ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

IMAGE RECOGNITION BASED ON SIMILARITY OF CURVE CHARACTERISTICS INVARIANT TO TRANSLATION

[1]DINA INDARTI, [2]SARIFUDDIN MADENDA, [3]TUBAGUS MAULANA KUSUMA, [4]DEWI AGUSHINTA R
[1]Department of Information Technology, GUNADARMA UNIVERSITY
[2]Department of Computer Engineering, STMIK JAKARTA STI&K
[3]Department of Electrical Engineering, GUNADARMA UNIVERSITY
[4]Department of Information Systems , GUNADARMA UNIVERSITY
E-mail: {[1]dina_indarti, [2]sarif, [3]mkusuma, [4]dewiar}@staff.gunadarma.ac.id

ABSTRACT

Objects similarity are required in image recognition, such as in quality control problems. Characteristics of objects in an image are represented by the characteristics of the detected edges. Curve is one of the detected edge beside point, line, and region. Curve provides geometric and structural information of the image. It gives better information than point. The existing research related to the similarity of curve characteristic are still facing problems associated with invariant to geometric transformations. Therefore, this research discusses how to use the characteristics of curve, so that the curve similarity can be invariant to translation. In using the characteristics of curve, the first step is curves tracking using 8 (eight) directions chain codes. Furthermore, the algorithms to determine the type of curve and to determine the similarity between curves which invariant to translation are explained. The definition of curve itself was used in determining the types of curves. The similarity of curves is determined based on the characteristics of curves. This proposed image recognition algorithm can detect similar image while geometric translation is occurred.

Keywords: *Curve Characteristic, Geometric Transformations, Image Recognition, Image Similarity, Translation*

1. INTRODUCTION

Computer vision is an automated process that integrates a large number of processes for visual perception, such as image acquisition, image processing, classification, recognition, and decision making. Computer vision can also be interpreted as a process of preparing a description of the objects contained in an image or identifying objects that exist in the image [1]. One of the problems in computer vision is object recognition. The problem in object recognition is to determine whether the image has a certain object or not in given image. In addition, the objects to be recognized often have a change in geometry, for example a change in position due to translation or rotation. Therefore, in object recognition, it is necessary to match the characteristics of objects in the image. Matching the characteristics of objects in image is a part of the computer vision which builds relationships between two or more images. Characteristics of objects in image represented by the characteristics of edges detection results. The detection results can be points, lines, curves, and regions.

The utilization of image recognition in object recognition problem is further applied in industrial field. Image recognition is used to determine whether the product quality suitable with established standards. Besides object recognition, similarity of object characteristics or image matching is also widely used in computer vision such as image registration, 3D reconstruction, robot navigation, structure from motion, and so on.

Currently, the major development occurred in research is on point matching. A point matching is most commonly used in image matching. In point matching, several methods have been investigated such as Speeded-Up Robust Features (SURF) [2], DAISY [3], the Gradient Location-Orientation Histogram (GLOH) [4] and the most famous is the Scale Invariant Feature Transform (SIFT) [5]. Point matching cannot provide geometrical and structural information although it can distinguish well between image and invariant to changes in 3D viewpoint, addition of noise and changes in lighting. In contrast, the curve provides the

Journal of Theoretical and Applied Information Technology
10th April 2015. Vol.74 No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

geometric and structural information of image. It gives better information than point matching. Therefore, curve matching is necessary and irreplaceable in many applications of image matching.

The curve is a one-dimensional geometric shape located in the plane or space. In general, the results of image edge detection have the form of curves in the plane. The types of curve that are generated from the edge detection image are a closed curve and an opened curve. Closed curve can be a kind of simple curves and non-simple curves. Opened curve can be a type of linear and non-linear curves either simple or not [6]. Types of curve used in this research are simple closed curves and simple opened linear and non-linear curves.

Research on curve matching is rarely performed compared to point matching and region matching. One method carried out curve-fitting method is an extension of SIFT descriptor and GLOH point [7]. The method of extending the idea of point descriptors SIFT and GLOH is using the Pixel Support Region (PSR) based on the characteristics of the area around the curve at 4 (four) directions. The drawback of this method is not invariant to change in scale. There is a curve-fitting method without using any limitations or prior information, called the Mean-Standard Deviation Line Descriptor (MSLD) [8]. However, MSLD have the same weakness which is not invariant to changes in scale.

The existing research related to the similarity of curve characteristic are still facing problems associated with invariant to geometric transformations. Therefore, this research discusses on how to use the curve characteristics, so that the curve matching can be invariant to the transformation, which in this case is geometric translation. In using the characteristics of curve, the first stage is searching form of curves using 8 (eight) directions chain codes which was more accurate than 4 (four) directions. Furthermore, algorithms to determine the types of curve and to determine the similarity between the curves that is invariant to translation is proposed. The algorithm was computationally implemented in MATLAB software.

2. RELATED WORK

2.1 Mean-Standard Deviation Line/ Curve Descriptor (MSLD/ MSCD)

MSLD/ MSCD is a method of matching line automatically from their neighborhood appearance without resorting to any other constraints or prior knowledge [8]. MSLD/ MSCD is constructed by following three steps:

1. For each pixel on line segment is defined as PSR and divided into non-overlapped sub-regions.
2. Each sub-regions obtained will be characterized by description vector.
   We get *Gradient Descriptor Matrix* (GDM) matrix with size of $4M \times N$ where M is sub-region number and N is pixel number on the line. But GDM matrix cannot be used directly because the size is still different with the line length.
3. Compute mean and standard deviation of the GDM matrix. Ilumination invariant, normalized mean and standard deviation used respectively to make a descriptor. Both are normalized separately because to maintain the influence of each to the descriptor.

The performance of the descriptor is tested using the correct ratio (CR) and the correct match (CM) criterions. The results show that MSLD/MSCD perform well on real images across rotation, viewpoint change, illumination change, and blur. Unfortunately, MSLD/MSCD is not scale-invariant.

2.2 Extend Point Descriptor For Line

This method extends SIFT and GLOH point descriptor for constructing line, curve, and region matching. The main orientation of line (curve, region) descriptor is determined by the average gradient direction of pixels on a line (curve, region). For each pixels on the line with length *N*, its SIFT and GLOH descriptor is computed based on rectangular (SIFT) or circle (GLOH) region centered at it. Then, they compute statistic vectors (Mean and Standard Deviation, Moments, Fourier coefficients) of these descriptors. This descriptor is also not scale-invariant [7].

2.3 Scale-Invariant Line Transform (SILT)

SILT detects line segments as local extrema in the scale-space. Then, each detected line segment is represented in a distinctive manner using Haar-like features. PCA is further deployed to improve upon the compactness and robustness of representation. Beside to achieve scale-invariant descriptor, SILT also handles the line matching

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

problems: not reliable of the end-points, line segments detected in the target image may be part of the line segment of the reference image, describing and representing a line effectively.

SILT is scale-invariant but the accuracy of SILT is not quite good (performance range about 70 - 65%) because the main objective of SILT is to cover scale-invariant while the accuracy is the second goal [9].

2.4 Line Matching Leveraged By Point Correspondeces

The basic idea of this approach is to use tentative point correspondences, which can be easily obtained by key point matching methods such as SIFT or DAISY, to significantly improve line matching performance, even when the point correspondences are severely contaminated by outliers [10].

Given two set of line segments, extracted from two images (the reference image and query image), $L_1 = \{l_1^1, l_1^2, \ldots, l_1^M\}$ and $L_2 = \{l_2^1, l_2^2, \ldots, l_2^M\}$. Then, a set of corresponding points $C = \{(x_1^i, y_1^i), (x_2^i, y_2^i), i = 1,2,\ldots K\}$ is obtained by a key point matching method. To achieve the rotation invariant, estimate the relationship of rotation between two images, θ from C. All of these are input to the line matching algorithm with threshold t_θ.

In order to match two lines with affine-invariant, the similarity of two line segments is derived from one line and two points. Then, line matching is done in keeping the uniqueness constraints. The meant of uniqueness is if l_1^i and l_2^j are a pair of matching lines, then there is no other line in L_2 such that its similarity to l_1^i is larger than similarity between l_1^i and l_2^j. There is also no other line in such that its similarity to l_2^j is larger than similarity between l_1^i and l_2^j. It allows many-to-one correspondences which can be happened when a line in one image is split into multi-segments. The resulting set of matching lines is represented as $LM = \{(l_1^{g(i)}, l_2^{f(i)}), i = 1,2,\ldots,N_L\}$ in which $(l_1^{g(i)}, l_2^{f(i)})$ represents a pair of matching lines and $g(i) \in [1, M], f(i) \in [1, N]$.

Compared to MSLD/MSCD, this method has better performance but in case of low-texture scenes, this method can be failed.

2.5 Line Matching Using Appearance Similarities and Geometric Constraints

In this method, the approach is based on three strategies to solve problems with matching line. The first is extracting lines in space scale so it makes the line matching algorithm invariant to changes in scale. Extraction using EDLines algorithm [11]. The second is checking the consistency of lines pair by combining similarity appearance and geometry constraints. Constraints geometry is the direction of line which is calculated from the histogram toward the line between reference image and target image. Third is solving matching problem by spectral methods [12].

3. METHOD

This study is divided into two stages. First stage is the formation of image database and the second stage is image recognition. In the first stage, edge detection, curve tracing, and curve type detection were performed. In the second stage, identical operations as in the first stage were performed. Furthermore, a comparison is conducted to determine the similarity between curve characteristics of a query image with the curve characteristic of all images stored in the database. The curve searching stage, curve type detection stage, and image recognition stage will be described in the following section.

3.1. Image Database Formation

During the database formation, edge detection was performed to extract image information from the curve components. Canny edge detection scheme was used for the above mentioned process [13]. In this stage, curve tracking, curve type detection and storage to database were taken place.

3.1.1. Curve searching

Following the edge detection process, the curves generated by the process will be searched to identify its type. Curve searching is performed per pixel by observing 8 (eight) directions of pixels [5].

Searching the curve is actually observing pixel by pixel value starts from the first row of first column, the second column and so on. If there is no white pixels with intensity of 255 found in the first row, then continue the search to the second row of first column, the second column and so on. The

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

search process ends if the first pixel found pixel which has a value of 255 in order to obtain the coordinates of the beginning of a curve. Then, the starting point will be stored. Afterwards, from this initial pixel, the next pixels or neighboring pixels is investigated in 8 (eight) directions. If the total intensity of neighboring pixels in the 8 (eight) directions is greater than or equal to 255, then the next pixel is still the same curve. Conversely, if the value is less than 255, then further search of initial pixels is needed until a pixel with pixel intensity of 255 is detected. The first pixel found is the starting point of the next curve. Coordinates of neighboring pixels in 8 (eight) directions are stored at the time of search process. Search continues until all the curves have been detected.

Pixel coordinates which are a pair of x and y that represents the position of pixel location are stored. The value of *x* and *y* are stored based on the 8 (eight) neighboring pixels. A value of -1 is also stored for partitioning coordinate values between curves. Flowchart of curve searching/tracking is shown in Figure 1.

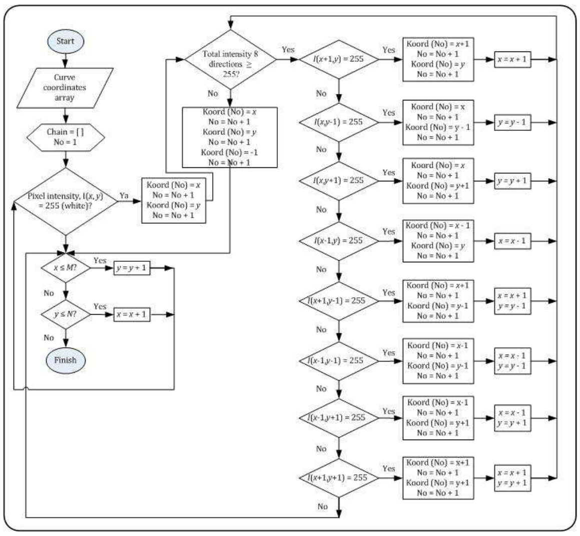


Figure 1: Flowchart of Curve Searching/Tracking

3.1.2. Detection of curve types

In the first phase of curve type detection is to identify whether a curve is classified as a closed curve or as an opened curve. If the *start* and *end* coordinates have the same value, then the curve is a closed curve. If it is not the same, then the curve is an opened curve [11].

The next phase is to determine whether the previously obtained opened curve is linear or non-linear. Angle calculation is needed to determine whether a curve is classified as a linear or a non-linear curve. Calculated angle is the angle between the starting point with each point from the midpoint to the end and the angle between the end points of each point from the starting point to the midpoint. If a curve has an angle amount exceeds a certain threshold, then the curve is classified to be non-linear. Otherwise, the curve is a linear curve. Angles threshold is used in this study to determine whether an opened curve is linear or non-linear. If the curve is opened, the curve is an opened non-linear curve.

Image with the identified curves shapes and types is stored in a database which will then be used in the process of image recognition. Figure 2 is a flowchart of curve type detection.

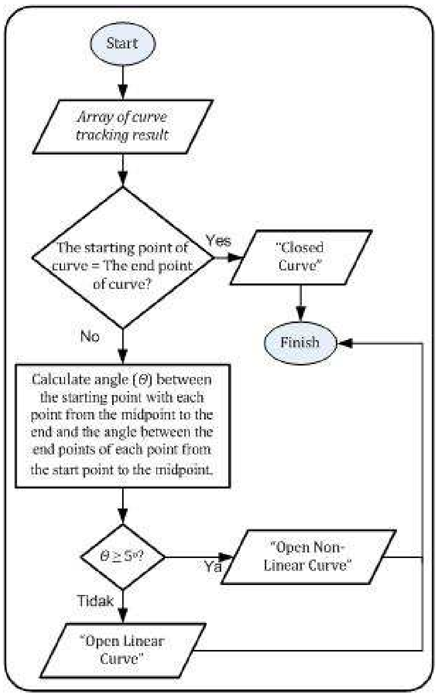


Figure 2: Flowchart of Curve Type Detection

After recognizing the type of curve, then the curve characteristics are stored in the database. For closed curves, stored data include coordinates, curve characteristic and the number of closed curves within the image. In opened linear curve, the stored data include coordinates, curves characteristics, opened curves linear gradient, and the number of opened linear curve. In non-linear opened curve, the stored data include coordinates curve characteristic and the number of opened non-linear curve. Curve coordinates contains a pair of *x*

Journal of Theoretical and Applied Information Technology
10th April 2015. Vol.74 No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

and *y*. The gradient of opened linear curve or straight line is obtained by dividing the difference between the final value of *y* with initial value of *y* and the difference between the final value of *x* with the initial value of *x*. The number of curves is obtained by calculating the number of -1 value in the coordinates array of curve search results.

3.2. Image Recognition

Image recognition is conducted by comparing the characteristics of query image with the characteristics of image in the database that has been previously stored. Characteristics of the query image are obtained in the same way with the image in the database. In the early stages or coarse level of similarity is the first filter to select the candidate images which have similarity with a query image. After the first filtering, images which have small probability of similarity with query image do not need to be processed in fine level similarity. So, the recognition stage can be faster. The criteria in coarse level similarity are having the same amount and same length of the corresponding curve in the database image and the query image. Later in the advanced stages or fine-level image similarity, recognition considers the geometric transformation which is translation. Figure 3 shows the flowchart of image recognition which consists of coarse level similarity and fine level similarity.

In coarse level similarity matching stage, the first criterion is having the same amount of curves for corresponding type of curve in the query image with image in databases. The value of last element of each curve types stored in the query image and image database is compared to determine the number of curve similarity. If the amount of curves for corresponding type of curve in the query image with image in databases is different, then the comparison is continued with the new image database. If the image database meets the common criteria, then second criterion will be further analyzed. The first criterion is having the same length of curves for corresponding type of curve in the query image with image in databases. If the length of curves for corresponding type of curve in the query image with image in databases is different, then the comparison is continued with the new image database. If it is identical, then the image will be further analyzed in the fine level similarity stage.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

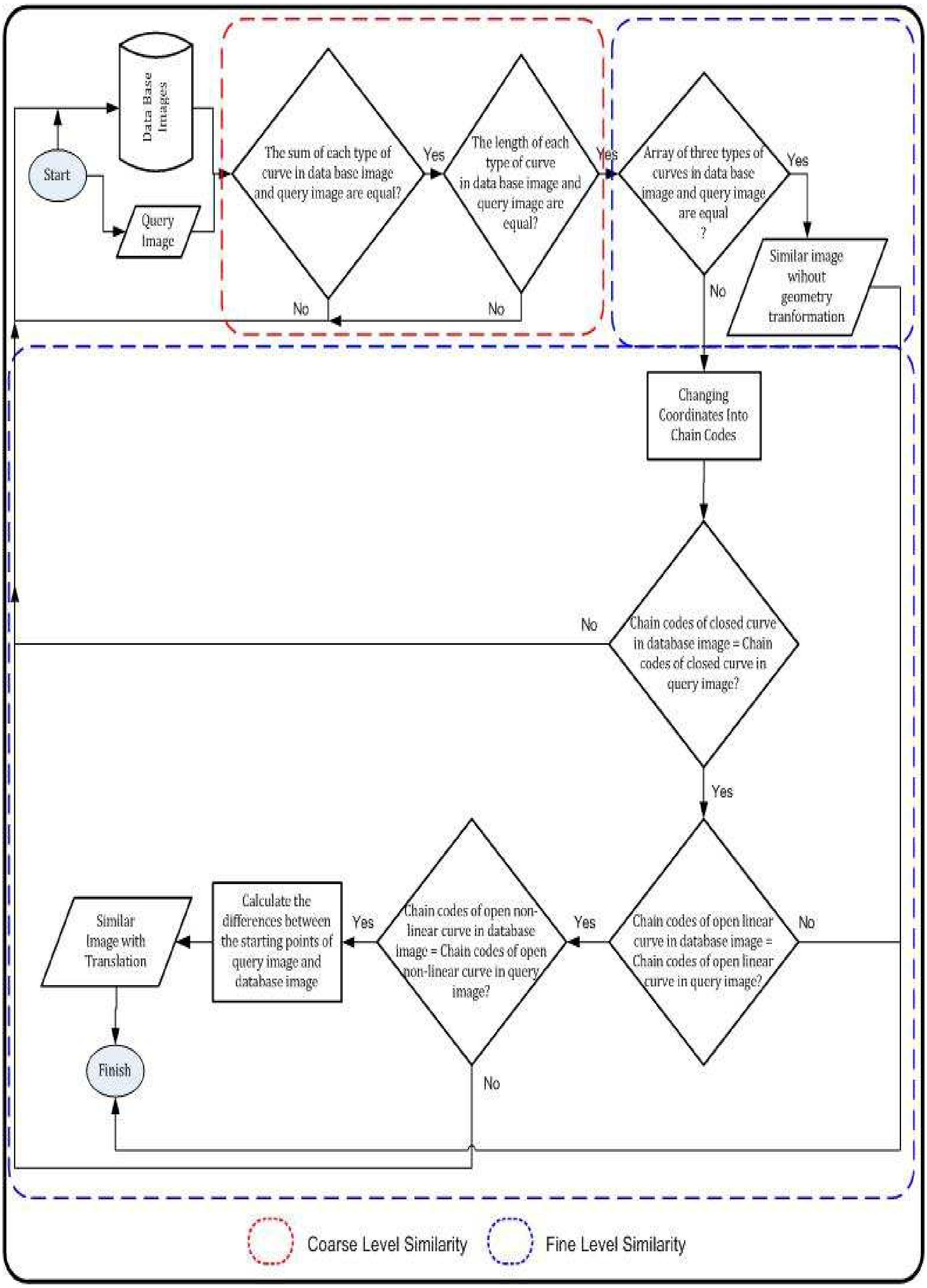


Figure 3: Flowchart of Image Recognition

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

At fine level similarity stage, the first criterion is the array elements of three types of curves in database image and query image (if both of them have all types of curves). If the elements of the array are not the same, then the comparison is continued with the new image database. If the elements of the array are the same, the next stage is to change the array that contains coordinates of curve into an array that contains the chain code of curves. The changes were made so that image recognition invariant to translation. Flowchart of the change of coordinates into a chain codes of curve is shown in Figure 4.

After changing the coordinates into chain codes has been completed then further analysis in fine level similarity was performed by checking the similarity of chain code database image and the query image as shown in Figure 3.

Comparing the difference between the initial coordinates of the image database and the query image is to determine whether the translation occurs or not. If the difference is 0 (zero) then the image in database is similar to the query image without translation. If the difference is not 0 (zero) then the image in database is similar to the query image with a translation equal to the difference between the initial coordinates of image database with query image.

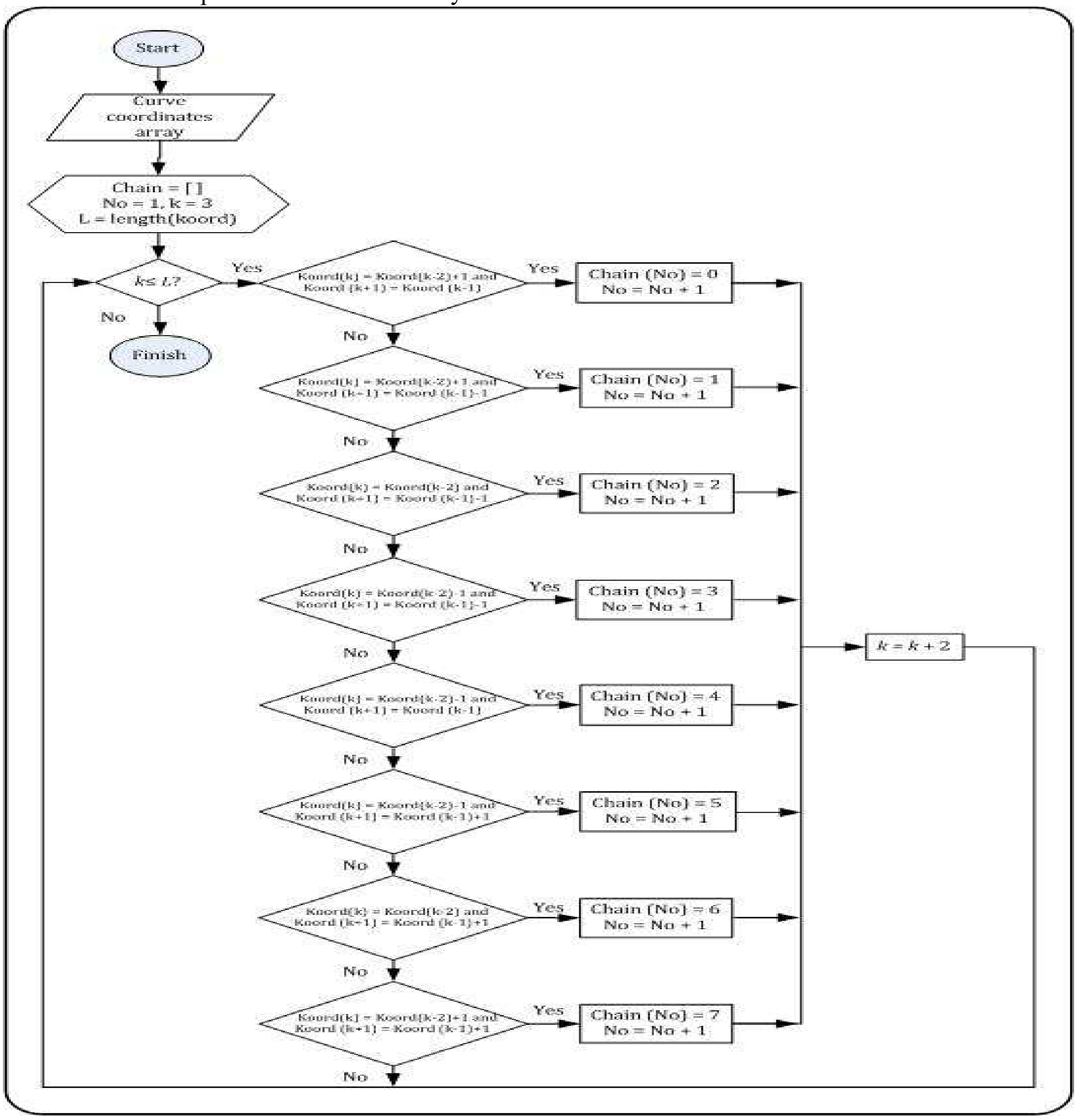


Figure 4: Flowchart of Changing Coordinates Into Chain Codes

Journal of Theoretical and Applied Information Technology
10th April 2015. Vol.74 No.1

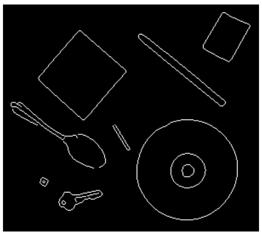
ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

4. RESULT

The implementation results consist of edge detection, the formation of image database and the image recognition. The result of image database formation consists of curve tracking and curve types recognition.

4.1. Canny Edge Detection Result

Figure 5 is an example of implementing Canny edge detection to the image.

(a) (b)

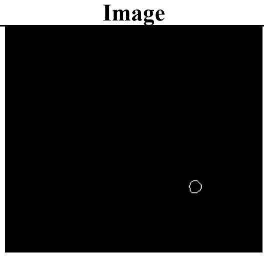
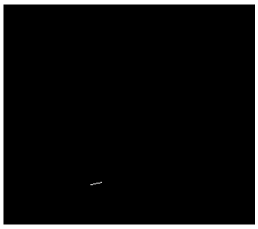
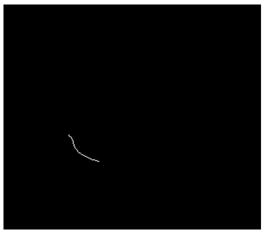
Figure 5: Image Extraction Using Canny
(a) Original Image (b) Edge Detection Results

Figure 5(a) shows the original image to be detected using the Canny method. Canny edge detection results are shown in Figure 5(b), which consists of 18 curves.

4.2. Results of Image Database Formation

The result of curve tracking is an array contained *x* and *y* coordinates. At the end of the array is a value of -1, which separates each coordinates of different curves in an image. Table 1 shows the result of curve searching.

Table 1: Curve Searching Result

No	Image
1.	 **Curve Searching Result (Coordinates (*x*, *y*)):** [221 180 220 181 219 181 218 182 217 183 217 184 217 185 216 186 216 187 216 188 217 189 217 190 217 191 217 192 218 193 219 194 220 194 221 194 222 194 223 194 224 194 225 194 226 193 227 192 228 192 229 191 229 190 230 189 230 188 230 187 230 186 230 185 229 184 228 183 227 182 226 181 225 180 224 180 223 180 222 180 -1]
2.	 **Curve Searching Result (Coordinates (*x*, *y*)):** [105 216 106 216 107 216 108 215 109 215 110 215 111 215 112 214 113 214 114 214 115 214 116 214 117 213 118 213 -1]
3.	 **Curve Searching Result (Coordinates (*x*, *y*)):** [77 153 77 154 78 155 79 155 80 156 80 157 81 158 81 159 82 160 82 161 82 162 82 163 83 164 83 165 83 166 84 167 84 168 85 169 86 170 87 171 87 172 88 173 89 174 90 174 91 175 92 176 93 176 94 177 95 177 96 178 97 178 98 179 99 179 100 180 101 180 102 181 103 181 104 182 105 182 106 182 107 182 108 183 109 183 110 183 111 184 112 184 -1]

Types of curve which can be detected are closed curve, opened linear curve, and opened non-linear curve. By using algorithm shown in Figure 2, image number 1 in Table 1 is detected as a closed curve. Image number 2 is an opened linear curve and image number 3 is an opened non-linear curve.

4.3. Results of Image Recognition

In Figure 3, it is shown that image in Figure 5(a) is similar with a number of images shown in Table 2.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Table 2: Image Recognition Result

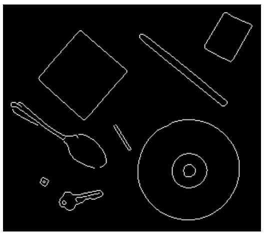
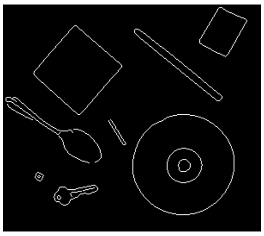
No	Image
1.	
2.	

Table 2 shows the image recognition results obtained two images similar to the query image. The first image is recognized as similar image without translation. The second image is recognized as similar image with translation of 14 pixels to the left and 16 pixels up.

5. FUTURE WORK

The proposed image recognition method in this paper is invariant to translation. Image recognition based on curve characteristics can be investigated so it can be also invariant to other geometrical transformations such as rotation and changing in scale.

REFERENCES:

[1] Munir, R, 2007. Pengolahan Citra Digital. Bandung: Informatika.

[2] Bay, H., Ess, A., Tuytelaars, T., and Gool, L. V., 2006. “SURF: Speeded Up Robust Features”. *Journal of Computer Vision and Image Understanding (CVIU),* 110 (3), pp. 346-359.

[3] Winder, S., Hua, G., and Brown, M., 2009. “Picking The Best DAISY”. *In Proc of International Conference On Computer Vision And Pattern Recognition*, pp. 178-185.

[4] Mikolajczyk, K. and Schmid, C., 2005. “A Performance Evaluation Of Local Descriptors”. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27 (10), pp. 1615–1630.

[5] Lowe, D. G, 2004. “Distinctive Image Features From Scale-Invariant Keypoints”. *Int. Journal of Computer Vision*, 60, pp. 91–110.

[6] Varberg, D., Purcell, E. J., Rigdon, S. E., 2007. Calculus Ninth Edition. New York: Pearson Prentice Hall.

[7] Liu, H. M., Wang, Z. H., and Deng, C., 2010. “Extend Point Descriptors For Line, Curve And Region Matching”. *In Proc of the Ninth International Conference on Machine Learning and Cybernetics (ICMLC)*, 1, pp. 214–219.

[8] Wang, Z., Wu, F., and Hu, Z., 2009. “MSLD: A Robust Descriptor For Line Matching”. *Journal of Pattern Recognition*, 42(5), pp. 941–953.

[9] Khaleghi, B., Baklouti, M., and Karray, F. O., 2009. “SILT: Scale Invariant Line Transform”. *IEEE International Symposium on Computational Intelligence in Robotics and Automation (CIRA)*, pp. 78–83.

[10] Fan, B., Wu, F., and Hu, Z., 2010. “Line Matching Leveraged By Point Correspondences”. *In Proc of International Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 390-397.

[11] Akinlar, C. and Topal, C., 2011. “EDLines: A real-time line segment detector with a false detection control”. *Pattern Recognition Letters*, 32, pp. 1633-1642.

[12] Zhang, L. and Koch, R., 2012. “Line Matching Using Appearance Similarities And Geometric Constraints”. *Pattern Recognition*, pp. 236-245.

[13] Canny, J., 1986. “A Computational Approach To Edge Detection”. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 8, pp. 679–698.

[14] Freeman, H, 1974. “Computer Processing Of Line-Drawing Images”. *Computing Surveys,* Vol. 6, No. 1, pp. 57-97.