在靜態場景中追蹤多個物體演算法

Multiple objects tracking in the video sequence with static scene

研究生：陳峻儀
指導教授：蔡文錦 教授

中華民國 九十六年六月
在靜態場景中追蹤多個物體演算法
Multiple objects tracking in the video sequence with static scene

研究生：陳峻儀
Student：Jun-Yi Chen

指導教授：蔡文錦
Advisor：Wen-Jiin Tsai
在靜態場景中追蹤多個物體演算法

學生：陳峻儀 ............................指導教授：蔡文錦

國立交通大學資訊科學與工程研究所碩士班

摘要

追蹤物體在智慧型監控系統方面是一個受矚目的議題, 如何在發生事故可以立即得知，並給予及時的幫助。這篇論文提出一個混合式追蹤物體的方法，用來追蹤物體的資訊包括有物體的輪廓、顏色、移動的區域性，會分別產生三個的物體相似度，以及會利用我們提出的相對應的演算法把這三個相似度整合起來進行物體的追蹤。在遮蔽物體方面，我們储存物體的輪廓資訊，最後利用這些資訊來作為分離相連物體的依據。本篇論文可以解決物體追蹤的問題包括剛性和非剛性物體的出現、消失、分裂、合併、遮蔽現象於場景中。

關鍵字：混合式物體追蹤，遮蔽，霍式轉換，分水嶺。
Multiple objects tracking in the video sequence with static scene

Student: Jun-Yi Chen  Dvisor: Wen-Jiin Tsai

Department of Computer Science
National Chiao-Tung University

ABSTRACT

In the recent years, there have been significant developments in the field of surveillance systems, where object tracking is a key technology. This thesis proposes a new hybrid object tracking method which combines region, edge and location-based methods in the algorithms. For region based method, we use watershed to segment objects into several regions. For edge based method, we use Hough Transform to transform edge from image domain to parameter domain for similarity comparison. The location based method is applied only when both region-based and edge-based methods can’t find the corresponding objects.

The experimental result shows that the proposed algorithm can track multiple objects in a video sequence with object appearance and disappearance, non-rigid and rigid movements, object splitting and merged as well as object occlusion. The success of tracking rate can be up to 97.9% for video sequence with static scene.

Keyword: hybrid object tracking, occlusion, Hough Transform, watershed.
Contents

抽象 ................................................................................................................................. 3

ABSTRACT .................................................................................................................... 4

Chapter 1 Introduction ............................................................................................. 9

Chapter 2 Related work ............................................................................................ 12
  2.1 Object Tracking ................................................................................................. 12
  2.2 Occlusion .......................................................................................................... 15

Chapter 3 Object Segmentation ............................................................................... 16

Chapter 4 The Proposed Object Tracking Method .................................................. 19
  4.1 Region-based method ....................................................................................... 20
  4.2 Edge-based method ........................................................................................... 24
  4.3 Location-based method ..................................................................................... 28
  4.4 Object tracking .................................................................................................. 30
  4.5 Occlusion .......................................................................................................... 31
    4.5.1 Key frame .................................................................................................... 36

Chapter 5 Speed up tracking rate ........................................................................... 38

Chapter 6 Experimental Results and Discussion ..................................................... 39
  6.1 Tracking success rate ....................................................................................... 42
  6.2 Tracking rate speedup ...................................................................................... 44
  6.3 Conclusion ......................................................................................................... 46

Appendix A Morphology ......................................................................................... 47

Appendix B Hough transform .................................................................................. 50

Bibliography ................................................................................................................ 55
List of Figures

**FIGURE 1:** EXAMPLE OF 2D AND 3D MODEL, GET FROM [1] FIG 4 AND [2] FIG 5 ..............................................13

**FIGURE 2:** EXAMPLE OF HUMAN CONNECTED REGIONS, GET FROM [3] FIG 1 .................................................13

**FIGURE 3:** EXAMPLE OF HUMAN CONTOUR TRACKING, GET FROM [4] FIG 4 .....................................................14

**FIGURE 4:** GET FROM [5] FIG 2 、 FIG3 .....................................................................................................14

**FIGURE 5:** MULTIPLE LAYERS OF DEPTH MAP [24]. ..................................................................................15

**FIGURE 6:** OBJECT SEGMENTATION FLOWCHART. ........................................................................................17

**FIGURE 7:** OBJECT SEGMENTATION EXAMPLE ............................................................................................18

**FIGURE 8:** HYBRID OBJECT TRACKING FLOWCHART ..................................................................................19

**FIGURE 9:** REGIONS FROM WATERSHED ALGORITHM ..................................................................................20

**FIGURE 10:** SIMILARITIES BETWEEN TWO FRAMES ..........................................................................................22

**FIGURE 11:** GENERALIZED HOUGH TRANSFORM [21]. ......................................................................................25

**FIGURE 12:** R TABLE [21] ..................................................................................................................................25

**FIGURE 13:** CANNY EDGE ...............................................................................................................................26

**FIGURE 14:** BINARY IMAGES OF R TABLE USING HOUGH TRANSFORM ..........................................................27

**FIGURE 15:** OBJECT SPLIT AND COMBINE ........................................................................................................28

**FIGURE 16:** DIFFERENT OBJECTS WITH DIFFERENT COLOR RECTANGLES ......................................................31

**FIGURE 17:** SHOWS START AND END OF OCCLUSION .....................................................................................32

**FIGURE 18:** OCCLUSION PROCESSING .............................................................................................................34

**FIGURE 19:** SHOW BINARY IMAGE BY EDGE-BASED METHOD ........................................................................35

**FIGURE 20:** RESULT PROCESS OF OCCLUSION HANDLING ................................................................................35

**FIGURE 21:** MULTIPLE KEY FRAMES OF RED BLOCK OBJECT .............................................................................36

**FIGURE 22:** USING SUCCESSIVE FOUR KEY FRAMES BEFORE OCCLUSION ......................................................37

**FIGURE 23:** USING KEY FRAMES BUFFER BEFORE OCCLUSION ........................................................................37

**FIGURE 24:** SCALED VIDEO FRAME ..................................................................................................................38

**FIGURE 25:** MULTIPLE OBJECTS TRACKING RESULTS .......................................................................................41

**FIGURE 26:** RATE OF ACCURACY IN VIDEO SEQUENCES ...............................................................................42

**FIGURE 27:** RATE OF ACCURACY IN VIDEO SEQUENCES ...............................................................................43

**FIGURE 28:** 320*240 VS 80*60 TRACKING RATE ...............................................................................................44

**FIGURE 29:** AVERAGE OF REGIONS / FRAME, EDGES / FRAME AND FRAMES / SECOND ..............................45

**FIGURE 30:** 3x3 DILATION OPERATION ................................................................................................................47

**FIGURE 31:** RESULT OF DILATION ......................................................................................................................48

**FIGURE 32:** TECHNOLOGICAL PROCESSES OF DILATION ..............................................................................48

**FIGURE 33:** 3×3 EROSION OPERATION ................................................................................................................49

**FIGURE 34:** RESULT OF EROSION .....................................................................................................................49

**FIGURE 35:** TECHNOLOGICAL PROCESSES OF EROSION ..............................................................................49

**FIGURE 36:** GET FROM [21] ............................................................................................................................51
List of Tables

TABLE 1: EDGE SIMILARITIES. ................................................................. 27
TABLE 2: EXPERIMENTAL THRESHOLDS .................................................. 39
Chapter 1  Introduction

In recent years, there have been significant developments in the field of tracking object systems. There are many approaches of object tracking proposed (e.g., [1, 3, 4, 5, 12]). Object tracking is usually applied to surveillance system, like accident of 911 happened in U.S.A and terrific case distributed all around the world. It is important to build camera to keep under surveillance in intersections, ports and airports…etc. Although it is more helpful to build camera to keep under surveillance for security, people must keep a close watch on monitor in control center. This method wastes manpower and cost expensive. A traditional surveillance system takes video sequence and stores it directly without further processing. An intelligent surveillance system makes up drawbacks of tradition surveillance system. It can uses software not only analyze images but also monitor and promulgate in real time. Therefore, identifying object to track that happened in the real time becomes a more important issue.

An intelligent surveillance system consists of following technologies

1. Object Segmentation: To detect dubious objects and segment it in the scene.

A simple way of object segmentation is to remove background from frame to keep objects, A sophisticated object segmentation method includes the remove of global or local luminance changing, shadow of object, dynamic background…etc.
2. Object Tracking: To identify objects and keep track of them in the video sequence.

An intelligent surveillance system can track objects by color, motion, edge, texture and models…etc. The major issues in object tracking is how to identify similarity of object and occlusion [15, 17, 18] in successive frames.

3. Object Classification: To classify what kind of objects appearing in the video sequence.

The objects to be identified would be human, car, vehicles…etc. Geometry method and affine transform are the typical ways to formulate shape change and deformation.

4. Behavior Recognition: To judge the behavior of objects in the video sequence.

It is hardest in intelligent surveillance system, because it must use object segmentation, tracking and classification to judge behavior of objects.

In this thesis, we focus on the subject of object tracking and propose a hybrid method which exploits region, edge and location in the algorithm for object tracking and occlusion handling. Using region information can tolerate large object deformation if color distribution on the object remains similarly. Using edge information can tolerate color change (for example, self-occlusion occurs) on the object if object contour remains similarly. Location information is based on the assumption that object motion should keep in a reasonable range of speed (can’t disappear suddenly). The experimental results show that the success rate can be up
to 98% if all these information are included in the tracking and occlusion handling. In order to speed up tracking rate, an acceleration method which can fast 21 times than original frame size of the tracking time and still keep the success rate. This thesis is organized as follows. Chapter 2 describes various object tracking methods and discuss their proposed conceptions, respectively, In chapter 3 is about object segmentation method. In chapter 4, a hybrid tracking method which exploits region, edge and location-based technologies is proposed. Occlusion handling method is also proposed in this section. Chapter 5, proposed an enhance tracking rate method. Finally, chapter 6 presents some experiment results and discussion.
Chapter 2  Related work

2.1 Object Tracking

Object tracking plays an important role in an intelligent surveillance system. Several kinds of methods have been proposed successively, like

1. Model-based.
2. Appearance-based.
3. Feature-based.
5. Hybrid-based.

Model-based algorithms are based on the prior knowledge of the objects. It must build up models of object first, and then use those models to track objects. A precise object tracking can be made if a complete, correct object model can be built up first. The drawback with model-based algorithm is that it can not track objects that are not associated with any model in its database. High computational complexity is also a common problem.[1, 2, 7].
Appearance-based algorithms track objects, by splitting connected pixels into many regions based on motion, color, texture…etc. The drawback with it is that it can’t deal with occlusion between objects and complex deformation. [3, 6, 7, 10, 14, 15, 17, 19].

Feature-based algorithms track objects, by extracting characteristic elements from object, in order to make comparison with others. The features used can be divided into two parts, one is global feature-based which is made up with color, area and barycenter; the other is local feature-based which is make up with line and apex. The drawback with it is that it is hard to identify objects that have same features [5].
Contour-based algorithms track objects, by monitoring the contour of object and updating those contours dynamically in successive frames. The contour can be made up with 2D or 3D mesh and edge. The drawback with it is that it can’t deal with large deformation and the object that is partially occluded. High computational complexity is also a common problem with those algorithms.[4, 8, 9, 11].

Hybrid-based algorithms are usually designed as a hybrid between region-based and feature-based, It first gets all regions of an object and then track those regions, by using features like color, motion, texture…etc. The drawback with it is that high computational complexity [12, 20].
2.2 Occlusion

Generally in tracking object, it is hard to identify same objects when occlusion happened, even if whole object is covered by other objects. In the field of occlusion, it can be divided into three classes, like self occlusion, partial occlusion and total occlusion. This thesis with reference to self occlusion and partial occlusion can be overcame by region-based and edge-based method, as for total occlusion with connected objects, we use edge-based method to split connected objects.

Before new edge-based method to solve total occlusion with connect objects, let us to understand which methods are usually used for solving this problem. In recent years, people want to break through traditional tracking technology, they use many cameras to shoot object from all directions, and an object has its characteristics on every angle, we use those characteristics on every angle to reform an intact object. This is a technology of multi-view. Because human has parallax, so it creates relation of depth in the scene, it usually named depth map in the field of multi-view [22, 23, 24]. As an example, figure 5 shows the depth map for each object in a frame. With depth map technology, object occlusion can be easily handled.

Figure 5: Multiple layers of depth map [24].
Chapter 3  Object Segmentation

Before going into object tracking discussion, the “objects” must be segmented from scene first. This thesis uses background subtraction method to segment objects from scene. The first frame captured by camera is chosen as background at first. And then the background will be updated with a succeeding frame if that frame contains no object. A frame is said to be containing objects if something left after background subtraction and noise removing. The equation (1) below is used for noise removing. Let P(N,i) denote the luminance value of ith pixel in current frame N, and B denote background frame.

\[
\text{If}(\text{abs}(P(N,i) - P(B,i)) \leq \text{Th1})

P(N,i)=0;
\]

End ................................................................. (1)

where i=1, 2, …, n. n is the number of total pixel in current frame N, and Th1 is threshold. However since background subtraction may fracture objects and produce many small holes, we use morphological dilation and erosion operators to combine these fragments and fill holes of objects. In morphology, dilation operation on a binary image is used to gradually enlarge the boundaries of regions of foreground pixels and the erosion operation is used to erode away the boundaries of regions of foreground pixels [See appendix A].
Figure 6 shows the object segmentation flowchart and figure 7 shows the example, where figure 7 (a) shows background frame, figure 7 (b) shows a test frame of video sequence, figure 7 (c) shows the test frame after background subtraction, figure 7 (d) shows its binary image after noise removing, figure 7 (e) shows the binary image after dilation and figure 7 (f) shows the binary image after erosion.

![Figure 6: object segmentation flowchart.](image)
<table>
<thead>
<tr>
<th>After background subtraction</th>
<th>After noise removing</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="c.png" alt="Image" /></td>
<td><img src="d.png" alt="Image" /></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>After Morphological dilation</th>
<th>After Morphological erosion</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="e.png" alt="Image" /></td>
<td><img src="f.png" alt="Image" /></td>
</tr>
</tbody>
</table>

Figure 7: object segmentation example.
Chapter 4  The Proposed Object Tracking Method

In this section, an object tracking method is proposed, which we use region, edge and location information to check object similarity, and then occlusion detection and handling method is employed to identify objects more precisely. The proposed framework is depicted in Figure 8, where the measure of region-based similarity (S1), edge-based similarity (S2), and the location-based similarity (S3) are described in sections 4.1, 4.2 and 4.3, respectively. The occlusion detection and handling method is described in section 4.4.

Figure 8: Hybrid object tracking flowchart.
4.1 Region-based method

For the region-based method, first we need to split object into several regions and then check the similarity between regions in two successive frames. In this thesis, the watershed method is used to split object into several regions. The idea behind the watershed method is to emulate the structure of watershed to segment an image. In fact we can imagine a scene as mountains with lakes, and the watershed is a boundary to split mountains and lakes, as well as lakes and lakes. In order to get catchment basin of each lake, when water flooded into lower part of surface, before any two lakes mixed, we would build a watershed to separate them. The process proceeds until water is immersed up to the highest of mountain. In this thesis, the gray levels of pixels are used to represent the depth of lakes or the height of mountains. This approach has been addressed in [6], and the example is given below in figure 9.

<table>
<thead>
<tr>
<th>Original image</th>
<th>Result image</th>
</tr>
</thead>
</table>

Figure 9: Regions from watershed algorithm.
After applying watershed method to split objects onto regions, color representative is calculated for each region. The color representative is a color consisting of dominant color components in a region. Assume a region consisting of six pixels below (81, 89, 92), (79, 89, 91), (81, 86, 94), (81, 86, 92), (83, 89, 90) and (80, 87, 92) where the three values in each parenthesis represent the Red, Green and Blue components of the pixels. In this case, 81 is dominant in Red component, 89 in Green component, and 92 in Blue component. Therefore, (81, 89, 92) is selected as the color representative of this region.

For each region frame N+1, we search frame N to find regions with color representative close to it. In stead of best match, close match is used in the search process. Two color representatives are said to be similarly if their color distance is less than a given threshold. The color distance between two color representatives is calculated as follows.

\[
\begin{align*}
\text{abs}(R1-R2) + \text{abs}(B1-B2) + \text{abs}(G1-G2) & \leq \delta \\
\sqrt{((R1-R2)^2 + (G1-G2)^2 + (B1-B2)^2)} & \leq \delta \\
\end{align*}
\]

To reduce the computation, a search range is defined on frame N such that only those regions inside the search range will be checked for the close match. The size and location of search range in frame N depends on the current region in frame N+1. After search process is done, each region in frame N+1 might be mapped to several regions (due to close match) in frame N. We use formula (3) to get object similarity denoted by (S1)
between two objects (object i in frame N+1 and object j in the frame N).

\[
S1[(n+1,i) , (n,j)] = \frac{O(n+1,i) \rightarrow O(n,j)}{O(n+1,i)} \tag{3}
\]

\(O(n , j)\) means total regions of object j, \(O(n+1 , i)\) mean total regions of object i and \(O(n+1 , i) \rightarrow O(n , j)\) means the number of regions in object i that are close match to object j.

Assume there are 17 regions in object 2 of frame N+1, and 18 regions in object 2 of frame N, then we have \(O(n+1,2)=17\) and \(O(n,2)=18\). If there are 13 regions in object 2 of frame N+1 that are close match with object 2 in frame N, then we have similarity between these two objects as \(13/17=0.7647\). Assume both frame N and N+1 have three objects, the similarity between every two objects need to be calculated. Figure 10 gives an example of similarity between two frames using proposed region-based method.

<table>
<thead>
<tr>
<th>Object Similarity ((S1))</th>
<th>Object 1</th>
<th>Object 2</th>
<th>Object 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Object 1</td>
<td>1.0000</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Object 2</td>
<td>0</td>
<td>0.7647</td>
<td>0</td>
</tr>
<tr>
<td>Object 3</td>
<td>0</td>
<td>0</td>
<td>0.5102</td>
</tr>
</tbody>
</table>

Figure 10: Similarities between two frames.
The region-based algorithm is summarized as follows.

(1) use watershed method to turn objects in frames N and N+1 into regions.
(2) Calculate color representative for each region.
(3) Define search range in the frame N for each region in frame N+1.
(4) In the search range, perform close match for each region in frame N+1.
(5) Measure object similarity by using formula {3}
4.2 Edge-based method

For edge-based method, we first apply Hough Transform on each pixel of the object edges to convert them from image domain to parameter domain. And then, in the parameter domain, a similarity check is performed for object tracking. There are several ways to transform image information from pixel domain to parameter domain. Affine Transform, for example, a widely used method in edge-based object tracking. However, due to its high complexity (with 6 unknown parameters) and restriction to rigid object only, we don’t apply it. The idea behind using Hough Transform in this thesis is based on the observation that two objects with little deformation (e.g. due to scaling and rotation) in the image domain can produce two highly correlation R tables in the parameter domain, which are good for object tracking. The Hough transform is a technique which is originally developed to isolate features of a particular shape within an image. Because it requires that the desired features be specified in some parametric form, the classical Hough transform is most commonly used for the detection of regular curves such as lines, circles, ellipses, etc. A generalized Hough transform can be employed in applications where a simple analytic description of a feature(s) is not possible. The main advantage of the Hough transform technique is that it is tolerant of gaps in feature boundary descriptions and is relatively unaffected by image noise. [See appendix B]. The Generalized Hough Transform (GHT) is defined as follows:
1. Give a characteristic point (like barycenter).

2. Build a empty R table, and its index is $\phi_i$, i=1, 2, M, angle increased $\pi/M$ from 0 to 180.

3. Aim at each point $(x, y)$ of edge, calculating $(r, \alpha)$
   
   \[
   r = \sqrt{(x - xc)^2 + (y - yc)^2}.
   \]
   
   \[
   \alpha = \tan^{-1}\left(\frac{y - yc}{x - xc}\right).
   \]

4. Calculate $\phi_i$, then put $(r, \alpha)$ into R table in which belong to $\phi_i$.

5. Repeat 4, until detected over of all points of edge, and it will get R table

\[
\begin{align*}
\phi_1: & \quad (r_1^1, \alpha_1^1), (r_2^1, \alpha_2^1), \ldots \\
\phi_2: & \quad (r_1^2, \alpha_1^2), (r_2^2, \alpha_2^2), \ldots \\
\phi_n: & \quad (r_1^n, \alpha_1^n), (r_2^n, \alpha_2^n), \ldots
\end{align*}
\]

Before using Hough Transform, we first turn each object into edges.
A lot of algorithms can be applied, like canny, sobel, … etc. Figure 13 shows the example of canny edge method.
Hough Transform takes object edges as input to produce a R table for each object. We then turn the R table into a two-dimension binary image and use the formula (4) to measure the edge similarity of the two objects.

\[
S2[(n+1,j), (n,i)] = \frac{E(n+1,j) \& E(n,i)}{E(n+1,j) | E(n,i)} \tag{4}
\]

\(E(n, i)\) means binary image of R table of object i on frame n, \(E(n+1, j)\) means that of object j on frame n+1, \(E(n, i) \& E(n+1, j)\) means the two binary images are calculated by “AND” operation, and \(E(n, i) \mid E(n+1, j)\) calculated by “OR” operation.

The edge-based method is summarized as follows:

1. Turn objects into edges.
2. Get R table of each object by Hough Transform.
3. Turn R table into a two-dimension binary image.
4. Apply formula (4) to compute the edge similarity between two
(5) Repeat (1) ~ (4) until all objects in frame N+1 are calculated. 

(6) Gets each max edge similarity value of object i in frame N+1.

<table>
<thead>
<tr>
<th>Frame n :</th>
<th>E(n,1)</th>
<th>E(n,2)</th>
<th>E(n,3)</th>
</tr>
</thead>
<tbody>
<tr>
<td>E(n,1) ~ E(n,x)</td>
<td><img src="binary1.png" alt="Image" /></td>
<td><img src="binary2.png" alt="Image" /></td>
<td><img src="binary3.png" alt="Image" /></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Frame n+1 :</th>
<th>E(n+1,1)</th>
<th>E(n+1,2)</th>
<th>E(n+1,3)</th>
</tr>
</thead>
<tbody>
<tr>
<td>E(n+1,1) ~ E(n+1,y)</td>
<td><img src="binary1.png" alt="Image" /></td>
<td><img src="binary2.png" alt="Image" /></td>
<td><img src="binary3.png" alt="Image" /></td>
</tr>
</tbody>
</table>

Figure 14: Binary images of R table using Hough transform.

<table>
<thead>
<tr>
<th>Edge_similarity(S2)</th>
<th>E(n,1)</th>
<th>E(n,2)</th>
<th>E(n,3)</th>
</tr>
</thead>
<tbody>
<tr>
<td>E(n+1,1)</td>
<td>1.0000</td>
<td>0.0936</td>
<td>0.1163</td>
</tr>
<tr>
<td>E(n+1,2)</td>
<td>0.2697</td>
<td>0.8073</td>
<td>0.4153</td>
</tr>
<tr>
<td>E(n+1,3)</td>
<td>0.2652</td>
<td>0.4320</td>
<td>0.7857</td>
</tr>
</tbody>
</table>

Table 1: Edge similarities.

Figure 14 and shows the binary images of R tables after applying Hough transform, and Table 1 lists the measure of edge similarities between objects.
4.3 Location-based method

The location-based method is based on the assumption that object movement should be in a reasonable range of speed. That is, objects appearing in frame N wouldn’t disappear suddenly in frame N+1. Based on this assumption, if both region-based and edge-based tracking methods can’t find the corresponding object, the objects in similar location on two successive frames will be considered as the same object. The formula (5) below is for location-based similarity measurement.

\[
S_3[(n+1, j), (n, i)] = \begin{cases} 
1 & \text{if abs}(B(n, i)-B(n+1, j)) \leq \text{Th5}.

0 & \text{Otherwise}.
\end{cases}
\]

\(B(n, i)\) means barycenter of object \(i\) in frame \(n\), \(B(n+1, j)\) = barycenter of object \(j\) in frame \(n+1\) and \(S_3[(n+1, j) \rightarrow (n, i)] = 1\) means movement in a reasonable range. Figure 15 shows an example for it, where \(B(n,i)\) denotes the barycenter of object in frame \(n\), and Location Merge Threshold = Th5.

Figure 15: Object split and combine.
(I) \( A \rightarrow B \) shows object breakup because of abs \( (B(n,i) - B(n+1,j)) \leq Th5 \).

(II) \( B \rightarrow C \) shows new object appearance because of abs \( (B(n,i) - B(n+1,j)) > Th5 \).

(III) \( C \rightarrow D \) shows object merge because of abs \( (B(n,i) - B(n+1,j)) \leq Th5 \) and \( F2_3.LB(n-t,m) = LB(n,m) \).

\( LB(n-t,m) = LB(n,m) \) is a condition of occlusion, which will be introduced in the next section.
### 4.4 Object tracking

After describing region-based, edge-based and location-based methods, we will get object similarity measures: $S_1$, $S_2$ and $S_3$, respectively. Our proposed method then explores these measures as below to do similarity check between why two objects $i$ and $j$, where object $i$ can be any object in frame $n$, and object $j$ in frame $n+1$

$$S_{ij} = W_1*S_{1ij} + W_2*S_{2ij} + W_3*S_{3ij}.$$ 

$$F(S_{ij}) = \begin{cases} 
\text{Object found (tracked)} & \text{if } S_{ij} > Th_4, \text{ where } W_1=W_2=1/2 \text{ and } W_3=0 \\
\text{New object} & \text{if } S_{ij} < Th_4, \text{ where } W_1=W_2=W_3=1/3. 
\end{cases}$$

The $F(S_{ij})$ represents the tracking result between object $i$ in frame $n$ and object $j$ in frame $n+1$. The object $j$ in frame $n+1$ is recognized to be the object $i$ in frame $n$ if the similarity between them is high (ie. $S_{ij} > Th_4$); otherwise, object $j$ in frame $n+1$ will be regarded as a new object. For the case of object found, three different weighting for $W_1$, $W_2$ and $W_3$ are used, where $W_1=W_2=1/2$ and $W_3=0$ means that if similarity measure using region-based and edge-based are high enough, then no need to consider the measure using location-based method; otherwise all the three measures $S_1$, $S_2$ and $S_3$ will be taken into consideration with equal weighting.
4.5 Occlusion

We will use edge-based method to distinguish different objects when they are in occlusion. There is a common issue in solving occlusion, that is, how to know that objects are in occlusion or not? The problem arises because the connected pixels can be a single object, or can be multiple objects in occlusion. Here we first present a method to detect the occurrence of object occlusion, and then an occlusion handling method is proposed to distinguish objects in occlusion so that each object can be identified and marked separately. We will put different objects into different rectangles associated with different colors as shown in Figure 16.

![Figure 16: Different objects with different color rectangles.](image)
Figure 17: shows start and end of occlusion

To detect the starting point of occlusion between objects, the x position of the object bartcenter is used. For the objects in figure 16, we have the x positions of barycenter for the objects in Blue and Red rectangles, respectively, shown in figure 17 (a), (b) and (c), there are both suddenly disappearance of barycenter for the object in Blue rectangle and Red rectangle, because Red rectangle is the best similar than Blue rectangle with White rectangle (at frame 149), so a suddenly change of barycenter position for the object in Red rectangle to White rectangle, at frame 149. From the corresponding frames in figure 16 (a) and (b). We Observe that the disappearance as well as the sudden change of barycenter position is due to “occlusion”, which results in a single connected object in White rectangle.

Let $X(n,i)$ denote the x position of barycenter for an object $i$ at frame
n. The conditions for detecting the start of object occlusion are

1. \( X(n,i) \neq 0, X(n,i+1) \neq 0 \) and \( X(n+1,i), X(n+1,i+1) \) does not exit.
2. \( \text{Th6} \leq \text{Abs}(X(n,i) - X(n+1,j)) \leq 2\text{Th6} \)

where the second condition also stands for the condition of occlusion termination. As an example in figure 16 (d) and (e), when occlusion terminates, the x position of barycenter for the object (object similarity is the best one) will have large change suddenly because the single connected range is separated into two objects. The corresponsive x position of barycenter is shown at frame 176 in the figure 17 (c). After describing how to detect the start and the end of the occlusion, we now illustrate how to process occlusion so that individual object in the single connected image can be identified as shown in figure 16 (c) and (f).

The following is the occlusion processing algorithm.

1. For the connect objects in occlusion, cut it into multiple pieces of rectangular regions according to size of object.
2. Get binary image of Hough Transform for each rectangular region.
3. Calculate edge similarity between each object in preview frame and those rectangular regions.
4. If object is distinguished, choose the best rectangular region that has largest edge similarity.
5. Repeat (2)–(4) until all the connected objects are separated.
As an example, figure 18 (a) shows the connected object, figure 18 (b) shows edge image of the connected object, figure 18 (c) shows the first piece of rectangular region (X1~X1.1) for similarity check, figure 18 (d) shows the second piece of rectangular region (X1~X1.2) and the third as well as the fourth pieces are shown in figure 18 (e) ~ (f), respectively. Figure 20 (b) shows the binary image of R-table for the first three pieces of rectangular regions (starting from X1); and figure 20 (a) shows that for the first four piece (starting from X2). Given the binary images of R table for two objects in key frames as shown in figure 19 (c) and (d), we obtain the piece of rectangular ranging from X1 to X1.3 to be object Blue because the similarity between them is highest; while the one ranging from X2 to X1.2 to be object Red.

Figure 18: occlusion processing
Figure 19: show binary image by edge-based method.

Figure 20: Result process of occlusion handling.
4.5.1 Key frame

The key frame is a selected frame where the object on it will be used for similarity check when in occlusion. Using two-levels key frame buffer to store key frames based on edge-based similarity $R_2$ in successive two frame. Since each level of frame buffer is limited, the latest frame that meets the edge-based similarity $R_2$ with replace the earliest one in the same level of frame buffer. The $R_2$ for each of the two-level frame buffer is defined as follows.

(1). $0.9 \leq R_2 \leq 1$

The first level frame buffer stores objects with small deformation (therefore, the $R_2$ for successive two frames is high).

(2). $0.5 \leq R_2 < 0.9$

The second level frame buffer stores objects with large deformation, for example, people turn to other direction, catch something or drop something suddenly.

<table>
<thead>
<tr>
<th>Level 1</th>
<th>Level 1</th>
<th>Level 2</th>
<th>Level 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.9894</td>
<td>0.96391</td>
<td>0.8935</td>
<td>0.99409</td>
</tr>
</tbody>
</table>

Figure 21: Multiple key frames of red block object

In general, the key frame update rate for the first level is faster than that in the second level because object deformation is small in successive
frames. Separating the key frame buffer into two levels is to keep the frame object with large deformation from being replaced with frame object with small deformation. Using multiple-level key frames is good for occlusion processing as the example shown in figure 22 and figure 23 where figure 22 shows tracking results without multiple key frames during occlusion (using the four frames before occlusion), figure 23 shows tracking results with two-level key frames during occlusion.

Figure 22: Using successive four key frames before occlusion.

Figure 23: Using key frames buffer before occlusion.
Chapter 5  Speed up tracking rate

The section presents result of speed up tracking rate. First, video sequence has been scaled down to 1/4 to speed up tracking rate, figure 24 shows the result that use block location of 1/4 size (Figure 24 (b)) to circle objects in original frame (Figure 24 (a)).

![Figure 24: Scaled video frame.](image)

(a) 320*240  (b) 80*60

Because the tracking rate of proposed algorithm is based on region and edge numbers, so we scaled original frame down to 1/4, in order to decrease region and edge numbers. And there are many advantages of down size like less noise, a few region and edge numbers and less memory.
Chapter 6  Experimental Results and Discussion

The section presents the experimental results of the proposed algorithm. The thresholds used are listed in table 2. There are different video resolutions 176*144, 320*240, 760*460 are used at frame rate 15 fps. Figure 25 (a) ~ (i) shows the tracking results where each video sequence has been scaled down to 1/4 to speed up the tracking rate. The results of figure 25 show the capability of the proposed method in object appearance and disappearance handling (see figure 25 (b), (c), (d), (e), (f), (g) and (h)), non-rigid and rigid object movement tracking (See figure 25 (a) ~ (h)), object split and merge tracking (see figure 25 (a) and (b)), object self occlusion and partial occlusion handling (see figure 25 (b) and (i)).

<table>
<thead>
<tr>
<th>Threshold Table</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Noise Remove</strong></td>
</tr>
<tr>
<td><strong>Search radius</strong></td>
</tr>
<tr>
<td><strong>Region color distance</strong></td>
</tr>
<tr>
<td>Th3</td>
</tr>
<tr>
<td><strong>Sum of Similarity</strong></td>
</tr>
<tr>
<td><strong>Location merge</strong></td>
</tr>
<tr>
<td><strong>Occlusion start/end</strong></td>
</tr>
<tr>
<td><strong>Occlusion segmentation</strong></td>
</tr>
</tbody>
</table>

Table 2: Experimental thresholds
| Hall sequence | ![Images](image1.jpg) ![Images](image2.jpg) ![Images](image3.jpg) ![Images](image4.jpg) |
| Lab sequence 1 | ![Images](image5.jpg) ![Images](image6.jpg) ![Images](image7.jpg) ![Images](image8.jpg) |
| Lab sequence 2 | ![Images](image9.jpg) ![Images](image10.jpg) ![Images](image11.jpg) ![Images](image12.jpg) |
| PETS-ICVS | ![Images](image13.jpg) ![Images](image14.jpg) ![Images](image15.jpg) ![Images](image16.jpg) |
| PETS-2000 | ![Images](image17.jpg) ![Images](image18.jpg) ![Images](image19.jpg) ![Images](image20.jpg) |
| PETS-2001 | ![Images](image21.jpg) ![Images](image22.jpg) ![Images](image23.jpg) ![Images](image24.jpg) |
Figure 25: Multiple objects tracking results
6.1 Tracking success rate

In figure 26, it shows rate of intact match, partial match and lose objects in several sequences. Partial match means that there are so many pixels between object and background or there is noise by the object. The average of intact match rate is 97.22%, object lose rate is 1.11% and partial match rate is 1.67%.

Because error of segmentation has brought about tracking error, after remove error of segmentation to make out tracking success rate. The figure 27 shows the result of tracking success rate without segmented error. The average of intact match rate is 97.93%, object lose rate is 0.66% and partial match rate is 1.41%.
Figure 27: Rate of accuracy in video sequences
6.2 Tracking rate speedup

This system can work 7-8 fps on scaling down to 1/4 size. Figure 28 shows tracking rate of original frames and scaled size. After scaling size, system can fast 21 times than tracking rate of original frame. The average of 320*240 size is 0.36 and 80*60 is 7.51 frames/second.

![Figure 28: 320*240 vs 80*60 tracking rate.](image)

In the figure 29 shows the relation between numbers of region and edge pixel and time. The left of figure 29 shows average number of regions and edge pixels which get 100 frames at random from video sequence, the right of figure 29 shows how many frames can be processed in one second and frame size is 80*60.
Moreover, this drawback of this system is object segmentation that noise is considered object, if noise is so large which became an object and shadow of object is obvious, like the regional luminance changing. So we can use those methods to handle object segmentation.

1. Average, median, running average
2. Mixture of Gaussians
3. Kernel Density Estimators
4. Mean shift (possibly optimised)
5. SKDA (Sequential KD Approximation)
6. Eigenbackgrounds

The other advantage use new edge-based that can deal with same color of multiple objects during occlusion, but if two binary image of objects similar with each other in key frame buffer, this new edge-based is imperfect, because it can’t identify them with same contour.

Figure 29: Average of regions / frame, edges / frame and frames / second
6.3 Conclusion

We presented an automatic tracking algorithm based on region, edge, location. Regions are area of object that got by watershed. Edge are pixel of object that got from canny edge, and use hough transform to handle this edge. Location is error of object location between frame N and frame N+1 less than a threshold will be same object, otherwise is a new object.

The proposed algorithm is capable of coping with multiple objects. Track management issues such as object appearance and disappearance, object non-rigid and rigid movements, object split and combine, object occlusion in indoor or outdoor video sequences

Because those drawbacks is discussed in chapter 4.2, so our future works is
(1) Tracking more objects with occlusion.
(2) Region and edge number reduction.
(3) Tracking objects with dynamic background.
(4) Deal with shadow of each object.
(5) Multi-view and depth map to overcome same contour objects.
(6) Run in real-time(15fps).
Appendix A  Morphology

Morphological dilation and erosion operators are used to combine the fragments or fill holes of object. In morphology, dilation operation on a binary image is to gradually enlarge the boundaries of regions of foreground pixels. Its definition with A and B as sets in $Z^2$, the dilation of A by B, denoted $A \oplus B$, is defined as

$$A \oplus B = \{z|[(B^\circ)z \cap A] \subseteq A\}.$$  

This equation is based on obtaining the reflection of B about its origin and shifting this reflection by z. The dilation of A by B then is the set of all displacements, z, such that $B^\circ$ and A overlap by at least one element. Set B is commonly referred to as the structuring element in dilation. Figure 30, 31 and 32 shows simple examples.

Figure 30: 3x3 dilation operation.
Erosion operation on a binary image is to erode away the boundaries of regions of foreground pixels. Its definition for A and B in $Z^2$ the erosion of A by B, denoted $A \ominus B$ is defined as $A \ominus B = \{z \mid (B) z \subseteq A\}$. 

In words, this equation indicates that the erosion of A by B is the set of all points z such that B, translated by z, is contained in A, as in the case
of dilation. Figure 33, 34 and 35 shows simple examples.

<table>
<thead>
<tr>
<th>0</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

Figure 33: $3 \times 3$ erosion operation

Figure 34: Result of erosion

Figure 35: Technological processes of erosion
Appendix B  Hough transform

The Hough transform is a technique which can be used to isolate features of a particular shape within an image. Because it requires that the desired features be specified in some parametric form, the classical Hough transform is most commonly used for the detection of regular curves such as lines, circles, ellipses, etc. A generalized Hough transform can be employed in applications where a simple analytic description of a feature(s) is not possible. Due to the computational complexity of the generalized Hough algorithm, we restrict the main focus of this discussion to the classical Hough transform. Despite its domain restrictions, the classical Hough transform (hereafter referred to without the classical prefix) retains many applications, as most manufactured parts (and many anatomical parts investigated in medical imagery) contain feature boundaries which can be described by regular curves. The main advantage of the Hough transform technique is that it is tolerant of gaps in feature boundary descriptions and is relatively unaffected by image noise.

1. Hough transform of straight lines:

For any point of binary image, the function through this point can be $F(x,y) = y-ax-b=0$. And a, b mean slope and intercept of straight line. This function can be a mapping of mutual constraint, it mean that from
image point \((x, y)\) map into multiple parameters \((a, b)\), or from \((a, b)\) map into \((x, y)\).

\[
\begin{array}{cc}
\text{Image domain} & \text{Parameter domain} \\
\end{array}
\]

Figure 36: Get from [21]

1.1 Accumulator:

As a result of hough transform map \((x, y)\) of each point into \((a, b)\), it can use a accumulator to take down the number of appearances \((a, b)\), then this highest number of appearance \((a, b)\) mean more representative straight line in image domain. For example

Figure 37: Get from [21]

1.2 Hough transform algorithm:

(1). Using edge to be characteristic point in image domain.

(2). Find each of characteristic point \((x, y)\) to calculate

First: for each “a” to calculate all \((a’, b’)\) pass through \((x, y)\).
Second: To accumulate once with \((a', b')\) in accumulator.

(3) Find the maxima in accumulator.

(4) Map maxima into image domain that mean a characteristic straight line.

1.3 Polar system:

When \(a = \infty\), we can’t to take down its accumulator, in fact, we usually use \((\rho, \theta)\) to take the place of \((a, b)\), and its function as follows:

\[x \cos(\theta) + y \sin(\theta) = \rho\]

![Figure 38](image)

Figure 38: Get from [21]

Each of straight line map into each point \((\rho, \theta)\), and each point in image domain map to a curve in parameter domain, as below

![Figure 39](image)

Figure 39: Get from [21]
1.4 Generalized Hough Transform:

In order to detect irregular shape, Ballard bring up Generalized Hough Transform (GHT) at first. Its definition as follows:

\[(r, \alpha) = \sqrt{(x - xc)^2 + (y - yc)^2}, \tan^{-1}((y - yc) / (x - xc))\]

(1) Build R table

1. Give a characteristic point (like barycenter).
2. Build an empty R table, and its index is \(\phi_i\), \(i = 1, 2, M\), angle increased by \(\pi / M\) from 0 to 180.
3. Aim at each point \((x, y)\) of edge, calculating \((r, \alpha)\)

\[r = \sqrt{(x - xc)^2 + (y - yc)^2}\]
\[\alpha = \tan^{-1}((y - yc) / (x - xc))\]
4. Calculate \(\phi\), then put \((r, \alpha)\) into R table in which belong to \(\phi_i\).
5. Repeat 4, until detected over of all points of edge, and it will get R table

\[
\phi_1: \quad (r_1^1, \alpha_1^1), \ (r_2^1, \alpha_2^1), \ldots
\]

\[
\phi_2: \quad (r_1^2, \alpha_1^2), \ (r_2^2, \alpha_2^2), \ldots
\]

\[
\phi_n: \quad (r_1^n, \alpha_1^n), \ (r_2^n, \alpha_2^n), \ldots
\]

Figure 42: Get from [21]
Bibliography


[10] Ferran Marques and Cristina MiLina, ”Object tracking for content-based functionalities”.


[12] Andrea Cavallaro, Member, IEEE, Olivier Steiger, Member, IEEE, and Touradj Ebrahimi, Member, IEEE., “Tracking Video Objects in Cluttered Background”, IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, VOL. 15, NO. 4, APRIL 2005.


[17]Candemir Toklu, A. Murat Tekalp, and A. Tanju

[18] Xiaodong Huang and Eric Dubois,” THREE-VIEW DENSE DISPARITY ESTIMATION WITH OCCLUSION DETECTION”, 2005 IEEE.


