cluster mass of elementary units and above direction filter layer is to output line segment represented by vector {X(X-coordinate of segment center point), Y(Y-coordinate of segment center point), α(angle of segment), L(length of segment)}. This computing model is completed parallel structure in theory. We design a data structure that is described by C++ language to realize the detecting action.

2.3. Combinations for different angles of line segments

Similarly matched angle in Table 3 is calculated by following steps. Firstly, to each continuous line, a certain elementary unit is outputted in detecting pattern of elementary unit layer, and it is represented by symbol i (0≤i≤0xC). Secondly, pattern sequence is got according relationship of pixels or computer units. Thirdly, each elementary unit has a similarly matched angle $\alpha(i)$ (α is mapping function), so we can computer all angles corresponding to pattern sequence and $\sum_{j=1}^{L_{pattern\_sequence}} \alpha(pattern\_at(j))$ ($pattern\_at(j)$ is mapping function). At last, the similarly matched angle of line is calculated by equation

$$\alpha' = \frac{\sum_{j=1}^{L_{pattern\_sequence}} \alpha(pattern\_at(j))}{L_{pattern\_sequence}}.$$

From experiment in Table 3, looking into the compositions of elementary unit patterns of ideal saw-

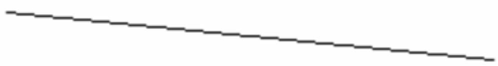
Index	Line Segment	Actual Angel α	Pattern Sequence	Similarly Matched Angle α'
1		16.59°	23112312312311231231231123123112312 31231123123123112312311231231231123 12312311231231123123123112312311231 23123112312312311231231123123123112 31231231123	17.64°
2		113.85°	89898989789898978989898978989897898 98989789898989789898978989898978989 89789898989789898978989898978989898 978989897898989897898989789898989	114.85°
3		172.94°	CB1111111CB1111111CB11111111CB11 11111CB1111111CB1111111CB1111111C B11111111CB1111111CB1111111CB1111 111CB1111111CB11111111CB1111111C B1111111CB1111111CB1111111CB11111 111CB1111111CB1111111CB1111111CB 1111111CB11111111CB1111111CB11111 11CB111	171.90°

Table 3: Combining patterns of elementary units for matched angle of line segment.

toothed line segment, we found the following rules: if the line whose angle is one of four directions 0° (180°), 45°, 90° and 135° is regarded as ideal line, the saw-toothed line is composed by ideal line segments whose number is limited and length is fixed. There are eight patterns of compositions. Following symbols defined in Table 1, a grammar G is an ordered quadruple of the form: $G=(V,T,P,S)$, where $V=\{S,\alpha,\beta_1,\beta_2,\beta_3,\beta_4,\beta,P,L,C_1,C_2\}$ is a nonterminal vocabulary, $T=\{a,b,c,d,e,f,g,h,i,j,k,l\}$ is terminal vocabulary, and $P=\left\{\begin{array}{l}\alpha\to a\mid g\mid d\mid j,\\ \beta_1\to bc\mid cb,\\ \beta_2\to ef\mid fe,\\ \beta_3\to hi\mid ih,\\ \beta_4\to kl\mid lk,\\ \beta\to\beta_1\mid\beta_2\mid\beta_3\mid\beta_4,\\ \ldots\ldots\end{array}\right\}$ is the production set of the grammar, $S\in V$ is identifies the starting symbol of G. Therefore the generative grammar of straight line is

$$\begin{array}{l}C_1\to\alpha^{n1}\beta^{n2},\\ C_2\to\alpha^{n3}\beta^{n2},\\ P\to(C_1)^{n4}C_2\mid(C_2)^{n4}C_1,\\ L\to(C_1'\mid C_2')\mid PL\mid(C_1'\mid C_2'),\\ L\to(C_1'\mid C_2')\mid LP\mid(C_1'\mid C_2')\end{array},$$

where C_1' and C_2' are sub-sequences of C_1 and C_2 respectively. The expression $[(\alpha^{n1}\beta^{n2})^{n4}(\alpha^{n3}\beta^{n2})]^n$ that is sequence of symbol is derived, where α, β belong to pattern set which is identified in Figure 5, $n1$, $n2$, $n3$, $n4$ and n are nonnegative integer, and $[(\alpha^{n1}\beta^{n2})^{n4}(\alpha^{n3}\beta^{n2})]$ is described as a repeated segment. Let $D_t(t\in T)$ denotes approximate angle of elementary unit at Table 1, and D_L (L is sequence of symbol) is sum of all approximate angles. Then according to method at Table 3, the similarly matched angle of straight line is

$$\frac{(D_\alpha*(n1*n4+n3)+D_\beta*(n2+1))*n}{(n1*n4+n3+n2+1)*n}=\frac{D_\alpha*(n1*n4+n3)+D_\beta*(n2+1)}{n1*n4+n3+n2+1}.$$

It is obvious that the angle of ideal saw-toothed line is not related with repeated times, but the angle of repeated segment. Let $\omega=\dfrac{n2+1}{n1*n4+n3}$, then the similarly matched angle is $\dfrac{D_\alpha+D_\beta*\omega}{1+\omega}$. It is got by directly observing that length of single segment has significant impact to angle of line and very short sequence (e.g. 3 pixels) doesn't look like straight line. We should find out the length of line segment that confines the change of similarly matched angle, even if an elementary unit is added. In other words, it is stabilized that the line is formed by pixels sequence. Supposed ω_1 is coefficient of original line, and a new coefficient is ω_2 after a pixel being increased, so the discrepancy value of two similarly matched angles is $\left|\dfrac{(D_\alpha-D_\beta)(\omega_2-\omega_1)}{(1+\omega_1)(1+\omega_2)}\right|$. No matter the new added elementary unit is α or β, generally $n2=1$ and $n4=1$, it is validated that the discrepancy value doesn't exceed 2° when the length of line is larger than 10 pixels. It is proved that similarly matched line is stabled and is tended to angle between two end points, after cumulated length is increased.

Comparing two angles at Table 2, we find they are very approximate. It is illuminated that design of 3-Pixel elementary unit is reasonable. All kinds of lengths and directions line segments can be combined by a little of categories, limited quantity and orderly elementary units.

The notable defect, simply detecting pixels are collinear, is that the red dotted line is intersected with the black lines in Figure 4, and the points of intersection are collinear, but the dotted line is not what we want to seek. Our method, which doesn't detecting line in pixel level but in the foundation of first step combination, can overcome this disadvantage.

3. Detecting and describing saw-toothed line segment based on clustering

3.1. Observations used on clustering

After an image is carried out edge process, we get outline. They are processed by 12 directions filter arrays, mass of short line segments are created which are tiny and fixed angle. Then we need to connect them to be longer line segments. If the number of lines is considered as computational complexity of image, the procedure of clustering combines lots of short line segments and reduces the complexity automatically.

So many detected elementary units that have space positions and patterns information spread on the plane, and we can discover the line by clustering method. It is not certain and has vast combinations of testing collinear of pixels that “where short line segments clustered together and which angle line formed”. However, our mental feeling is that “the lines jump out apparently by themselves”. There are several beneficial observations that shouldn’t be neglected to decrease complexity.

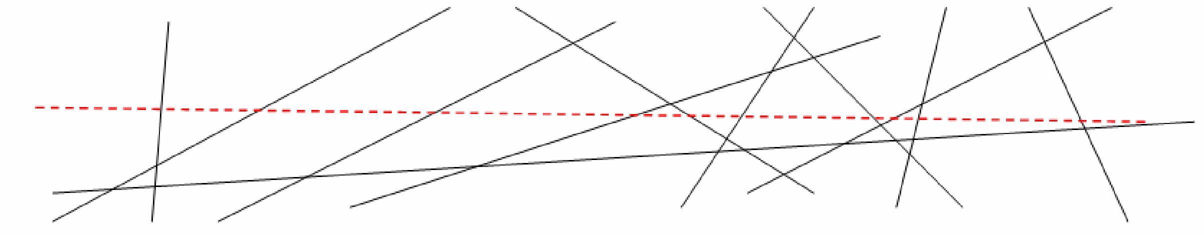


Figure 4: The red dotted line is intersected with the back lines.

(a) The collinear pixels are neighbor or neighbor by transfer relationship. If the elementary units are separated on the space, they shouldn’t be collinear. So they can be omitted to detect.

(b) “Elementary Unit 1” with “Elementary Unit 7” and “Elementary Unit 2” with “Elementary unit A” aren’t collinear obviously in Table 1.

(c) Our mission is to selectively combine plenty of elementary units in the space. But the decision standard is that if they belong to same class, they can form a straight line, and new elementary unit is added, at last the angle of line becomes stable.

(d) If some elementary units can form a line, there is little difference between the created line and ideal line. It is small swing between top and bottom or based on cross point. And the swing scope should under one pixel. If the swing range is changed, the different accuracy line can be created, which is helpful to analyze the edges of image, for those edges aren’t ideal saw-toothed line.

3.2. The simplified clustering method

The above-mentioned method is bases on the principle of PDP, whose construction is complicated. If we review complexity of nerve system of biology that has visual sensation, we will not feel strange about such simple function being needed so complicated construction. The simplified clustering algorithm to detecting line is got by traditional method of concentration.

Step 1: Randomly choose discovered elementary units as clustering seeds. Each seed belongs to one class, marked a flag.

Step 2: The seed starts growing by choosing elementary units around it. If the new elementary unit’s space position isn’t adjacent to any elementary unit in the class, we refused adding it into the class at this time. The distinguish standard of adding new elementary into class is to enhance probability that new extended class forms a line. Each time the seed is defined by the elementary unit at the end of line segment that is formed by the class.

Step 3: If many points belonging to same line are selected as seeds, the classes that are created by seeds respectively have same elementary units. The more elementary units are same, the more probability lines formed by classes are collinear. At this time, it can be considered to merge them and get larger class that is longer straight line.

Step 4: The class is line, if it can’t be extended. We begin to remove some elementary units in discovered line to simplify computer space. Returning to pixel level and pick one pixel, if elementary units of all adjacent pixels are in the class, the elementary units should be deleted from elementary unit space. We repeat above process unit there is no available elementary unit to delete. Go to Step 1 and repeat again.

4. The experimental results

An experiment system was developed based on the hierarchy network computing model and algorithm above-mentioned. Several contrastive experiments are showed as below. The images in the experiments are carried out edge process. The edge detecting arithmetic operators is Canny, but the matrix pixels don’t be formed vectors.

Experiment 1: Figure 5 illuminates comparison for detecting broken line. Figure 5 is divided into four regions. Original image is shown at top left corner, and there are several discontinuous short line segments. The line segments discovered by our algorithm at bottom left corner are represented by vectors whose format is (start point coordinate, end point coordinate, length). The lines that are drawn by start point and end point based on line vectors are displayed at top right corner, and the color of line is to distinguish different lines. The bottom right is placed image which has lines are detected by classical Hough algorithm. Comparing two images at right side, we can obviously find that our algorithm discovered the broken area of lines, but Hough algorithm didn’t. HT regards the short line segments that are collinear as a long line, if it doesn’t be enhanced.

Experiment 2: Figure 6 shows an image including a convex quadrangle at top left corner, which is added noise to disturb, and edges of the convex quadrangle is blurred. There are lots of pixels having collinear relationship, but they aren't in same line. Comparing two images at right side of Figure 6, algorithm presented in this paper can recognize the quadrangle from the noise points, but HT has large of error discovered lines which don't exist in the original image.

Experiment 3: Many short lines are arrayed as grating at the top left corner of Figure 7. If the length value of every short line is smaller than 1/3 count of the lines, for HT, the vote of the line which isn't true line in man's eyes is larger times than the obviously existent short line's, during transforming and calculating. So whatever the threshold is changed at peak-detection, error detecting cannot be avoided. However our algorithm totally avoids this problem.

Experiment 4: Table 8 is an experiment comparing our algorithm with other classical line-detecting algorithm at compute time and space consuming. 100 true scene images whose size is 640*480 are input at AMD XP1.8G 256M memory computer. From contrast, our algorithm improves compute speed and space consuming remarkably. Higher speed and less space consuming are key points to application in embedded system robot.

Experiment 5: Ceramic tile image is shown at the top left corner of Figure 9, and the two-value result of image is placed at the bottom left corner. The lines are discovered by our algorithm are presented at top right corner, and experiment result of HT is at bottom right corner. It is illuminated that result of our algorithm is very clear and it is of advantage to discover main feature of ground.

Experiment 6: The ground texture image composed by lawn, cement step and bricks is shown at top left corner of Figure 10. The two-value pixel image is placed at bottom left corner. The line detecting result of our algorithm is presented at top right corner. The result of HT is at bottom right corner. It is indicated that our method can accurately discover edge by contrast images. Robot could gain meaningful clues to make decision for ground situation based on the result of our algorithm.

Experiment 7: The building image is shown at top left corner, and two-value result of image is placed at bottom left corner. The line vectors of detecting build edge by our algorithm are presented at top right corner, and the result of HT is at bottom right corner. From contrast experiment, it is illuminated that sketch lines of building are accurately discovered by our algorithm, but lots of redundant and fault lines are created by HT.

Experiment 8: A car image is shown at top of Figure 12. The line segments discovered by our algorithm and represented by different colors are placed at bottom right corner. The result of HT is at bottom left corner. The strong signal given by the contrast experiment result is that our algorithm provides the good foundation of feature description to recognize car.

5. Conclusions

If the pixels of image are regarded as physical feature, then the line combined by orderly arrayed pixels belongs to semantic feature area. The aim of detecting line is to acquire the most basic semantic features that are sources of intersection, parallel and clip angle etc. features. So image representation and description method based on line segments supplies basic technique for image meaning abstracting and pattern recognition based on feature.

The clustering algorithm to detect straight line presented in this paper doesn't suit other types of outlines. However the patterns of elementary unit constructed line can be combined to form other types of curves according other rules. So pattern representation and description method based on local micro features is not only used for line, but also can be applied for more complex graphics, only if the suitable syntax features are found, which is advantage of syntax pattern recognition. The patterns of elementary unit designed in this paper have strong ability to describe the scene outlines and are adapted to represent the image meaning, furthermore, the realized represented level is above the physical level. It is the fact that the pattern recognition methods which lack meaning description have lots of defaults, whereas the method in this paper aims at improving this issue.

The features representing and detecting method presented in this paper is completely based on model, rather than according to samples. So our method doesn't need training, and there are not threshold value learning and generalization issues, therefore it has very high adaptability.

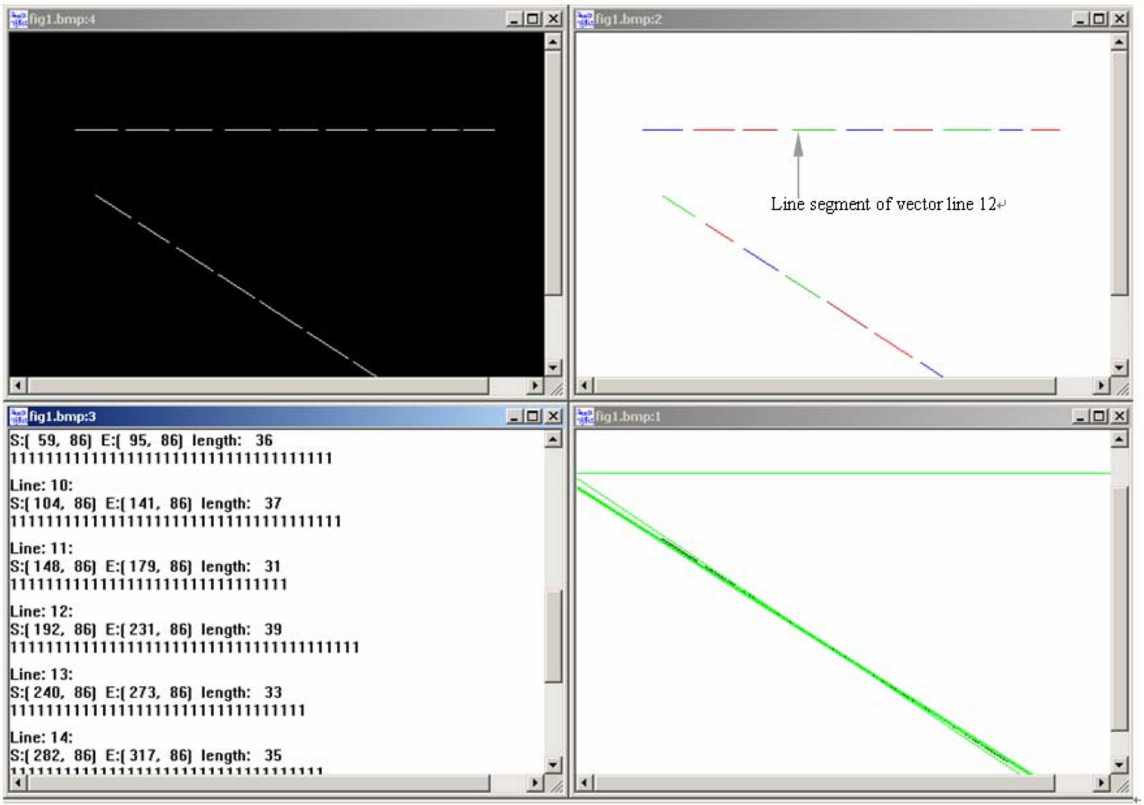


Figure 5: Comparison for detecting broken line.

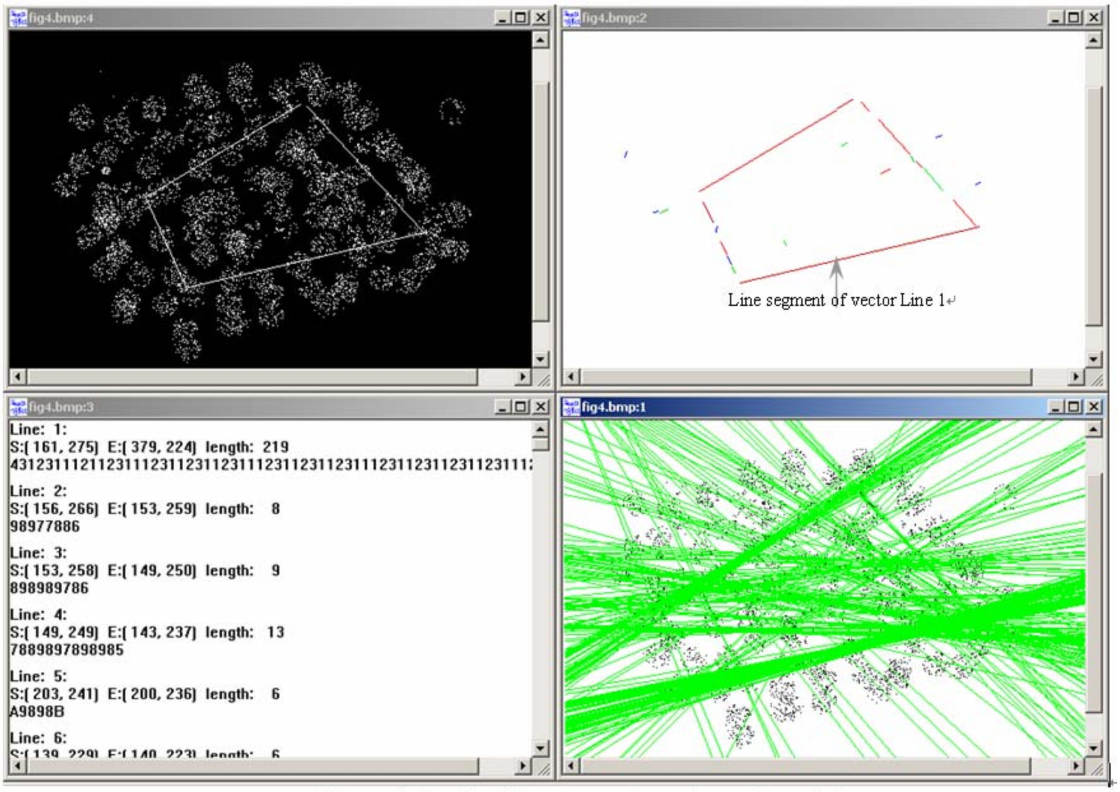


Figure 6: Result of the process image has noise points.

Figure 7: Result of detecting lines which are arrayed as grating.

	SHT algorithm	RHT algorithm	PHT algorithm	Our algorithm
Time (Second)	150.2	43.9	12.7	11.5
Memory(Byte)	2880K	54.78K	24.92K	10.73K
Scanned Pixels	6,738	1570	1639	6,421

Table 8: Comparing performance for line detecting algorithms

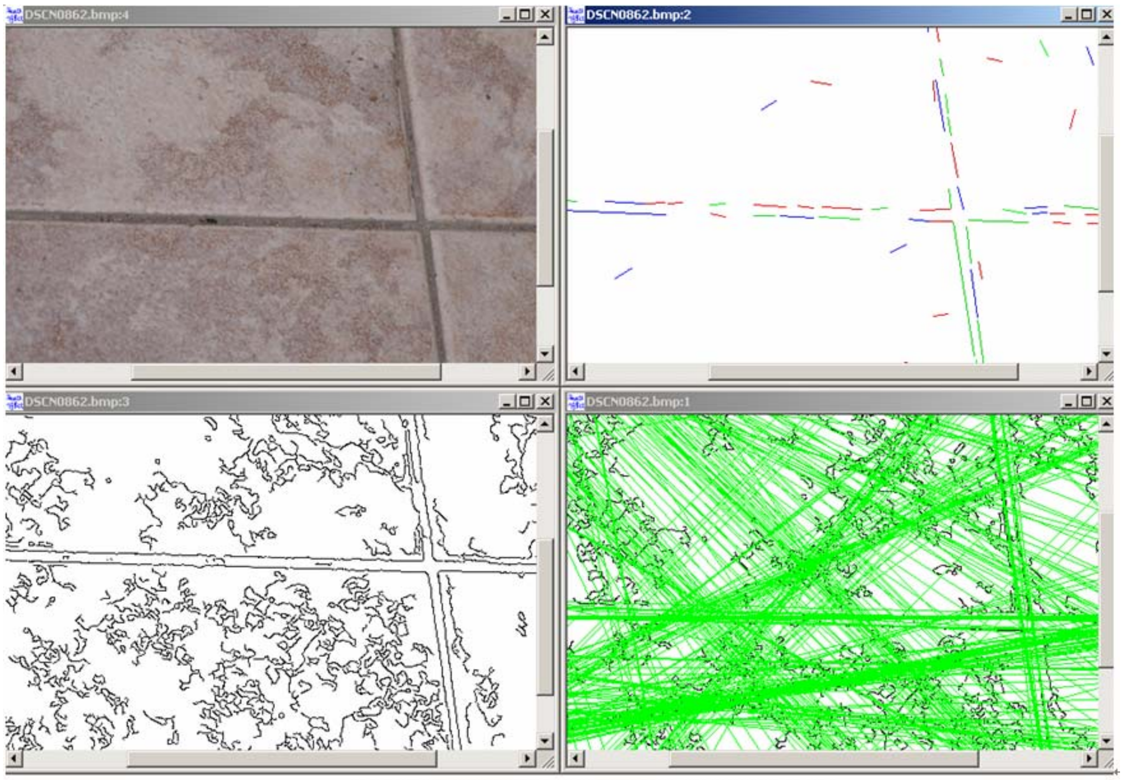


Figure 9: Detecting line for Ceramic tile image.

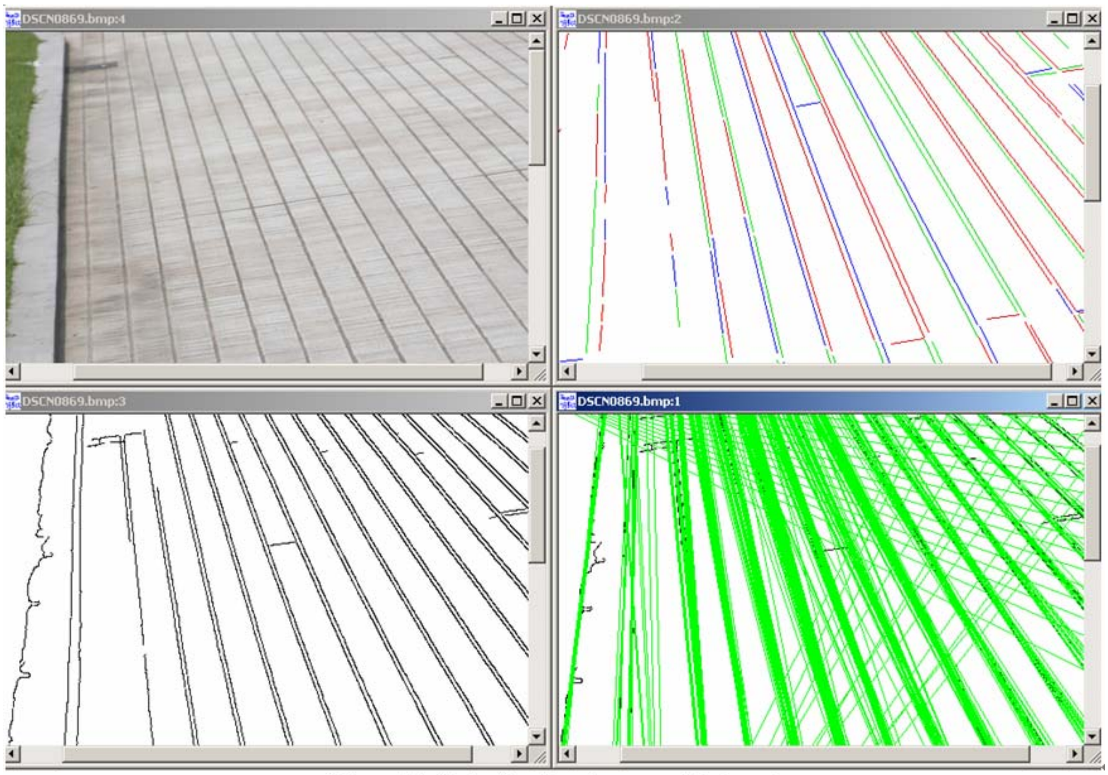


Figure 10: Detecting line for ground texture image.

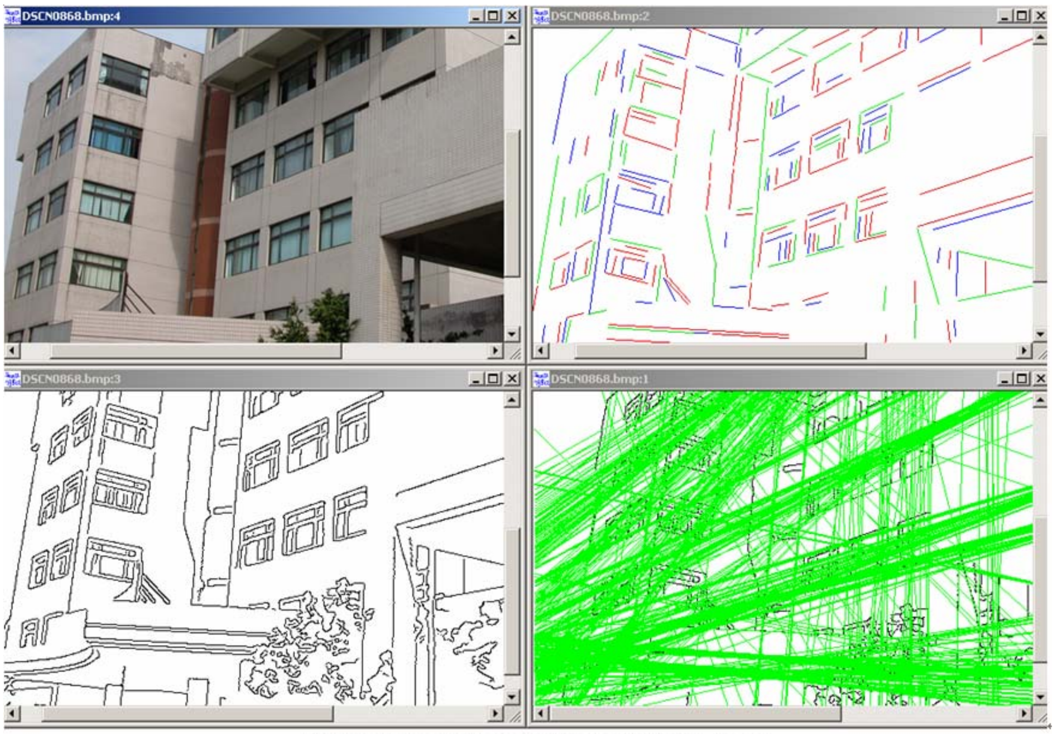


Figure 11: Detecting edge line for building image.

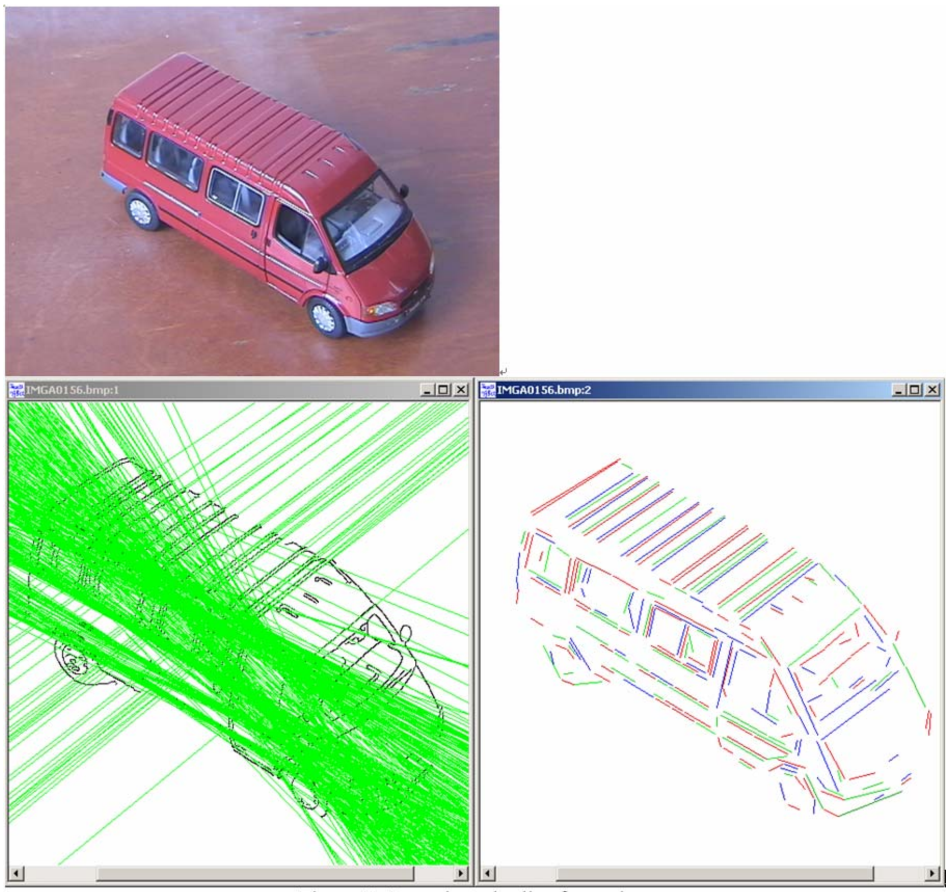


Figure 12: Detecting edge line for car image.

Acknowledgement

This work is supported by the National Science Foundation of China under Grant No 60303007, and China Basic Research Project (973) under grant no. 2001CB309401.

References

[1] P.V.C. Hough, Method and means for recognizing complex patterns, U.S. Patent 069654, 1962

[2] J. Illingworth and J. Kittler, A Survey of the Hough Transform, Computer Vision, Graphics and Image Processing, vol. 44, issue: 1, 1988, 87-116.

[3] D. Ben-Tzvi, V.F. Leavers, and M.B. Sandler, A Dynamic Combinatorial Hough Transform, Proc. Fifth Int'l Conf. Image Analysis, 1990, pp. 152-159.

[4] Y. Zhang and R. Webber, A Windowing Approach to Detecting Line Segments Using Hough Transform, Pattern Recognition, vol. 29, 1996, pp. 255-265.

[5] N. Kiryati, Y. Eldar, and A.M. Bruckstein, A Probabilistic Hough Transform, Pattern Recognition, vol. 24, 1991, pp. 303- 316.

[6] Kultanen, P.; Xu, L.; Oja, E, Randomized Hough transform (RHT), Pattern Recognition, 1990. Proceedings, 10th International Conference on, vol. 1, June 1990, pp. 631 -635

[7] B. Gatos, S.J. Perantonis, and N. Papamarkos, Accelerated Hough Transform Using Rectangular Image Decomposition, Electronics Letters, vol. 32, no. 8, 1996, pp. 730-732.

[8] S.J. Perantonis, B. Gatos, and N. Papamarkos, Block Decomposition and Segmentation for Fast Hough Transform Evaluation, Pattern Recognition, vol. 32, no. 5, 1999, pp. 811-824.

[9] T. Van Veen and F. Groen, Discretisation Errors in Hough Transform, Pattern Recognition, vol. 14, 1981, pp. 137-145.

[10] W. Niblack and D. Petrovic, On Improving the Accuracy of the Hough Transform: Theory, Simulations and Experiments, Proc IEEE CS Conf. Computer Vision and Pattern Recognition, 1988, pp. 574-579.

[11] J. Illingworth and J. Kittler, The Adaptive Hough Transform, IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 9, 1987, pp. 690-698.

[12] P. Palmer, J. Kittler, and M. Petrou, Using Focus of Attention with the Hough Transform for Accurate Line Parameter Estimation, Pattern Recognition, vol. 27, 1994, pp. 1,127-1,134.

[13] A. L. Kesidis, Nikos Papamarkos, On the Inverse Hough Transform, IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 21, no. 12, 1999, pp. 1329-1343.

[14] A. Neri, Optimal Detection and Estimation of Straight Patterns, IEEE Trans. Image Processing, vol. 5, no. 5, 1996, pp. 787-792.

[15] R. O. Duda and Peter E. Hart, Use of the Hough Transformation To Detect Lines and Curves in Pictures, Graphics and Image Processing, Communications of the ACM, vol. 15, no. 1, January 1972, pp. 11-15

[16] Ron Shpilman, Victor Brailovsky, Fast and robust techniques for detecting straight line segments using local models, Pattern Recognition Letters, vol. 20, 1999, pp. 865-877.

[17] Hans-Christoph Nothdurft, Latency effects in orientation pop out, Vision Research, vol. 42, 2002, pp. 2259-2277.

[18] Leo Poom, Visual summation of luminance lines and illusory contours induced by pictorial, motion, and disparity cues, Vision Research, vol. 41, 2001, pp. 3805–3816.

[19] David F. Rogers, Procedural Elements for Computer Graphics (2nd edition), McGraw-Hill, 1998, 65-78

[20] WEI Hui, HE Xingui, A simulated annealing algorithm for combination optimization in line-finding, Computer engineering and design, Vol.21(3)，2000，6-11

WEI Hui was born in November 1971. He received his PhD in Department of Computer Science at Beijing University of Aeronautics and Astronautics in July 1998. From September 1998 to November 2000 he was a postdoctoral in the Department of Computer Science and Institute of Artificial Intelligence at Zhejiang University. In November 2000 he jointed the Department of Computer Science and Engineering at Fudan University as an associate professor. His research interests include artificial intelligence and cognitive science.

WANG Zengjin was born in November 1977. He got his B.S. degree in computer science from East China University of Science and Technology in 2001, and M.S degree in computer science from Fudan University in 2004. His research interest is computer vision.

LIU Bin was born in June 1980. He got his B.S. degree in computer science from Beijing University of Aeronautics and Astronautics in 2002, and M.S degree in computer science from Fudan University in 2006. His research interest is computer vision.