

Integrating Color and Gradient into Real-Time Curve Tracking and Feature Extraction for Video Surveillance

Huiqiong Chen and Qigang Gao

*Faculty of Computer Science, Dalhousie University Nova Scotia,
Canada*

1. Introduction

Efficient curve detection and feature extraction is a very important step in many video-related applications, such as video content analysis and representation, surveillance systems, medical diagnoses, etc. For example, in video surveillance systems, curve tracking and feature extraction can be used in detecting moving targets from a video, allowing potential interesting events to be identified and analyzed for surveillance purposes. Curve detection usually includes edge detection and post processing procedures such as thinning, curve fitting or edge following, etc. Curve detection can significantly reduce less important data in a video frame while preserving structural information. Perceptual features can be extracted from curves for analysis or recognition purpose. However, Conventional edge detectors provide only an output of edge pixels. It is difficult to extract perceptual features directly from the edge detection results. Post-processing is then needed to remove noise, fill gaps, and fit edge pixels into curves. Unfortunately, most post-processing is too time-consuming for use in real-time applications (Fan et al., 2001).

Most edge detection techniques fall into two categories, gradient based methods and second order methods. Gradient-based methods detect edges based on the first derivative of the intensity. Examples include the Sobel, Prewitt, Roberts, and Canny operators, in which the Canny operator (Canny 1986) is the one of most commonly used edge detector. The second order methods find edges by searching for zero crossings in the second derivative of the intensity. Examples of the second order methods include the *Laplacian*, *Marr-Hildreth operators*, etc.

In color images, the color information also can be used to determine discontinuities in the color space (Cheng et al. 2001). Perez and Kock claimed in (Perez & Koch, 1994) that hue in HSI is more robust to certain types of highlights, shading, and shadows than the components in RGB, normalized RGB, or CIE color spaces. The edges with small hue change are removed from the Canny detector output in (Perez & Koch, 1994). A compass operator is proposed in (Ruzon & Tomasi, 1999), which considers distribution of pixel colors during edge detection. A 2D edge detection functional is used in (Qian & Huang, 1996), which is guided by the zero-crossing contours of the Laplacian-of-Gaussian (LOG) to find the edge locations.

In curve feature extraction, the Hough Transform is a well known technique for detecting curve features. It transforms the image space into the parameter space to find possible

features. Hough transform is tolerant to edge gaps, but its computational cost grows exponentially with the number of parameters used to represent the curves. Some efforts were made for modifying the Hough transform to reduce the dimensions of the parameter space (Yip et al. 1992). A constrained Hough transform is proposed in (Olson, 1999) for handling localization errors. Curve fitting divides a curve into segments and fits segment with lines, circular arcs or high order curves (Pei & Horng, 1995). Another way of finding curves is converting an edge image to a graph and search curves based on the criteria of the shortest path (Cheng et al. 2004). These methods use relatively complex mathematic models or search strategies; therefore they are computationally expensive.

Gao and Wong presented a curve detection technique called GET (Generic Edge Token) edge tracker in (Gao & Wong, 1993) based on gradient information. **The GET edge tracker can detect the edge traces and partition the traces in terms of Generic Edge tokens at the same time.** In this paper, we present an improved color-based GET tracker, which can produce more accurate GET map from image by using the integrated color and gradient information in tracking decision making. The new tracker can improve the accuracy and robustness of GTE detection while retains the real-time performance. Compared to other methods, the new tracker can provide perceptual curve features in video frames with low time expense, which makes it an effective solution for analysis and recognition tasks in video surveillance applications.

The rest of this paper is organized as follows. Section 2 presents the concepts of Generic Edge Token model. Sections 3 discusses the details of the new color-based GET tracker. In Section 4, experimental results and analysis are provided. Section 5 gives the conclusions.

2. Generic Edge Token model

Generic Edge Tokens (GETs) are perceptually significant image primitives which represent classes of qualitatively equivalent shape features (Gao & Wong, 1993). A complete set of GETs includes both Generic Segments (GSs) and curve partition points (CPPs). Each **GS** is a perceptually distinguishable edge curve segment with linear or nonlinear feature whereas each **CPP** is some type of junction between GSs.

The classification of GSs is ideally based on the best break-down curves in terms of the perceptual characteristics of GSs. Fig. 1 (a) shows curve partition examples with GSs and CPPs. This method partitions each curve based on the discontinuation of the monotonicity of descriptive characteristics including both direction information and geometry data. Each GS has its own unique descriptive characteristics and represents a general class of qualitatively equivalent curve segments. A generic segment gs is expressed by

$$gs = \{x | p(x)\} \quad (1)$$

where x is an edge point, p indicates some perceptual property, and gs denotes a set of connected points sharing the property p (Zheng & Gao, 2003).

The property p is the monotonic characteristics of GS which can be qualitatively defined by a set of binary functions. Given a segment $y = f(x)$ and its inverse function $x = \varphi(y)$, their first derivatives are represented by $f'(x)$ and $\varphi'(y)$ respectively. The property p of a point x can be fully described by the function set:

$$p(x) = \{f(x), \varphi(y), f'(x), \varphi'(y)\} \quad (2)$$

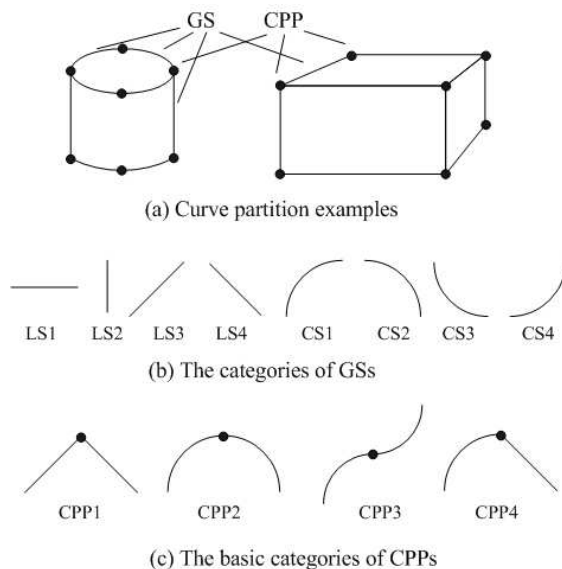


Fig. 1. Generic Edge Token model

GSs can be classified into 8 categories as Fig. 1(b) shows. Each type of GS shares one or more similar properties defined in Equation 2.

The points which break down curves into GSs are the positions on the curves at which the transitions of monotonicity take place. These perceptually significant breaking points, i.e., CPPs, are the general types of joints of GSs. A CPP can group two or more GSs into a perceptual structure. There are 4 basic categories of CPPs, as Fig. 1(c) shows. The CPPs are perceptually stable features and hence very useful for grouping curve structures.

The process of the GET tracker can be described as follows. A selective raster scan is first done to determine strong initial edge pixels, i.e. pixels with high significance, for edge tracking. The significance of a pixel $p(x, y)$, where (x, y) is the pixel location in an image, evaluates the difference between $p(x, y)$ and its neighbor pixels, therefore it can be used to measure the probability of the pixel $p(x, y)$ being an edge pixel.

Beginning from each strong edge pixel, the tracking processes search for the best neighbor edge pixel in order to form a trace. For any edge pixel $p(x, y)$, the next edge pixel on the trace will be picked up as follows. Let $p_n(x, y, d)$ be the neighbor of $p(x, y)$ in direction d , where d is the index of search direction as indicated in Fig. 2. There are eight search directions defined, where each direction is indexed by a number ranged in $[0, 8)$, i.e. east (0), north-east (1), north (2), north-west (3), west (4), south-west (5), south (6), south-east (7). In the curving tracking process, the candidates of the next pixel following $p(x, y)$ on the trace can be expressed by a set of neighbors of $p(x, y)$ in specific tracking direction d :

$$S(p(x, y), d) = \{p_n(x, y, d), p_n(x, y, (d-1) \bmod 8), p_n(x, y, (d+1) \bmod 8)\} \quad (3)$$

The tracking direction d at $p(x, y)$ goes in the same direction as the tangent to the curve at the point (x, y) :

$$d(p(x,y)) = \begin{cases} \lfloor (\arctan(\nabla p_y / \nabla p_x) + 90) / 45 \rfloor & \nabla p_x < 0 \\ 0 & \nabla p_x = 0, \nabla p_y > 0 \\ 0 & \nabla p_x = 0, \nabla p_y < 0 \\ \lfloor (\arctan(\nabla p_y / \nabla p_x) + 270) / 45 \rfloor & \nabla p_x > 0 \end{cases}$$

where $(\nabla p_x, \nabla p_y)$ is the significance of $p(x, y)$ in horizontal and vertical directions respectively. The significance measurement will be discussed in section 3.2.

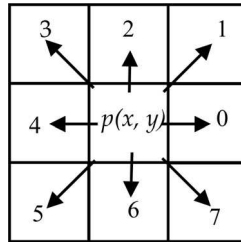


Fig. 2. 8-direction neighbors of a pixel $p(x, y)$. Each direction is labeled by an arrow and indexed by a number

The pixel $p(i, j)$ in $S(p(x, y), d)$ with the most significant is selected as the next edge pixel on the trace:

$$p(i, j) \in S(p(x, y), d) | (\forall (p(i', j') \in S(p(x, y), d), (i', j') \neq (i, j))): \max(\nabla p_x(i, j), \nabla p_y(i, j)) > \max(\nabla p_x(i', j'), \nabla p_y(i', j')) \tag{4}$$

The tracking continues upon reaching an endpoint where no more new edge pixels can be found. If the end point meets the initial point, a closed curve trace is formed; otherwise, this trace is an open trace and the endpoint is one end of the trace. The search of remaining trace pixels begins at the initial point again and proceeds in the opposite direction until the other endpoint of the trace is found. Once a curve has been found, it can be partitioned into GSs, and the type of partitioned GS is determined based on the GET model described in Fig. 1. The tracking processes repeats until all the selected initial edge pixels have been searched. CPPs can be dynamically detected in the edge tracking process.

3. Integrating color and gradient into curve tracking

A GET map provides a rich edge based description of image content including connected edge traces, GS and CPP features. However, in a gradient-based tracker, only gray level properties were applied for determining next best edge pixels in a GET map. The decision may not be sensitive to the situations where pixel candidates have weak gradient, but strong color difference. Since color is always an important factor for human perception of an image, a color-based GET tracker, which integrates color property into edge tracking for better performance of GET extraction, is proposed. Compared to the gradient-based tracker, the color-based tracker produces more accurate GET maps. Fig. 3 shows the processes of color-based edge tracking, which includes noise suppression, initial tracking point selection, and color-based curve tracking.

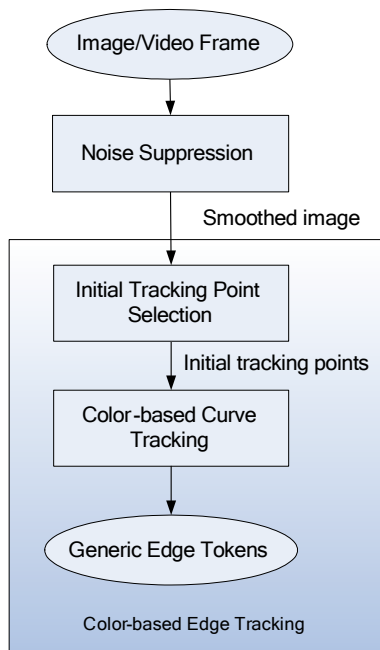


Fig. 3. The architecture of color-based edge tracking

3.1 Noise suppression

Noise suppression during GET tracking can help to reduce the effect of noise in an image during GET detection. Noise suppression is an important step for video surveillance applications. Compared to other images, image sequences in surveillance videos are likely to contain more noises due to the camera noise, light and other surveillance conditions. Before the tracking process starts, a bilateral filter is pre-applied to the image for noise removal to avoid possible jitter edges caused by noise.

Unlike tradition domain filters such as Gaussian filter, **bilateral filter** is a filter that *uses combined domain and range filtering for image smoothing*. It can smooth image noise while preserving most of edge information (Tomasi & Manduchi, 1998). Both domain and range filtering are combined in Bilateral filtering, that is, the spatial distribution of image intensities is considered in the filter. Each pixel value in an image is smoothed by an average of similar (domain) and nearby (range) pixel values, which can be expressed as a normalized weighted average of its neighbors:

$$k(p) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} c(\xi, p) s(f(\xi), f(p)) d\xi \quad (5)$$

where $c(\xi, p)$ means the closeness between p and its neighborhood pixel ξ , and $s(f(\xi), f(p))$ measures the similarity between p and ξ .

Fig. 4 shows the comparison of GET map extraction on images with/without applying the bilateral filter. The bilateral filter efficiently preserves most of the useful edge information and partially removes noise edges.

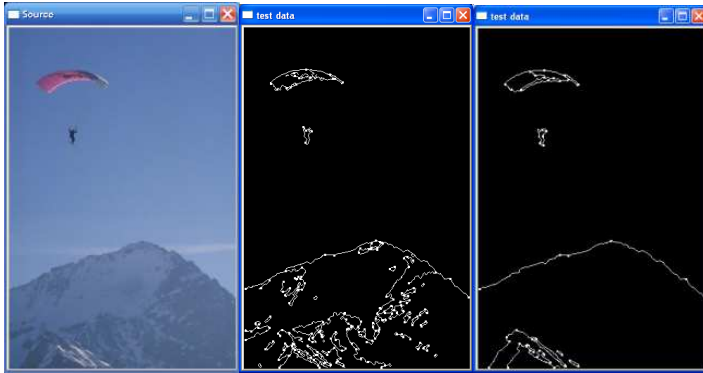


Fig. 4. Comparison of GET map extraction with/without applying bilateral filter. From left to right: A sample image; GET map extraction without bilateral filtering; GET map extraction with bilateral filtering.

3.2 Color based edge decision making

In the color-based GET tracker, integrated color and gradient information is used to determine the significance in the steps of initial edge pixel selection and edge tracking. Although RGB color space is commonly used in color image display, it is not generally appropriate for image representation because of the high correlation between the three color components and non-linearity with human perception. In other words, color differences perceived by human vision cannot be measured by their distance in RGB space. HSI space, which is more intuitive to human vision, is adapted for the color-based tracker. The HSI color space stands for Hue (H), Saturation (S), and Intensity (I), in which the color and intensity information are separated. The Hue value ranges from [0, 360) indicating color information. For example, red has a hue value of 0, green has a value of 120, and blue has a value of 240. The conversion between RGB and HSI can be found in many references (Cheng et al., 2001).

In the gradient-based tracker, the significance of an edge pixel $p(x, y)$ is measured by the gradient magnitude at (x, y) whereas in the color-based tracker, the significance $\nabla p(x, y)$ can be described by the normalized significance of $p(x, y)$ on both color and intensity in HSI color space.

$$\nabla p(x, y) = (\nabla p_x(x, y), \nabla p_y(x, y)) \tag{6}$$

$$\nabla p_x(x, y) = |d_x h(x, y), d_x I(x, y)|, \nabla p_y(x, y) = |d_y h(x, y), d_y I(x, y)|$$

where $(\nabla p_x(x, y), \nabla p_y(x, y))$ is the significance of $p(x, y)$ in horizontal and vertical directions, $h(x, y)$ and $I(x, y)$ indicate the hue and intensity components of $p(x, y)$ respectively, d_x/d_y is the normalized significance in x/y direction.

The significance of intensity can be calculated as

$$d_x I(x, y) = \frac{\partial I(x, y)}{\partial x}, d_y I(x, y) = \frac{\partial I(x, y)}{\partial y} \tag{7}$$

Since HSI space suffers from the hue value discontinuity, i.e. the colors close to hue value of 0 are similar to the colors close to hue value of 360, we define the normalized significance of color as:

$$d_x h(x, y) = \begin{cases} \nabla h_x(x, y) * 256 / 180 & \nabla h_x(x, y) \leq 180 \\ (\nabla h_x(x, y) - 360) * 256 / 180 & \nabla h_x(x, y) > 180 \\ (\nabla h_x(x, y) + 360) * 256 / 180 & \nabla h_x(x, y) < -180 \end{cases}, \quad (8)$$

$$d_y h(x, y) = \begin{cases} \nabla h_y(x, y) * 256 / 180 & \nabla h_y(x, y) \leq 180 \\ (\nabla h_y(x, y) - 360) * 256 / 180 & \nabla h_y(x, y) > 180 \\ (\nabla h_y(x, y) + 360) * 256 / 180 & \nabla h_y(x, y) < -180 \end{cases},$$

$$\nabla h_y(x, y) = \frac{\partial h(x, y)}{\partial y}, \quad \nabla h_x(x, y) = \frac{\partial h(x, y)}{\partial x}$$

Here $dh(x, y)$ is normalized in consistency to $dl(x, y)$, which is ranged in $[-256, 256]$. Specifically, in a gray level image, $d_x h(x, y) = d_y h(x, y) = 0$.

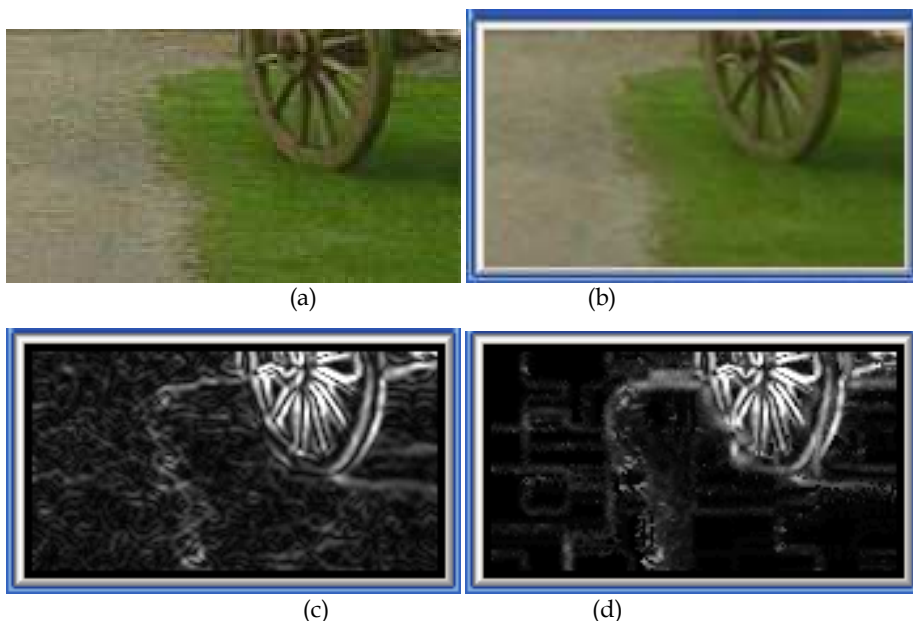


Fig. 5. Comparison of gradient-based and color-based significance map; a) an image; b) image after Bilateral filtering; c) gradient-based significance map ; d) color-based significance map

Fig. 5 shows a comparison between the gradient-based and color-based significance map of a sample within a green grass region. Each significance value is normalized to the range of

[0,255] for display purposes. The higher a pixel's intensity is in the gradient-based or color-based significance map, the greater its gradient-based significance or color-based significance value. The grass region still contains texture after bilateral filtering, as Fig. 5 (b) shows. The gradient-based significance map of the grass and gray ground regions contains many noise pixels due to texture in these areas. The gradient-based significance map in (c) has much more noise than the color-based significance map (d) in the texture areas; and the region boundary edges are more evident in the color-based significance map.

3.3 Color-based curve tracking

The curve tracking of the color-based tracker on an image I can be described as follows:

Step 1. Filtering; apply bilateral filtering to the image I .

Step 2. pick up initial pixels; image I is scanned both vertically and horizontally by pre-defined scan interval δ . Let t_{init} be the significance threshold for initial points, $S(p)$ be the set of initial pixels.

For any scanned pixel $p(x, y)$ do

if $\nabla p(x, y)$ satisfies $\nabla p_x(x, y) > t_{init} \vee \nabla p_y(x, y) > t_{init}$,

$S(p) = S(p) \cup p(x, y)$;

Step 3. Sort each pixel in $S(p)$ by significance descending.

Step 4. Edge tracking; From the first element in $S(p)$, for each initial pixel $p_{start} \in S(p)$ do if p_{start} is not an edge pixel yet, perform the following curve tracking process:

1. Edge tracking starts from p in the initial direction $d_{init} = d(p_{start})$, where $d(p_{start})$ can be calculated using Equation 3.
2. In each step of tracking, if $p(x, y)$ is the current pixel on the trace, a pixel with highest normalized color significance $p(i, j)$ is selected as the possible next edge pixel from a set of candidates, i.e. $S(p(x, y), d(p(x, y)))$, by using Equation 4.
3. Let t_s be the significance threshold for edge pixels:

if $\max(\nabla p_x(i, j), \nabla p_y(i, j)) < t_s$,

one end of the trace has been reached; mark $p(i, j)$ as an endpoint of the trace; tracking goes back to p_{start} and starts tracking in the direction $\text{mod}((d_{init}+4), 8)$ by repeating (2)-(3);

else if $p(i, j)$ is already marked as an edge pixel,

a closed trace is formed. A new tracking starts from a new initial point and repeats (1)-(3).

else

mark $p(i, j)$ as edge pixel, set $p(i, j)$ as current tracking point; go back to (2).

4. The curving tracking process of (1)-(3) repeats until no more initial points are left. GS partition and CPP detection can be dynamically found during the tracking process.

When the color-based tracker is applied to gray-scale images, only intensity information is considered during curve tracking. The tracker is efficient since the strategy of image sample scanning and edge tracking only needs to process a portion of relevant image pixels. Fig. 6 shows an example of edge tracking results by using gradient-based and color-based trackers. Compared to gradient-based tracker, the color-based tracker is able to pick up edges between regions with weak gradient but different colors (see the edge of green grass region). It gives a better edge map with more accurate edges.

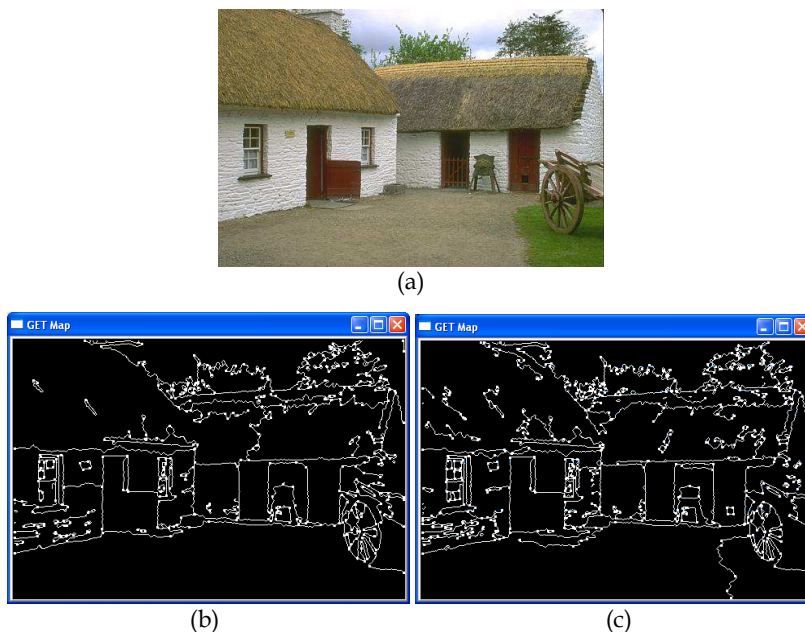


Fig. 6. Edge tracking by using the gradient-based and color-based trackers; a) original image; b) result of edge tracking with gradient only; c) result of edge tracking with integrated color and gradient properties

4. Experiments and evaluation

In this section, the experimental results on both images and video sequencings are presented and analysis. The algorithm of the color-based GET tracker is implemented using C++.

Fig. 7 illustrates output of the color-based GET tracker on images. A comparison with alternative methods on the same set of test images is also provided in our testing. Since the Canny detector is one of the most efficient and commonly used edge detectors, we choose the Canny detector and a contour/curve detection method provided in (Grigorescu et al., 2004) for comparison. This contour/curve detection method is based on the Canny detector and it takes an additional surround suppression step which eliminates texture edges while leaves the contours of objects and region. A Gaussian filter is applied for noise suppression before Canny edge detection. The parameters in both methods are adjusted for each input image to obtain an optimal output edge map. The outputs of GET maps from the color-based GET tracker are also shown in Fig. 7. Bilateral filtering is applied to the GET tracker. Each CPP point is marked in the GET maps.

Fig. 7 shows that, compared to the other two methods, the color-based tracker performs qualitatively good in terms of 1) improving the accuracy of edge; the color-based tracker is able to apply both color and gradient evidence of an image for edge tracking in that the GETs are detected more accurately without increasing computation cost; 2) more robustness in suppressing noise and 3) providing perceptual structures that other methods cannot provide. The color-based tracker can extract GET structures at the same time as tracking, which are semantically sound curve features for image/video analysis.



Fig. 7. For each example: a) original image; b) edge map detected by Canny edge detector; C) curves detected by Canny-based curve detection (Grigorescu et al., 2004); d) GET map detected by the color-based GET tracker.

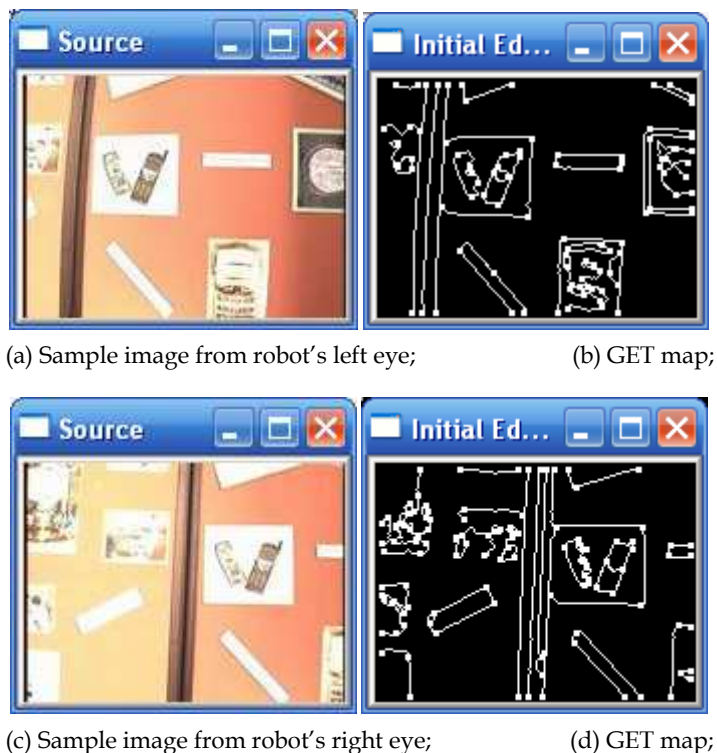


Fig. 8. Robot vision system: edges maps and perceptual features detected by the color-based tracker.

The color-based tracker provides an ideal solution for video applications that requires content analysis and object recognition, such as video surveillance system, robot vision system, etc. The data used for performance testing on video applications come from IPAMI group, Dalhousie University (<http://flame.cs.dal.ca/~IPAMI>), KOGS/IAKS, Universität Karlsruhe (http://i21www.ira.uka.de/image_sequences/) and CAVIAR project (<http://homepages.inf.ed.ac.uk/rbf/CAVIARDATA1/>).

Fig. 8 gives an example of applying the color-based edge tracker to a robot vision system. Two video sequences are taken by robot eyes (two cameras). The size of each video frame is 160×120 pixels, and the frame rate reaches 9 fps. Fig. 8(a) shows an image taken by robot left eye and (c) shows the image with same scene taken by the right robot eye. The GET features detected from each frame will be used for motion analysis, object detection, 3D reconstruction in later stages. The average execution time of GET detection for each frame is 30 milliseconds when the color-based tracker runs under Intel 1.66GHz CPU. It means that the color-based tracker can process each frame in the videos in real-time.

Fig. 9 shows GET maps of moving object detected from surveillance videos. For each frame pair in a video, the GET map is extracted from a frame by the GET tracker, and then moving GETs are used to group moving objects in the video sequence (Chen & Gao, 2008). Each moving object can be perceptually described by the GETs and CPPs extracted by the GET

tracker. In our experiment, the size of each frame in the videos varies from 320 by 240 pixels to 640 by 480 pixels. For a video stream with the frame size of 320 by 240 pixels and the frame rate of 10 frames per second, the average processing time for GET extraction is 50 milliseconds under Intel 1.66GHZ CPU.



Fig. 9. Video surveillance system: GET maps and perceptual features of moving objects detected by the color-based tracker.

The tests above show that the computational cost of the color-based tracker is low enough for real-time applications. The execution time of curving detection and partitioning for an image depends on the number of edge pixels in the image. On average, the execution time for each image/video frame with $160 \times 120 \sim 640 \times 480$ pixels is around 30-70 milliseconds when the color-based tracker runs, which gives the color-based tracker full capability of real-time processing for video applications, such as video content analysis, surveillance system, etc.

5. Conclusions

This paper presents an extended perceptual curve tracker using both color and gradient properties. The system can track edge traces and extract semantic curve features at same time. The enhanced GET tracker provides a more robust and effective solution for edge detection, curve feature extraction with real-time performance. The method has the following technique attributes. 1) It only needs to selectively process a subset of relevant pixels in an image. 2) It detects perceptual curve features without using parameter-based

curve fitting, and the major computation involved is logic inference. 3) It integrates both color and gradient properties into edge decision making so that edges between two color regions with weak gradient can be accurately detected. 4) It improves the robustness in terms of noise handling. The tracker's real-time performance is achieved mainly because of the characteristics of 1) and 2). This edge tracking and curve detection method is very suitable for various real-time applications where edge-based features are needed (Chen et al., 2006) (Reilly & Chen, 2007) (Chen & Gao, 2008).

6. References

- Canny, J. (1986) A Computational Approach to Edge Detection, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 8, No. 6, pp. 679-698
- Chen, H.; Rivait D., & Gao Q. (2006) Real-Time License Plate Identification by Perceptual Shape Grouping and Tracking, *Proceedings of the 9th IEEE Conf. on Intelligent Transportation Systems*, pp. 1352-1357
- Chen, H.; & Gao. Q. (2008) Intelligent Video Analysis for Vehicle Surveillance by Perceptual Edge Feature Grouping, *Computer Vision Research Progress*, Edited by F. Columbus, PA: Nova Science Inc., NY, USA, pp. 83-107
- Cheng, H. D.; Jiang, X. H.; Sun, Y. & Wang, J. L. (2001) Color Image Segmentation: Advances and Prospects, *Pattern Recognition*, Vol. 34, No. 12, pp. 2259-2281
- Cheng, Z.; Chen, M. & Liu, Y. (2004) A robust Algorithm for Image principal Curve Detection, *Pattern Recognition Letters*, Vol. 25, No. 11, pp. 1303-1313
- Fan, J.; Yau, D. K. Y. & Aref, W. G. (2001) Automatic Image Segmentation by Integrating Color-Edge Extraction and Seeded Region Growing, *IEEE Trans. on Image Processing*, Vol. 10, No. 10, pp. 1454-1466
- Gao, Q. & Wong, A. (1993) Curve Detection based on Perceptual Organization. *Pattern Recognition*, Vol. 26, No. 1, pp. 1039-1046
- Grigorescu, C.; Petkov, N. & Westenberg, M. A. (2004) Contour and Boundary Detection improved by Surround Suppression of Texture Edges, *Image and Vision Computing*, Vol. 22, No. 8, pp. 609-622
- Olson, C. F. (1999) Constrained Hough Transforms for Curve Detection, *Computer Vision and Image Understanding*, Vol. 73, No. 3, pp. 329-345
- Pei, S. C. & Horng, J. H. (1995) Fitting Digital Curves using Circular Arcs, *Pattern Recognition*, Vol. 28, No. 1, pp. 107-116
- Perez, F. & Koch, C. (1994) Toward Color Image Segmentation in Analog VLSI: Algorithm and Hardware, *International Journal of Computer Vision*, Vol. 12, No. 1, pp. 17-42
- Qian, R. J. & Huang, T. S. (1996) Optimal Edge Detection in two-Dimensional Images, *IEEE Trans. Image Processing*, Vol. 5, pp. 1215-1220
- Ruzon, M. A. & Tomasi, C. (1999) Color-Edge Detection with the Compass Operator, *Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 160-166
- Reilly, D. & Chen, H. (2007) Toward fluid, mobile and ubiquitous interaction with paper using recursive 2D barcodes, *Proceedings of the 3rd International workshop on Pervasive Mobile Interaction Devices*, pp. 20-23
- Tomasi, C. & Manduchi, R. (1998) Bilateral Filtering for Gray and Color Images, *Proceedings of the IEEE International Conf. on Computer Vision*, pp. 839-847

- Yip, R.K.K.; Tam, P.K.S. & Leung, D.N.K. (1992) Modification of Hough Transform for Circles and Ellipses Detection using a 2-Dimension Array, *Pattern Recognition*, Vol. 25, No. 9, pp. 1007-1022
- Zheng, X. & Gao, Q. (2003) Generic Edge Tokens, Representation, Segmentation and Grouping, *Proceedings of the 16th International Conf. on Vision Interface*, pp. 388-394



Video Surveillance

Edited by Prof. Weiyao Lin

ISBN 978-953-307-436-8

Hard cover, 486 pages

Publisher InTech

Published online 03, February, 2011

Published in print edition February, 2011

This book presents the latest achievements and developments in the field of video surveillance. The chapters selected for this book comprise a cross-section of topics that reflect a variety of perspectives and disciplinary backgrounds. Besides the introduction of new achievements in video surveillance, this book also presents some good overviews of the state-of-the-art technologies as well as some interesting advanced topics related to video surveillance. Summing up the wide range of issues presented in the book, it can be addressed to a quite broad audience, including both academic researchers and practitioners in halls of industries interested in scheduling theory and its applications. I believe this book can provide a clear picture of the current research status in the area of video surveillance and can also encourage the development of new achievements in this field.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Huiqiong Chen and Qigang Gao (2011). Integrating Color and Gradient into Real-Time Curve Tracking and Feature Extraction for Video Surveillance, Video Surveillance, Prof. Weiyao Lin (Ed.), ISBN: 978-953-307-436-8, InTech, Available from: <http://www.intechopen.com/books/video-surveillance/integrating-color-and-gradient-into-real-time-curve-tracking-and-feature-extraction-for-video-survei>

INTECH

open science | open minds

InTech Europe

University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China

Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

© 2011 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the [Creative Commons Attribution-NonCommercial-ShareAlike-3.0 License](#), which permits use, distribution and reproduction for non-commercial purposes, provided the original is properly cited and derivative works building on this content are distributed under the same license.