Half Ellipse Detection

Nikolai Sergeev and Stephan Tschechne

Institute for Neural Information Processing, University of Ulm,
James-Franck-Ring, 89069 Ulm, Germany

`{nikolai.sergeev,stephan.tschechne}@uni-ulm.de`

Abstract. This paper presents an algorithm of half ellipse detection from color images. Additionally the algorithm detects two color average values along the both sides of a half ellipse. In contrast to standard methods the new one finds not only parameters of the entire ellipse but also the end points of a half ellipse. The paper introduces a new way of edge and line detection. The new detector of edges in color images was designed to extract color on the both sides of an edge. The new line detector is designed to optimize the detection of endpoints of a line.

Keywords: Edge detection, line detection, ellipse detection.

1 Introduction

In the literature there can be found several methods of ellipse detection e.g. [Tsuji and Matsumoto, 1978, H.K. Yuen, 1988]. Most of them use Hough transformation [Hough, 1962].

The new object recognition system introduced in [Sergeev and Palm, 2011] uses half ellipses. In other words not only ellipse parameters but also end points of an ellipse segment are needed. For that reason a new half ellipse detection algorithm was developed.

The algorithm consists of four steps: edge detection, line detection, combination of lines to chains, detection of half ellipses in line chains. These will be consecutively described in the paper.

The first step is one of the best discussed topics in the literature. Unfortunately such edge detectors as [Canny, 1986, Prewitt, 1970] work with grayscale images and do not extract color on the both sides of an edge. On this account a new edge detector was developed.

Hough transformation is normally used to detect lines. As the original Hough transformation is not quite suitable to locate endpoints of a line a new algorithm was designed.

2 Edge Detection

At first the new concept of an edge will be introduced. A pixel corner will be denoted as a vertex. An edge is a pair of neighbored vertexes. There are horizontal and vertical edges.

Fig. 1. A 3×3 pixel grid. One black pixel with RGB values $(0,0,0)$ at the center. Other pixels are white and have RGB value $(1,1,1)$.

Fig. 2. The set of vertexes of the pixel grid

Fig. 3. Horizontal and vertical edges

Fig. 4. Good(left) and bad(right) transition of the red RGB component in a line

Now it will be described how vertical contrast edges are detected. The horizontal ones are getting detected in exactly the same way after having transposed the pixel grid. Each line is getting processed separately. A line of pixels is a sequence of RGB vectors $(p_i)_{i \in \{1, \dots, n\}} \subseteq \mathbb{R}^3$. For $1 \leq i < j \leq n$ contrast intensity $CI(i, j)$ is defined as

$$CI(i, j) = \frac{\|p_j - p_i\|}{j - i + 2} - \left(\sum_{k=i}^{j-1} \|p_{k+1} - p_k\| - \|p_j - p_i\| \right). \quad (1)$$

The first term $\frac{\|p_j - p_i\|}{j - i + 2}$ says the contrast intensity is the higher the bigger the difference and shorter transition is. Number 2 or generally $j - i + 2$ is chosen heuristically. This choice is due to adapt the edge detection to human visual perception. The second term $\sum_{k=i}^{j-1} \|p_{k+1} - p_k\| - \|p_j - p_i\|$ is due to diminish the contrast intensity in case of a "bad" transition as showed in Fig. 4. In case of a "good" transition the term is equal zero otherwise it is positive.

The first step of edge detection is to find $1 \leq i < j \leq n$ pairs with

$$i \leq k < l \leq j \Rightarrow CI(k, l) \leq CI(i, j) \quad (2)$$

In other words contrast intensity of (i, j) -transition should be bigger than contrast intensity of any inner transition. An (i, j) -transition satisfying 2 will be denoted as (i, j) -true transition. In the second step such (i, j) -true transitions have to be excluded for which a (k, l) -transition exists with

$$k < i < j \leq l \vee k \leq i < j < l. \quad (3)$$

In the next step a central edge has to be extracted from an (i, j) -transition as Fig.5 shows. In the last step each side of the extracted edge gets a RGB value. An edge is defined as an ordered pair of vertexes. Having the first and the second vertex an edge gets the left and the right side. As Fig.6 shows the left side of the edge gets the RGB value of the left outer pixel of a transition. The right side - that of the right outer one.

Fig. 5. An edge at the center of a transition

Fig. 6. Color extraction for an edge

3 Line Detection

At first a heuristic way to check if three discrete points "build" a line will be introduced. For the well known Manhattan norm $\|\cdot\|_M$ defined as

$$\|a\|_M = \sum_{i=1}^2 |a_i| \quad (4)$$

three vertexes a, b, c build a line or $line(a, b, c) = 1$ if

$$\|c - a\|_M = \|c - b\|_M + \|b - a\|_M \quad (5)$$

and

$$\left| \frac{\|b - a\|_M}{\|c - a\|_M} |c.x - a.x| - |b.x - a.x| \right| \leq 1. \quad (6)$$

The last term can be formulated as integer multiplication

$$\|c.x - a.x\| \|b - a\|_M - |b.x - a.x| \|c - a\|_M \leq \|c - a\|_M. \quad (7)$$

A line is defined as an ordered sequence of edges $(v_1^i, v_2^i)_{i \in \{1, \dots, n\}} \in \prod_{i \in \{1, \dots, n\}} \mathbb{R}^2 \times \mathbb{R}^2$. To check if point p lies on a line the algorithm takes the first point of the line *first* and tests for every vertex v_j^i of every edge if $line(first, v_j^i, p) = 1$. To check if an edge (p, q) lies on a line $(v_1^i, v_2^i)_{i \in \{1, \dots, n\}}$ both vertexes p, q of the edge get tested if they lie on the line. Finally $line(begin, p, q) = 1$ has to be true.

The process of line detection is the process of line proceeding. A typical situation is: A line is given. It has to be checked if there is an edge in direct neighborhood of the last point of the line which lies on the line in the sense defined above. If so a new edge gets added to the line. Initially a line consists of a single edge. The purpose is to proceed it maximally.

Fig. 7. Two edge candidates to proceed line. The bad one offends the first condition (5) of the line test.

4 Making Line Chains

Last section delivers a set of lines. The purpose of this section is to combine them to chains. Surely there are a lot of possible ways. To avoid the unnecessary details the most trivial one will be described.

To initialize a chain one line has to be selected randomly. The chain now consists of this one line. One of the endpoints of the initial line gets chosen arbitrarily. Let it be denoted as p . Then a set of all lines with one endpoint in some neighborhood of p has to be built. If the set is not empty a line $\tilde{l}$ has to be selected with for example the angle between l and $\tilde{l}$ closest to 180° . The selected line $\tilde{l}$ should be added to the chain and the procedure should be repeated for the remaining endpoint of $\tilde{l}$. The procedure must be iterated until the chain can not be proceeded any more. The entire process has to be replicated for the remaining end point of the initial line l .

One line can be part of only one chain. After having been chosen a line gets tagged as a used one and can not be used any more. It is due to reduce the number of produced chains and therefore run time.

The next chain should be initialized with a line untagged as a used one.

5 Definition, Representation and Detection of a Half Ellipse with Color

5.1 Definition of a Half Ellipse

For $\mathbb{C} = \mathbb{R}^2$ and $P(\mathbb{C})$ standing for power set of $\mathbb{C}$ a half ellipse is defined as a pair $(e, B) \in \mathbb{C}^2 \times P(\mathbb{C})$ with $e_1 \neq e_2$ for which $(a, b, t_0, \delta) \in [0, \infty) \times [0, \infty) \times \mathbb{R} \times \{-1, 1\}$ as well as $(c, \beta) \in \mathbb{C} \times \mathbb{R}$ exist so that

$$B = \left\{ T_c R_\beta \begin{pmatrix} a \cos t \\ b \sin t \end{pmatrix} \middle| t \in \begin{matrix} [t_0, t_0 + \delta\pi] \\ \cup \\ [t_0 + \delta\pi, t_0] \end{matrix} \right\} \quad (8)$$

and

$$e_1 = T_c R_\beta \begin{pmatrix} a \cos t_0 \\ b \sin t_0 \end{pmatrix}, \quad (9)$$

$$e_2 = T_c R_\beta \begin{pmatrix} a \cos(t_0 + \pi) \\ b \sin(t_0 + \pi) \end{pmatrix}. \quad (10)$$

T_c, R_β stand for translation, scaling and rotation respectively. The set of half ellipses will be denoted with HE . In other words a half ellipse consists of end-points $e_1, e_2 \in \mathbb{C}$ and of a set of bow points $B \in P(\mathbb{C})$. There are mainly two reasons to use half ellipses. An affine transformation $A \neq 0$ always maps a half ellipse onto another half ellipse. The second reason is the variety of half ellipses as Figure 8 shows.

Fig. 8. Examples of half ellipses

5.2 Rotation, Translation and Scaling Invariant Representation of a Half Ellipse

Now a unique rotation, translation and scaling invariant representation of a half ellipse will be introduced. At first two preliminary definitions are needed. For $x, y \in \mathbb{C}$ with $x \neq y$ the function $F_{x,y}^1$ is defined as

$$F_{x,y}^1 : \begin{cases} \mathbb{C} \rightarrow \mathbb{C} \\ z \mapsto \frac{z-x}{y-x} \end{cases}. \quad (11)$$

Figure 9 shows the geometric meaning of the mapping. $F_{x,y}^1$ is an affine transformation with $F_{x,y}^1(x) = 0$ and $F_{x,y}^1(y) = 1$. The second function $F^2 : HE \rightarrow \mathbb{C}$ is defined as

$$F^2(e, B) = \begin{pmatrix} \max_{x \in B} \left| \left(F_{\frac{e_1+e_2}{2}, e_1}^1(x) \right)_1 \right| \\ \max_{x \in B} \left| \left(F_{\frac{e_1+e_2}{2}, e_1}^1(x) \right)_2 \right| \end{pmatrix}. \quad (12)$$

Fig. 9. Geometric meaning of $F_{x,y}^1(z)$

Finally the invariant representation $F^3 : HE \rightarrow \mathbb{C}$ is defined in such a way that for always existent $x, y \in B$ with

$$F^2(e, B) = \begin{pmatrix} \left| \left(F_{\frac{e_1+e_2}{2}}^1, e_1(x) \right)_1 \right| \\ \left| \left(F_{\frac{e_1+e_2}{2}}^1, e_1(y) \right)_2 \right| \end{pmatrix} \quad (13)$$

$$F^3(e, B) = \begin{pmatrix} z - \text{SIGNUM}(z) \\ \left(F_{\frac{e_1+e_2}{2}}^1, e_1(y) \right)_2 \end{pmatrix} \quad (14)$$

with

$$z = \left(F_{\frac{e_1+e_2}{2}}^1, e_1(x) \right)_1. \quad (15)$$

For (e, B) in Figure 10

$$F^3(e, B) = \begin{pmatrix} M_1 - 1 \\ M_2 \end{pmatrix}. \quad (16)$$

It can be shown that for each $x \in \mathbb{C}$ a half ellipse $(e, B) \in HE$ exists with $F^3(e, B) = x$. Additionally for two half ellipses $(e, B), (\tilde{e}, \tilde{B}) \in HE$ with $F^3(e, B) = F^3(\tilde{e}, \tilde{B})$ it can be shown that they can be transformed in each other through translation, rotation and scaling. On the other side $F^3(e, B)$ is invariant to translation, rotation and scaling of (e, B) .

Fig. 10. Representation of a half ellipse

5.3 Extraction of a Half Ellipse from a Sequence of Lines

This section shows how to check whether a chain of lines is a half ellipse. The chain should be interpreted as a pair $(e, B) \in \mathbb{C}^2 \times P(\mathbb{C})$. e_1, e_2 are the endpoints of the line chain. $B \subseteq \mathbb{C}$ consists of all points of the chain.

Now $(a, b, t_0, \delta) \in [0, \infty) \times [0, \infty) \times \mathbb{R} \times \{-1, 1\}$ and $(c, \beta) \in \mathbb{C} \times \mathbb{R}$ must be found for which the corresponding half ellipse $(e, \tilde{B})$ would have $F^3(e, \tilde{B}) = F^3(e, B)$. At first $F^2(e, B) = M \in \mathbb{C}$ has to be determined. It is trivial to calculate as $x \in B$ just have to be inserted in $F^1_{\frac{e_1+e_2}{2}, e_1}(\cdot)$. For

$$c = \sqrt{\frac{(M_1^2 + M_2^2) + \sqrt{(M_1^2 + M_2^2)^2 - 4M_2^2}}{2}} \quad (17)$$

and

$$d = \sqrt{\frac{(M_1^2 + M_2^2) - \sqrt{(M_1^2 + M_2^2)^2 - 4M_2^2}}{2}} \quad (18)$$

it can be shown that a can be chosen as $\frac{\|e_1 - e_2\|}{2}c$, b as $\frac{\|e_1 - e_2\|}{2}d$. It is known that

$$\cos^2 t_0 = \frac{1 - d^2}{c^2 - d^2}. \quad (19)$$

For $t = \arctan(\sin t_0 / \cos t_0)$ set $t_0 = -t$ if $F^3_{1/2}(e, B) \geq 0$ or $F^3_{1/2}(e, B) \leq 0$. Otherwise set $t_0 = t$. If $F^3_2(e, B) \geq 0$ than $\delta = 1$ otherwise $\delta = -1$.

The system says (e, B) is a half ellipse if all $x \in B$ lay in some ε -neighborhood of $(\tilde{e}, \tilde{B})$ with respect to maximum norm, which can be analytically determined as (a, b, t_0, δ) are known and only intersection points of four lines with the half ellipse are to be found Fig.11.

Fig. 11. Checking if a point is in an ε -neighborhood of a half ellipse with respect to maximum norm

5.4 Color Extraction

Similarly to an edge a half ellipse has the first and the last point. Hence it also has the left and the right site. A half ellipse is a sequence of lines. A line is a sequence of edges. The color of the right side of a half ellipse is set as the arithmetic mean of the right side RGB values of all edges belonging to the half ellipse. The color of the left site is set correspondingly.

6 Experimental Results

This paper describes the most important components of a half ellipse detector. Some details had to be omitted to spare space. The experimental results described below are valid for one implementation of the algorithm. The half ellipse detector was developed for a new object recognition system. To evaluate the system the well known database COIL-100 (Columbia Object Image Library) was used. The data set is described in [Nene et al., 1996]. It contains 7200 color images of 100 3D objects. One image is taken per 5° of rotation.

The computer used in the experiments has a processor Intel(R) Core(TM)2 Duo CPU P8600 @2.40 GHz 2.40 GHz and 4.00 GB RAM. The system is implemented in Java.

There were made 2 experiments with slightly different parameter settings. In the first experiment 18 views(1 per 20°) were used to learn each object. The remaining 5400 images were analyzed. A recognition rate of 99.2% was reached. In the second experiment 8 views(1 per 45°) were used to learn an object. The other 6400 were analyzed. A recognition rate of 96.3% was reached.

Average time demand to extract half ellipses from one COIL-100 image is 95 milliseconds.

The Table 1 compares the system with alternative methods. It is based on the results described in [Obdrzalek and Matas, 2011],[Yang et al., 2000] and [Caputo et al., 2000].

Table 1. Comparison with alternative results

Method	18 views	8 views
LAFs	99.9%	99.4%
Sergeev and Palm	99.2%	96.3%
SNoW / edges	94.1%	89.2%
SNoW / intensity	92.3%	85.1%
Linear SVM	91.3%	84.8%
Spin-Glass MRF	96.8%	88.2%
Nearest Neighbor	87.5%	79.5%

7 Conclusion and Future Work

There have been mainly two reasons to develop the algorithm described in this paper. First of all the endpoints of half ellipses are very important for the object representation used in the new system [Sergeev and Palm, 2011]. Secondly the color on the both sides of a half ellipse has to be extracted.

This paper does not deliver a complete description of a half ellipse detector. As the space is limited it only contains a terse depiction of the principle elements of the algorithm. Nevertheless the remaining trivial details can be elaborated easily.

The running time as well as the quality of the detector still have to be improved significantly. There are probably a variety of equivalent heuristic means to upgrade the quality. At the same time only one way of running time optimization appears to be realistic: parallelization.

References

- [Canny, 1986] Canny, J.: A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (1986)
- [Caputo et al., 2000] Caputo, B., Hornegger, J., Paulus, D., Niemann, H.: A spin-glass markov random field for 3d object recognition. In: *NIPS 2000* (2000)
- [H.K. Yuen, 1988] Yuen, H.K., Illingworth, J.: Ellipse detection using the hough transformation. In: *4. Alvey Vision Conference* (1988)
- [Hough, 1962] Hough, P.V.C.: Method and Means of Recognising Complex Patterns. US Patent 3069654 (1962)
- [Nene et al., 1996] Nene, S.A., Nayar, S.K., Murase, H.: Columbia Object Image Library, COIL-100 (1996)
- [Obdrzalek and Matas, 2011] Obdrzalek, S., Matas, J.: Object recognition using local affine frames. In: *BMVC* (2011)
- [Prewitt, 1970] Prewitt, J.: *Picture Processing and Psychopictorics*, ch. (75-149). Academic Press, London (1970)
- [Sergeev and Palm, 2011] Sergeev, N., Palm, G.: A new object recognition system. In: *VISAPP 2011* (2011)
- [Tsuji and Matsumoto, 1978] Tsuji, S., Matsumoto, F.: Detection of ellipses by a modified hough transform. *IEEE Trans. Comput.* (1978)
- [Yang et al., 2000] Yang, M.-H., Roth, D., Ahuja, N.: Learning to recognize 3D objects with sNoW. In: Vernon, D. (ed.) *ECCV 2000*. LNCS, vol. 1842, pp. 439–454. Springer, Heidelberg (2000)