FOLLOWING A MOVING OBJECT IN A BOUNDED AREA

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
ÇANKAYA UNIVERSITY

BY

HASAN EMRE AKKUŞ

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
COMPUTER ENGINEERING

JULY 2009

Title of the Thesis : **Following a Moving Object In a Bounded Area**

Submitted by : **Hasan Emre AKKUŞ**

Approval of the Graduate School of Natural and Applied Sciences, Çankaya University

Prof.Dr. Yahya Kemal Baykal
Acting Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science/Computer Engineering.

M.R. Tolun

Prof.Dr. Mehmet Reşit Tolun
Head Of Department

This is to certify that we have read this thesis and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science/Computer Engineering.

R. Hassanpour

Asst. Prof.Dr. Reza Hassanpour
Supervisor

Examination Date : 08.07.2009

Examining Committee Members

Prof.Dr. Mehmet Reşit Tolun (Çankaya Univ.) M.R. Tolun

Asst. Prof.Dr. Reza Hassanpour (Çankaya Univ.) R. Hassanpour

Dr.Ersin Elbaşı (Tübitak)

STATEMENT OF NON-PLAGIARISM

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work

Name, Last Name : Hasan Emre, AKKUŞ
Signature :
Date : 08.07.2009

ABSTRACT

FOLLOWING A MOVING OBJECT IN A BOUNDED AREA

Akkuş, Hasan Emre
M.S.c., Department of Computer Engineering
Supervisor : Asst. Prof.Dr. Reza Hassanpour

July 2009, 42 pages

The aim of this thesis is to improve the usage of a device called **teleprompter** which is used worldwide for all kinds of presentations. Our purpose is to give a free motion capability to the presenter who uses the teleprompter. Current usage of this device restricts the motion capability of the presenters. The fundamental approach is to detect the presenter's location and turning the screen of the prompter to the presenter.

We use a camera to capture the motion of the presenter. Then we analyze these images captured by the camera and we detect the exact point of the presenter on the area where s/he realizes her/his speech. After we detect the exact point of the presenter, we establish a connection between the presenter and the teleprompter. Then, we turn the teleprompter's screen to the presenter. So the presenter could easily see and read the article displayed over the teleprompter's screen.

The software we developed has two main modules. First module analyzes the presenter's motion and it calculates the angle between the exact point of the teleprompter and the exact point of the presenter. Then it sends this angle's information to the second module which simulates the teleprompter's behavior.

Keywords: Teleprompter , Motion Detection , Motion Analyze

ÖZ

SINIRLANDIRILMIŞ BİR ALAN İÇERİSİNDE HAREKET TAKİBİ

Akkuş, Hasan Emre
Yüksek lisans, Bilgisayar Mühendisliği Anabilim Dalı
Tez Yöneticisi: Yardımcı Doçent Doktor Reza Hassanpour

Temmuz 2009, 42 sayfa

Bu çalışmanın amacı bir sunum esnasında kullanılan ve dünya çapında kullanıcı kitlesi bulunan teleprompter cihazının kullanım özelliklerini artırmaktır. Amacımız teleprompter kullanıcısına rahat hareket etme imkânı vermektir. Halen kullanılmakta olan teleprompter cihazları sunumu gerçekleştiren kişinin hareket kabiliyetini sınırlamaktadır. Temel yaklaşımımız sunucunun alan üzerindeki yerini bulup, cihaz ekranını sunucuya doğru döndürmektir.

Sunucunum haraketlerini analiz edebilmek için bir kamera kullandık. Bu kameranın çektiği görüntüleri analiz ederek sunucunun sunumunu gerçekleştirdiği alan üzerinedeki bulunduğu noktayı tespit ettik. Bir sonraki işlem olarak sunucunun bulunduğu nokta ile teleprompter'ın bulunduğu nokta arasında bir bağlantı kurduk. Bu bağlantıya gore teleprompter'ın ekranını sunucuya doğru döndürdük. Böylelikle sunucu, ekranı kolaylıkla görebilmekte ve ekran üzerindeki yazıları kolaylıkla okuyabilmektedir.

Geliştiridiğimiz yazılım iki ana modülden oluşmaktadır. Birinci modül sunucunun hareketini analiz eder ve sunucunun bulunduğu nokta ile teleprompter'ın bulunduğu nokta arasındaki açıyı hesaplar. Daha sonra bu açı bilgisini ikinci modüle gönderir ve ikinci modül de bu açı bilgisine gore teleprompter'ın davranışlarının simülasyonunu gerçekleştirir

Anahtar Kelimeler: Teleprompter, Hareket Takibi, Hareket Analizi

ACKNOWLEDGMENTS

I wish to express my deepest gratitude to my supervisor Asst. Prof.Dr. Reza Hassanpour for his guidance, advice, patience, criticism, encouragements and insight throughout the research.

I would also like to thank Prof. Dr. Mehmet Reşit Tolun for his suggestions and comments and my cousin Barbaros Preveze for his helps.

Also I must thank to my friends Burak Uytun, Murat Ekici and Sümeyra Ertem because of their technical helps and Nazan Preveze for her advices.

I offer my sincere thanks to my parents for their supports to my decisions for 26 years.

TABLE OF CONTENTS

STATEMENT OF NON-PLAGIARISM iii
ABSTRACT...iv
ÖZ v
ACKNOWLEDGMENTS vi
TABLE OF CONTENTS.. vii
LIST OF FIGURES……………………………………………………………x

CHAPTERS:

1. INTRODUCTION…………………………………………………...….1

1.1 A Brief Overview of the Thesis…………………………….……..4

2. OVERVIEW…………………….…………………………………......6

2.1 Pre-working…………………….……...……………………8

2.1.1 Hardware …….…………..………..……………...……………..…8

2.1.2 Software…………………………………..…………………........8

2.1.3 Location of Camera………………..……………..………………..9

3. REVIEW OF THE METHODS USED……………………………………..10

3.1 Image Comparing Methods…….…………………..…………...………10

3.1.1 Comparing Current Image with a Reference Image……….………10

3.1.2 Comparing Current Image with the Previous Image(s)………...….11

3.2 Morphology………………..…………………………………..………..11

3.2.1 Erosion……………………………………...…...………11

3.2.2 Dilation…………………….……………………...………12

3.2.3 Opening……………………………….………......……………13

3.2.4 Closing………………...…………..…………………...………..13

3.3 Connected Components………………..……………………………..13

4. PROPOSED METHODS……………………………………...…………16

5. REFERENCE (BACKGROUND) IMAGE AND STANDARD DEVIATION....18

5.1 Handling Data…………………………..…………………….………18

5.2 Standard Deviation……….……………………..…………………...…….19

5.3 Image Color Format……………………………………….……..……..20

5.4 Calculating Reference Image's Pixel Values and Standard Deviation…....21

6. MOTION DETECTION……………………………….………..……..23

6.1 Detection of Instant Pixel Motion……………………………...…….23

6.2 Following the Object…………………………………….…….….25

7. CONTROLING THE SCREEN…………………………….….…...…….28

7.1 Establishing a Connection between the Presenter and the Prompter…..….28

7.2 A Problem: Initial Value of the Angle β…………………………………..30

7.3 Turning Control of the Screen……………………………………………..31

8. EXPERIMENTAL RESULTS……………………….…………..………34

8.1 Creating Reference (or Background) Image…………………………...…..34

8.2 Image Comparison………………………………………....………35

8.3 Morphological Operations…………………………………….……..36

8.4 Connected Components and Blob Filtering………………………....….37

8.5 Following the Blob………………………………..………………..38

9. CONCLUSIONS AND FUTURE WORK……………………….…………………39

9.1 Conclusion……………………………………………………....……….39

9.2 Future Work…………………….………………………….……………..40

9.2.1 Background Image Update……………………...……………………...…40

9.2.2 Motion Detection of the Presenter…………………………………….....40

9.2.3 Analyzing the Motion If the Blobs Crash……………………………...….41

REFERENCES……………………...…………………………………………R1

APPENDICESY……………………...………………………...……………………A1

Output Results of the Software Developed……………………………………...A1

LIST OF FIGURES

Figure 1.1 Sample Teleprompter Usage………...……………………....……………..…2

Figure 1.2 Speakers can easily read the article reflected to a transparent glass……..2

Figure 1.3 Presenter walks on the area where s/he realizes her/his speech…………3

Figure 3.1 After erosion is implemented to first image, second is handled ………..12

Figure 3.2 First image is the image which the erosion operation is applied before. Second image is the image which dilation is applied to the first image...12

Figure 3.3 3, 4, 6 and 8 – Connectivity samples are shown……………..………14

Figure 5.1 Assigning pixel values to variables…………………………….….…….21

Figure 5.2 Adding these variables to the buffer provides to find the mean value of each pixel and to calculate the reference image……………....21

.

Figure 6.1 The work flow for instant image's motion detection…………………...24

Figure 6.2 The projection of a human……………………….………..……….26

Figure 6.3 Calculating the exact point of the presenter………………….………27

Figure 7.1 The projection of the teleprompter and the middle point of the projection………………..………………………….…………28

Figure 7.2 The exact points and the angles……………….….…………….…..29

Figure 7.3 Prompter initialization…………………………………….…………30

Figure 7.4 Some possible directions of the teleprompter…………………….…….31

Figure 7.5 The exact point of the prompter and the presenter……………….………..32

Figure 7.6 Direction of the prompter's screen after the angle is calculated………..33

Figure 8.1 Reference image is created after 75 frames are captured……………..……34

Figure 8.2 Black pixels are in motion and white pixels are not……………………..35

Figure 8.3 After the morphological methods are applied to image, noisy pixels are cleared……………………………………………………..36

.

Figure 8.4 First box represents the blobs left after the small sized blobs are cleared. Second box shows the biggest blob in red……………........37

Figure 9.1 If the objects pass very close to each other……………………………..41

Figure A.1 Initial direction of the prompter. Detection process does not start yet…A1

Figure A.2 When the system perceives the motion………………………………...A2

CHAPTER 1

INTRODUCTION

When presenters realize their speeches in front of a crowded society, one of the most important criteria is not to lose eye to eye connection with audiences. This important criterion improves the attention of audiences to the presenter and the topic of the speech. For long term speeches, it is very difficult to talk spontaneously on a specific topic. Usually presenters take notes and read them from a paper. Focusing to notes and reading them causes to lose the eye to eye connection between audience and presenter. This is also acceptable for newsreaders. During on air programs, looking to a paper and reading the content of the program reduces the attention of viewers. To prevent this reduction, Fred Barton, Jr., Hubert J. (Hub) Schlafly, and Irving Berlin Kahn found a new simple mechanical device in 1950s called “Teleprompter”[1]. . Thus, the article can be read from the glass easily. When you look towards to this specific glass, you can read the article but you cannot see anything if you look to glass behind. Audiences do not recognize that the presenter reads his/her speech topic. Prompters are located between the presenter and the audiences. Since the audiences are located at the behind of the prompter, they cannot see the article but the presenter can. So, when the presenter reads the article, audiences think that the presenter looks to them. Thus, eye to eye connection between the presenter and the audiences never disconnects.

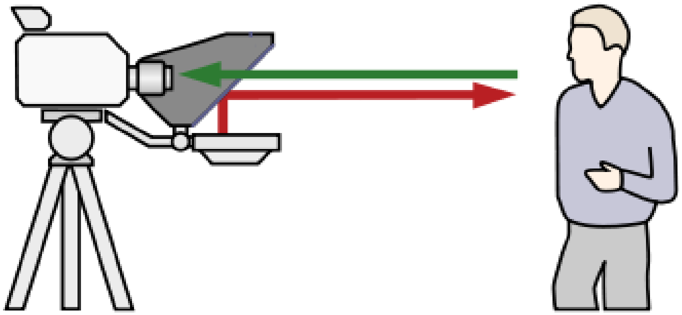


Figure 1.1: Sample Teleprompter Usage [1]

For the last 20 years, Teleprompters attracts the attention of politicians. When they talk to public, two Teleprompters are placed to the left and right side of the politician. Presenter looks to the right-side located teleprompter when s/he wants to look right-side located audiences of her/him, and vice versa s/he looks to the left-side located teleprompter if s/he wants to look left side located listeners.

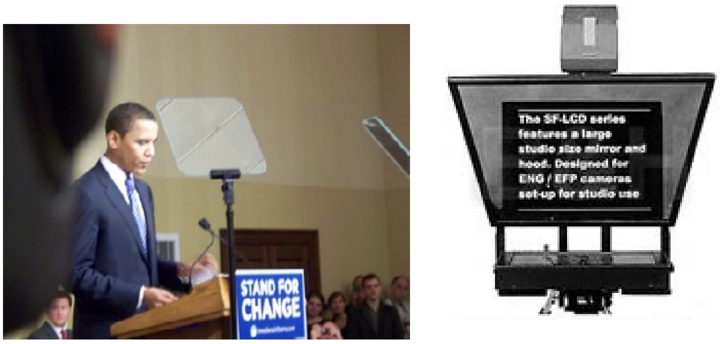

STAND FOR CHANGE
The SF-LCD series features a large studio size mirror and hood. Designed for ENG / EFP cameras set-up for studio use

Figure 1.2: Speakers can easily read the article reflected to a transparent glass.

B. Obama is one of the best known teleprompter users.

As it is seen in Figure 1.2, two Teleprompters are located to the right and left side of B.Obama. Audiences, located back of the teleprompter, just see transparent glass but the presenter, located front of the teleprompter, sees and reads the article easily.

The fundamental working principle of Teleprompters has not been changed since it was found. But, this working principle restricts the motion of presenter. Presenter must be located between right and left side located Teleprompters to see and read the article displayed over glass. The aim of this thesis is to remove this restriction. Our aim is to give a free motion capability to the presenter. We decided to rotate the screen of teleprompter with the right angle where the presenter is located in the area.

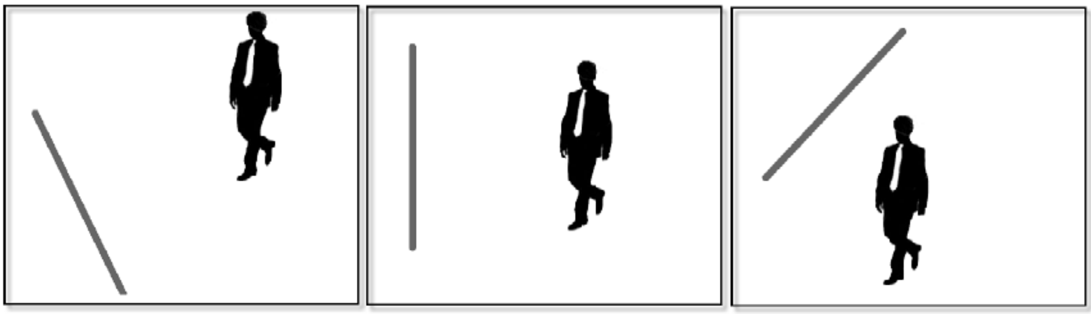


Figure 1.3: Presenter walks on the area where s/he realizes her/his speech.

As seen in figure 1.3, presenter walks on the area where the speech is realized and the teleprompter turns to the presenter. Thus, presenter can read the article and should not be located on a fixed area.

1.1 A Brief Overview of the Thesis

- Introduction
- Reference(or Background) image
- Instant image comparison with reference image
  - This comparison creates a binary image with black pixels which means the pixel is in motion and white pixels which means the pixel is not in motion.
- Applying morphology (erosion + dilation) to binary image created previous step.
  - The binary image created at the previous step may have some noisy pixels. In this step our aim is to delete all these noisy pixels
  - First we apply erosion method to the image and we make the edges smaller. Since noisy pixels are too small, then they disappear.
  - After erosion is applied to the image, we apply dilation method to take the small sized edges to their original size.
  - Applying erosion and dilation consecutively to the binary image clears the noisy pixels.
- After the morphology methods are applied to the binary image, all the edges can be handled clearly without any noise. Some of these edges have connectivity between each other. In this step we find the connected components. These components are called blobs.

  Some of these blobs are too small than we expected. In this step, we delete these small blobs and we ignore these blobs from all steps after now.

- At this step, *main motion following* process starts. Blobs are followed based on some criteria. Our criterion is the biggest blob criteria. We supposed the biggest blob is the blob which we want to follow.
    - In this step the biggest blob is found and filled with red color. Thus, the image has three colors that:
        - Red : The biggest blob that we follow
        - Black : Other blobs in motion
        - White : No motion exists (motionless areas)
- After the biggest blob is detected, we start to follow the motion of it in this chapter.
    - The exact point of the blob is determined and we determine the turning angle of the teleprompter's screen in this step. The exact point of the prompter and the exact point of the blob are used for this angle determination.

CHAPTER 2

OVERVIEW

The display orientation adjustment can be done by estimating the current position of the presenter and rotating the display accordingly. The assumption is that the moving object (in this case the program presenter or the anchorman) can only move in an almost planar way. This assumption reduces the complexity of the position estimation in a 3D environment.

The motion detection module receives a wide-angle camera image as input and computes the difference between consecutive images within a local field. The motion segmenter then uses a region-growing technique to identify contiguous blocks of motion within the difference image.

In this project we investigated two complementary facts for the detection of moving objects by a stationary observer (the camera). The first is based on the fact that, in a rigid environment, the image is constrained to contain the pixel values of the background where there exist no moving object, and pixel values of the moving object when a motion occurs. This fact can be used to estimate the location of the moving object(s) by modeling the background and subtracting the captured images from it. The second fact is that the pixels defining the region of a moving object are contiguous and are not randomly distributed in the image.

To use the first fact, the background image should be modeled. The model should consider the differences stemming from the changes in lighting and noise. A sequence of images taken by a stationary camera can be used to generate the model. Statistical characteristics of the pixel values including the mean and variance are used for establishing a reliable range of changes for the intensity or color value at each pixel. This information is used for segmenting the moving areas in the following stages.

The second fact is useful in eliminating the noise and lighting change effects. A moving object is defined by a contiguous group of pixels which are different than the background by a threshold value. This will allow us to eliminate random points by detecting connected components and ignoring the objects with an area less than a given threshold. The components having an area over a threshold are the candidates for the moving object region. These candidate regions are controlled in the next stage using their texture and position descriptors. An object is supposed to preserve its color and texture values during the motion and between consecutive frames. The poison of the object also cannot change randomly. This means the motion should conform to the estimated direction and velocity of the object which are obtained from previous frames.

The procedure is also simplified by obtaining the input data from the motion plane of the object so we are planning to have the camera positioned on top of the scene covering the anchorman and the screen. This positioning strategy will let us reduce the dimensionality of the problem from three to two. The background image therefore, will be the image of floor and the object on it such as tables, chairs etc.

2.1 Pre-working

2.1.1 Hardware

First, to determine the motion of presenter, the instant location/coordinates must be determined. Our aim is to determine the coordinates of the presenter in a bounded area instantly. So, to analyze the location of presenter, instant images must be captured by one or more than one cameras. Capturing with more than one camera obligates the processor since we handle more than one images at a time but we can achieve 3D image analyze.

In our project, only the instant coordinate of presenter is needed. No more data is needed like the distance between the speaker and the teleprompter or how long the presenter is. So we don't need to handle 3D image. This means one camera is enough to gather necessary data.

The data gathered from camera must be processed by a processor. So, one computer is enough for our project but core 2 or core 2 duo processors are better.

2.1.2 Software

The software we develop to determine the presenter's location is based on c++ and Intel openCv technologies. Intel openCv is an open source library which is used for image processing operations [2].

The simulation of teleprompter is developed on .NET technologies found by Microsoft [3].

2.1.3 Location of Camera

We suppose that, the area that the presenter takes place is a rectangle. On this rectangle, we need to know the exact point where the presenter is located. So, we should analyze the area from top view. This means, locating the camera to the top of the area will capture instant images from above the presenter.

CHAPTER 3

REVIEW OF THE METHODS USED

Several methods have been tested for the best detection so far. In this chapter, a few numbers of them will be analyzed.

3.1 Image Comparing Methods

3.1.1 Comparing Current Image with a Reference Image

Motion detection algorithms are based on comparing images and detecting the pixels in motion. In this project two images, current and background (reference) image, are compared. Basically, the pixel assumed as in motion if the current pixel value is not equal to reference image pixel value. This method is also called as *background subtraction* [4].

Main problem of this method is the changes in background. Changes in background may be perceived as motion. When a new object is included to the background, it is sensed as a motion at the first including time but later it is not. So, background image must be updated during the detection process.

3.1.2 Comparing Current Image with the Previous Image(s)

Comparing current image with previous image(s) will give a result for all pixels if the pixel in motion or not. This method is the pure motion detection method but it has some disadvantages. Most cameras capture frames with noises. This method perceives these noisy pixels as motion. These pixels could be removed by other techniques like *erosion* but since the previous images always changes, our reference image pixels values changes dynamically. We don’t want this change. When a car or a person just passes back of the image, it could be perceived as motion. Sometimes we don’t consider the motions as a motion arise from such small changes.

3.2 Morphology

When a camera captures a sequence of images, some of pixels may be captured with noise. These noisy pixels may be perceived as motion. In other words, these noisy pixels cause incorrect detection. Another problem is the detailed motions. Even flying of a bird faraway from camera is perceived as a motion. Sometimes we may want to remove such small motions.

3.2.1 Erosion

Erosion is a methodology to remove such irrelevant motions from a binary image [5]. The point of erosion is to make smaller than normal such small or noisy motions with a threshold value. If the motion detected area is smaller than this threshold value, then this area disappears as a result of erosion methodology.

$$A \ominus B = \{z \mid (B)z \subseteq A\} \text{ ……………………………………………………………… } (3.1)$$

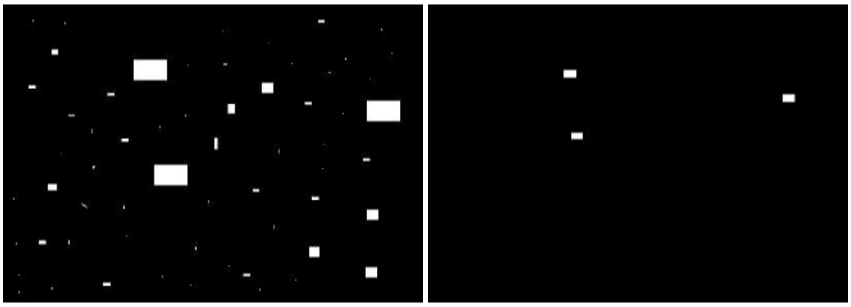


Figure 3.1: After erosion is implemented to first image, second is handled

3.2.2 Dilation

Small regions detected on an image may sometimes need to be enlarged. In image processing, after erosion is applied to an image large areas get smaller and small areas disappear. Since areas get smaller, real data is destroyed. To prevent this data loss, after we make them smaller, we should re-size them and we should adjust these areas into their original size. Dilation methodology enlarges these areas by a threshold value.

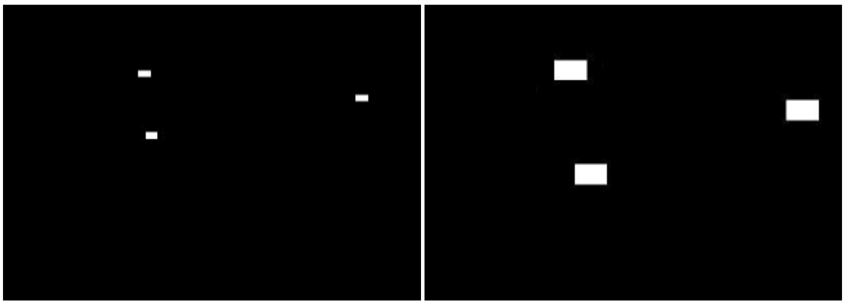


Figure 3.2: First image is the image which the erosion operation is applied before. Second image is the image which dilation is applied to the first image.

As it is seen, when we apply erosion and dilation consecutively to an image, small and noisy regions disappear and the main regions stay on image.

Applying erosion and dilation to an image consecutively clears noisy and small areas form the image then re-size large areas into their original size. Thus, unwanted regions are destroyed from the image.

$$A \oplus B = \left\{ z \mid (B)z \bigcap A \neq \emptyset \right\} \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots (3.2)$$

3.2.3 Opening

The opening method of morphology is the dilation of the erosion of a binary image. This method removes small objects from image, clears small holes, smoothes the contour of objects…

$$A \circ B = (A \ominus B) \oplus B \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots (3.3)$$

3.2.4 Closing

The closing method of morphology is the erosion of the dilation of a binary image. This method is used for eliminating small holes, filling graphs…

$$A \cdot B = (A \oplus B) \ominus B \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots (3.4)$$

3.3 Connected Components

In graph theory [6], if two vertices are connected to each other and there is no vertices connected to these vertices, then it is said that these vertices are *connected* *components* [7]. If a point connected to other at 3 points then is called 3-connectivity, if 4 points it is called 4-connectivity and same as 6 and 8.

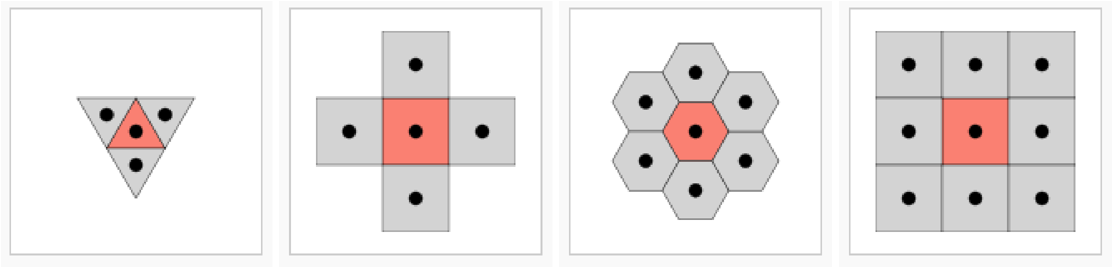


Figure 3.3: 3, 4, 6 and 8 – Connectivity samples are shown [8]

In motion detection, after some appropriate methods are applied and noisy pixels are disappeared, connected components give a brief and clear view of objects on the image. We can see all the objects clearly after we determine connected components. There may appear one or more connected components on images and we may need to give a unique name or a unique number to each component to separate each from others. Giving a unique name or a unique number to each connected component is called *Component Labeling* [9].

In graph theory [6], a blob is a point or an object on an image. To detect a motion, as a first process, the blob that we want to detect its motion should be determined. After the blob is determined, other blobs are eliminated. Removing other blobs from the image facilitates the process. To detect blobs, commonly used method is analyzing the neighbor pixels of a pixel. This process is applied to all pixels in an image. If 4-connectivity connected components needed to be detected, then north, west, south and east pixels are checked and additionally northeast, northwest, southeast, southwest pixels are checked if 8-connectivity connected components are needed to be analyzed. If none of these pixels are labeled as a black pixel in a binary image, then this point is labeled as a region value. If only one pixel is pointed as a black pixel, this pixel is labeled as an element of the current region. If more than one pixel are in black and they are not labeled in a region, then all of them is labeled as in the current region. If more than one pixel in the current region and some of them or all of them are labeled in the other regions, then these multiple pixels are assigned current or other region [8].

CHAPTER 4

PROPOSED METHODS

In this thesis, we decided to use the reference image comparing method. Comparing the instant image with a dynamic background image gives better results. Otherwise, small motions, flight of a bird, are detected as motion. One important point is the changes in reference image. Reference image should be updated during the detection process. Some objects could be placed into the area where we try to detect the motion like tables or chairs. This update operation is not included in our project but we add updating reference image to our project as a future work in chapter 9. Determining reference image is discussed in chapter 5 in details.

After the reference image is prepared, instant images are compared with this reference image. As a result of this comparing, we get a binary image with black pixels represent motion and white pixels represent stability. Since some noisy pixels and small motions are detected as motion, we use morphological methods to remove them. With erosion we clear noisy pixels and small motions and then we enlarge these reduced areas into their original size with dilation.

After dilation, some independent components are handled. If these independent components are very close then they supposed as one component called *connected* *component* or *blob*. After all components are identified, we suppose the biggest component is the object which we want to track. All the blobs are ignored but the biggest blob. We focus to the biggest blob and we follow the motion of it. We find the midpoint of the biggest blob and we use this midpoint as a reference point during the detection process. The average value on x axis is the mean value of the biggest and the smallest points horizontally and the average value on y axis is the mean value of the biggest and the smallest points vertically. The point of (x, y) on the area is the exact point that we want to follow.

We can handle the absolute point of the object we want to follow untill now. Now we need to establish a relation with this point and the point where the teleprompter is located. The direction of the teleprompter is determined based on this relation.

CHAPTER 5

REFERENCE (BACKGROUND) IMAGE and STANDARD DEVIATION

Our project is fundamentally based on comparing the pixels which is handled instantly and the motionless pixels which we hold at the background of our software as *background image*. The values of these motionless pixels are calculated before the motion detection process is started. The image, generated by these motionless pixels, is called *reference image* or *background image.*

5.1 Handling Data

Reference image is created after taking a sequence of images of the area where the speech will be recognized. This capturing process must be done when the area is purely empty. After a sequence of images is handled we calculate the mean value of these images. What we mean by average image is the division of the total sum of each pixel of the sequence to the number of images of sequence.

To calculate the total values of pixels, we should create a buffer first. This buffer holds the total sum of values of each pixel. All the values of new images captured by camera are added to this buffer during process and total values of each pixel are handled when capturing image sequence operation is ended.

$$Avarage\ Value\ of\ Pixel = \frac{Total\ Values\ Of\ Pixel}{Total\ Number\ of\ Images\ At\ The\ Image\ Sequence} \quad (5.1)$$

Detecting motion of a pixel is based on comparing instant pixel value and the same pixel's background image value. Theoretically, if the pixel values are not same, it can be said that the pixel is in motion. But practically, the pixel values can change in a close bound. So we can assume that if the absolute value of the subtraction of instant image's pixel value from the background image's pixel value is less than a value x, pixel is not in motion and pixel is in motion otherwise. Since we take the absolute value of the subtraction, our result must be greater than zero; our bound value is between zero and x value. How can we determine the value x value?

5.2 Standard Deviation

When calculating the mean value of the pixels, the biggest and the smallest values affect the background image's pixel value's distributions. In this case, central distribution, called *standard deviation*, is being calculated.

$$\delta(x,y) = sqrt\left(\frac{Sq(x,y)}{N} - \left(\frac{S(x,y)}{N}\right)^2\right) \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots.(5.2)$$

In the formula 5.2, *N* is the total number of frames captured; *Sq(x, y)* is the squares of the values for each pixel; *S (x, y)* is the sum of each pixel value.

When calculating the standard deviation formula, firstly, the mean values of the pixels and the square values of these mean values are calculated. After calculating the subtraction of these two values, square root of the subtraction is calculated. The final calculated value is the *standard deviation* of the pixel value.

Now, mean values and standard deviation values of each pixel are calculated. The image created by mean values is called *background image* or *reference image.*

5.3 Image Color Format

Several image formats are being used for image processing. BGR and RGB formats are the best known formats which hold each pixel’s value in a color sequence of Red, Green and Blue. A color value may be changed with respect to the light. The value of light green and bottle green are different. Since Teleprompters can be used both outside or inside area or at a sunny or clouded day, the light affects the values of pixels. This effecting may cause changing of pixel values. Using HSV method ignores this changing since it holds values as Hue, Saturation and Value. The reference image values and standard deviation values are calculated from HSV formatted images. .

Our image processing tool openCv captures frames in BGR format. For this reason, we convert the format of image from BGR to HSV as a first process during all capturing processes.

5.4 Calculating Reference Image's Pixel Values and Standard Deviation

To calculate the reference image and the standard deviation, we create a buffer that holds the total values of images' pixels. To calculate the reference image, as a first process, we capture a sequence of image. When capturing these images, there must be no motion appear at the area where the speech will be realized, because we want to handle the simplest state of the area. After an image is captured, we convert the image format into HSV format. In HSV format of an image, there are 3 * width of image * height of image data existing. Because there are width of image * height of image pixels exist in an image and H, S and V values are held for each pixel. As an example; the first three data points the first pixel's H, S and V values. And second three data points the second pixel's HSV values and so on. We assign these three values to three variables as double.

```
double pixelValue_H  = (double)(unsigned char) capturedImage->imageData[i];
double pixelValue_S  = (double)(unsigned char) capturedImage->imageData[i + 1];
double pixelValue_V  = (double)(unsigned char) capturedImage->imageData[i + 2];
```

Figure 5.1: Assigning pixel values to variables

These three variables are added to the buffer. The buffer holds the pixels' total pixel values by adding the current value to the buffer.

```
this->bufferOfMeanImage_HSV[i]     += pixelValue_H;
this->bufferOfMeanImage_HSV[i + 1] += pixelValue_S;
this->bufferOfMeanImage_HSV[i + 2] += pixelValue_V;
```

Figure 5.2: Adding these variables to the buffer provides to find the mean value of each pixel and to calculate the reference image.

After capturing process completed, dividing the buffer's value to the number of images captured will give us the mean value of the related pixel. We apply this process to all pixels. As the result, for each pixel we handle a mean value. The image created by these mean values is called *reference image* or *background image.*

As seen in the formula 5.2, when calculating the standard deviation, total value we hold in buffer constitutes core processes. First we calculate the squares of buffer values over total number of images and we calculate square of mean value. After we take the subtraction of these two values then we take this subtraction's square root. This result gives us the *standard deviation* for each pixel.

CHAPTER 6

MOTION DETECTION

After creating reference image and calculating the standard deviation, we can start to detect the motion. Our detection is based on subtracting the instant image captured by camera from our reference image.

Theoretically, it can be assumed; if the absolute value of the subtraction of instant images pixel values from reference images pixel values is not zero, then the pixel is assumed as in motion. But in practice, assuming the pixel is in motion if the absolute value of the subtraction is out of a range.

6.1 Detection of Instant Pixel Motion

Before the detection process starts, we determine a motion range. If the absolute value of the subtraction process is greater than this range, then we assume that the pixel is in motion.

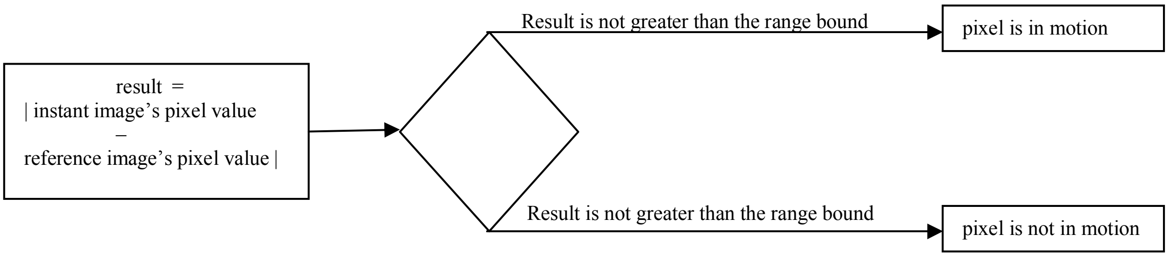

result =
| instant image's pixel value
–
reference image's pixel value |
Result is not greater than the range bound
pixel is in motion
Result is not greater than the range bound
pixel is not in motion

Figure 6.1: The work flow for instant image's motion detection

Standard deviation is used at this point. Standard deviation value holds a value related to the upper bound of our range value. The ***Three Sigma Rule*** says us "the value is in the region of the normal distribution cannot be lie at a distance from its mathematical expectation or more than three times the standard deviation" [10]. Thus, we assume our absolute difference of a pixel between "0" and "3 x Standard Deviation "value. Since our difference is an absolute value, we can say that *if the absolute value of the difference of the instant image's pixel value and the reference image's pixel value is greater than 3 times of related standard deviation value, then the pixel is in motion.*

Instant images we capture from a camera are in BGR format and the reference image is in HSV format. So, before analyzing the instant images, we should convert instant images' image format into HSV. After we compare the reference image and the instant image, we handle a result for each pixel as *the pixel is in motion* or *the pixel is not in motion.* Since our output result may only have two probable results, we create a new image in *binary format* and set each pixel color as black if the pixel is in motion or white if not. With the help of this binary image we can see all the pixels weather if the pixel is in motion or not.

This binary image shows us the pixels in motion clearly. But some pixels may affect from noise. We use erosion method to disappear all these noisy pixels. Using erosion reduces the objects both in motion or not. Thus, the noisy pixels in motions disappear too. After we reduce the objects, we use dilation method to enlarge these objects and we bring these objects to the original size.

6.2 Following the Object

After morphology is applied to the binary image we created to see the pixels if in motion or not, pixels in motion can be seen clearly. The objects in this binary image are seen exactly. An object on the image is consisting of several pixels on x-axis and y–axis. In our project, we need only one exact point to determine the exact place of the presenter on the area. We assumed that, this exact point is calculated by taking the mean values of both x and y axis. We find the biggest and the smallest values on x axis and y axis. After adding the biggest and the smallest values together for an axis, dividing the sum by two gives us the exact point for the related axis.

Blob filtering is removing some blobs from an image. In this thesis, since we follow a human as an object, our criteria changes according to requirements. Ex: We can separate the exact human body which we want to follow from others according to clothes that s/he wears. We can follow the human who wear a red jacket or so on… As a first process, we detect the non-white pixels in the binary image. All these non-white places are parts of blobs. After detecting all blobs, we remove some blobs which are smaller than we expected. Ex: When we capture an area where the camera is located x meter above, we assume that the projection of an average human body's radius is y cm.

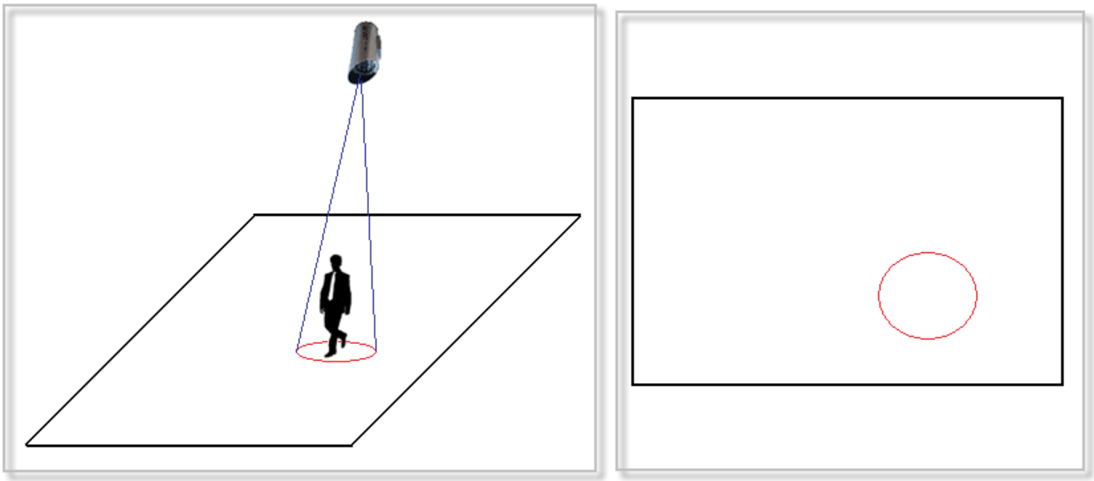


Figure 6.2: The projection of a human

In the first box of figure 6.2, the 3D projection of the human is drawn as a red circle. This projection is shown as a red circle in a 2D perspective in the second box of figure 6.2. All these projections are identified as blobs. Of course, a human's blob is different than some other object's blobs like a chair or a table. We determined an average human blob area and we remove all the blobs smaller than this area. After removing them, only the blobs equal or greater to our average blob size left.

In this thesis, we assume that, the human whose projection of his/her blob on the binary image is the largest one, this human is the presenter. So, we will follow the biggest blob on the binary image and we will remove the other blobs left.

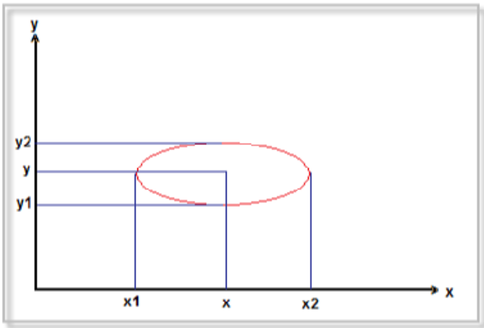


Figure 6.3: Calculating the exact point of the presenter

In Figure 6.3, the blob of the presenter is simulated. We find the greatest and the smallest values of the blob's projection on x and y axes. When we add the blob's the biggest and the smallest points on the both x and y axes and we divide these totals by two, we handle the exact point where the presenter presents his/her speech. In the figure 6.3, the biggest point on x axis is shown as x_2, the smallest point on x axis is shown as x_1, the biggest point on y axis is shown as y_2 and the smallest point on y axis is shown as y_1. The exact point of the presenter on x axis is shown as x and the exact point of the presenter on y axis is shown as y in the figure. The point (x, y) is called as exact point of the presenter on the area and this exact point changes frame by frame. We rotate the teleprompter's screen according to that exact point.

$$x = \frac{x1 + x2}{2} \; ; \; y = \frac{y1 + y2}{2} \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ (6.1)$$

CHAPTER 7

CONTROLING THE SCREEN

In this chapter, *how the direction of the screen is determined* is explained. We assumed that the teleprompter is located between the audience and the presenter. If the teleprompter stands behind the presenter, turning process of screen will not work correctly. Our aim is to stand the teleprompter's screen vertically to the presenter. Thus, presenter can see the screen easily.

7.1 Establishing a Connection between the Presenter and the Prompter

In the previous chapter, we have discussed how to find the exact point of the presenter. Now we have to create a relationship between the exact point of the presenter and the exact point of the teleprompter. We assume that the 2D projection of the screen's middle point is the exact point of the teleprompter.

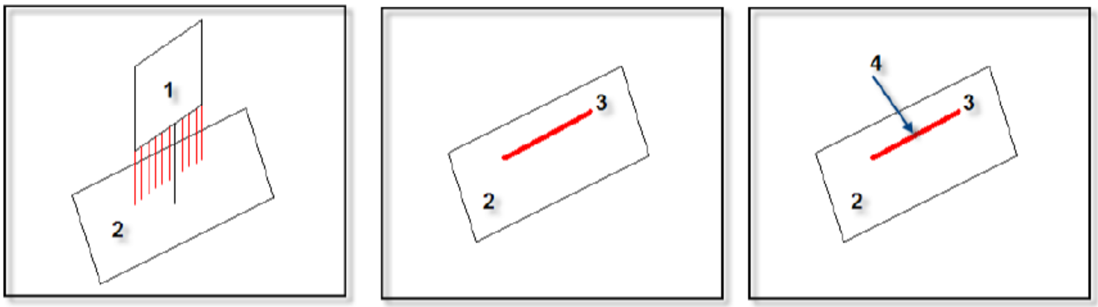


Figure 7.1: The projection of the teleprompter and the middle point of the projection

In figure 7.1, a teleprompter (1) is shown on a plane (2) in the first box. In this picture, the red lines are used to determine the projection of the teleprompter's screen. The projection line (3) is shown in the second box of figure 7.1. We assume that the exact point of the teleprompter is the middle point (4) of the projection (3) which is shown in the third box of figure 7.1. Now, our aim is to create a relationship between the exact point of the presenter and the exact point of the teleprompter. When the connection between these two points is established, we turn to prompter's screen to the presenter in 90°.

After the connection is established between the two exact points, we draw a virtual line between these two points. The angle between the virtual line and the x-axis is identified as alpha (α) and the angle between the upper bound of projection of the teleprompter and the x-axis is defined as beta (β). If the sum of these two angels is 90° or near to 90° then we assume that the presenter can see the prompter's screen easily.

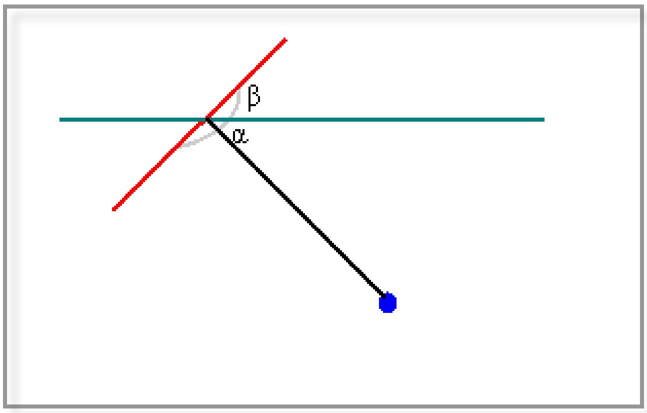


Figure 7.2: The exact points and the angles

In figure 7.2, the projection of the prompter's screen is shown as red, the exact point of the presenter is shown as blue and the x-axis is shown as green color. The angle α is changes according to the place of the presenter. One important point is; the x-axis line passes over the exact point of projection. Here, our aim is to turn the projection line toward to x-axis by the angle of β. The value of β angle is (90 – α) °.

7.2 A Problem: Initial Value of the Angle β

Since we do not know the initial direction of prompter's screen and also the direction of the projection, we must initialize the direction. If we draw two virtual lines from the corner points of the projection to the exact point of the presenter and if the intersection of these lines' angle is a right angle or close to a right angle then the direction of the prompter is initialized.

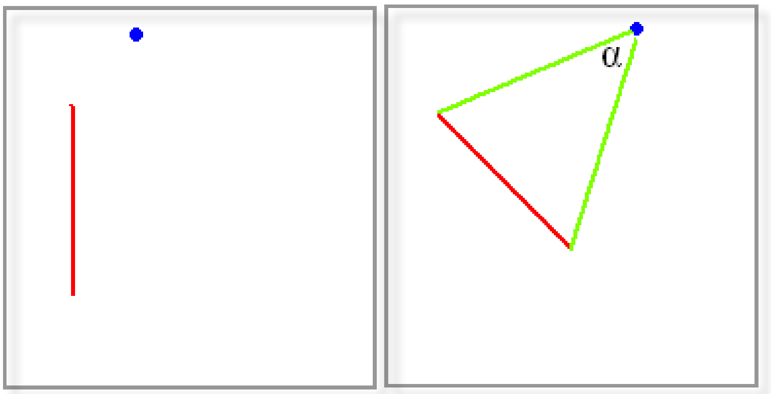


Figure 7.3: Prompter initialization

In figure 7.3, first box shows the exact point of presenter in blue and prompter in red. We assume that the presenter just entered the area and the motion detection process is not started yet. In the second box, we draw two virtual lines in green from the corners of the projection to the exact point of the presenter. We turned our prompter's screen to the presenter up to when we the angle α is set to 90°.

7.3 Turning Control of the Screen

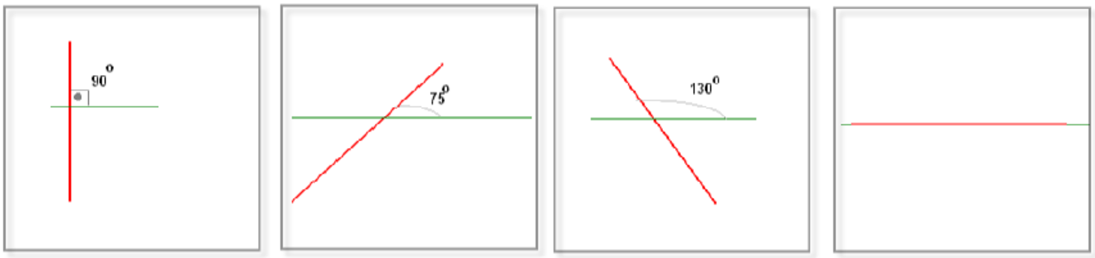


Figure 7.4: Some possible directions of the teleprompter

In Figure 7.4, some possible directions of the teleprompter are shown. The projection of the teleprompter is shown in red and the x axis is shown in green.

In the first box, the initialized direction is focused. The teleprompter intersects the x axis in a right angle. In the second box, the teleprompter is turned to its right side by 15° from the initial point and the angle between the teleprompter and the x axis becomes 75°. In the third box, vice versa, the teleprompter is turned to its left side by 40° and the angle between the teleprompter and the x axis becomes 130°. In the forth box, the teleprompter is turned by 90° to its left and it becomes parallel to the x axis. We assume that the teleprompter can turn 90° to the left and to the right maximally. So it can turn in 180° range.

We developed a method that controls the direction of the prompter. This method takes the angle value between the teleprompter and the x axis. After taking this parameter, method turns the teleprompter's screen. Here, the important point is to calculate the degree.

We assume that the teleprompter is located at a point on the area and we know that exact point of the teleprompter. Calculating the exact point of the presenter is mentioned at chapter 6. So, we know the exact point of the presenter and the teleprompter.

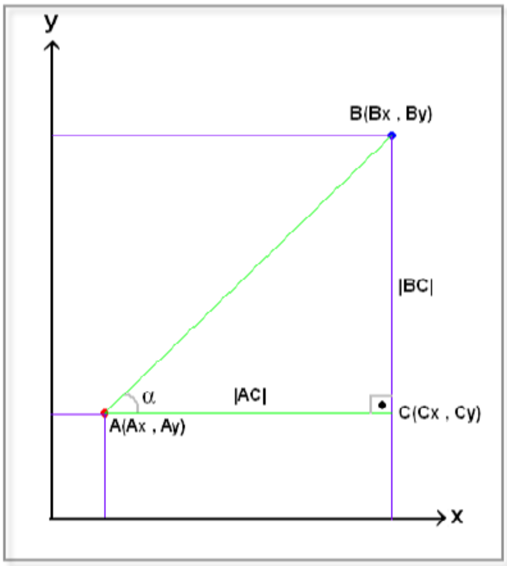


Figure 7.5: The exact point of the prompter and the presenter

In figure 7.5, the exact point of the prompter is shown as a red dot and the exact point of the presenter is shown as a blue dot. The point A represents the location of the teleprompter and the point B represents the location of the exact point of the presenter. The projections of the points A and B are shown in purple. We draw two virtual green lines from the point A to the point B and to the point C. Point C is the intersection point of the projection of the point B to the x axis and the second virtual line we draw.

$$|AC| = |Cx - \ Ax\ | \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \ldots (7.1.1)$$

$$|BC| = |By - Ay\ | \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots . (7.1.2)$$

$$\alpha = \arctan\left(\frac{|\mathrm{BC}|}{|\mathrm{AC}|}\right) \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots (7.2)$$

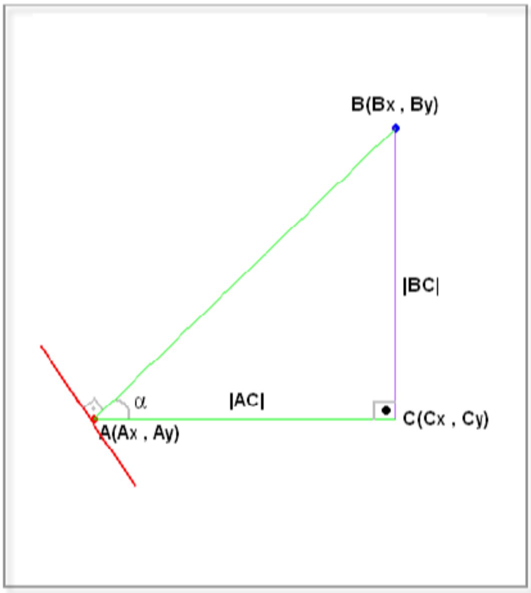


Figure 7.6: Direction of the prompter's screen after the angle is calculated

In Figure 7.6, the direction of the prompter is shown. After the angle α is calculated, our system turns the prompter's screen to its left with an angle α°. We send the value of α as a parameter to the method which controls the direction of the prompter.

CHAPTER 8

EXPERIMENTAL RESULTS

In this chapter, the results of our experiments are mentioned. After each process, we take a screen shot of the output results. Our main process outputs are shown below.

8.1 Creating Reference (or Background) Image

As a first process, 75 frames are captured consecutively and all the pixel values are added the relative pixel value in a buffer. Then the value of the buffer is divided by 75, total number of images, and the mean value of each pixel is calculated. A reference image is created by these values shown in Figure 8.1.

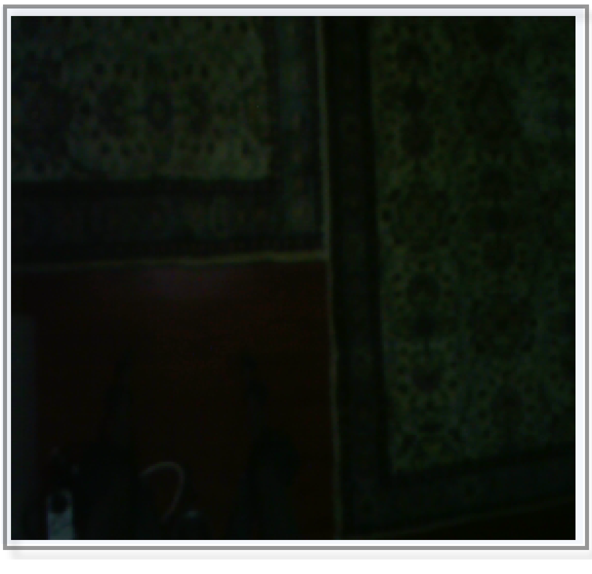


Figure 8.1: Reference image is created after 75 frames are captured

Also standard deviation of each pixel is calculated during the creation of background image process.

8.2 Image Comparison

During the following process, after we capture an instant image, we compare the image’s pixel values with reference image’s pixel values by a ratio of standard deviation. If the absolute difference between the instant image’s pixel value and the reference image’s pixel value is greater than three times of standard deviation, then the pixel is said to be in motion. We create a binary image in this step to show the pixels in motion or not. If the pixel in motion then the pixel is shown in black in the binary image and vice versa the pixel is white if the pixel is not in motion.

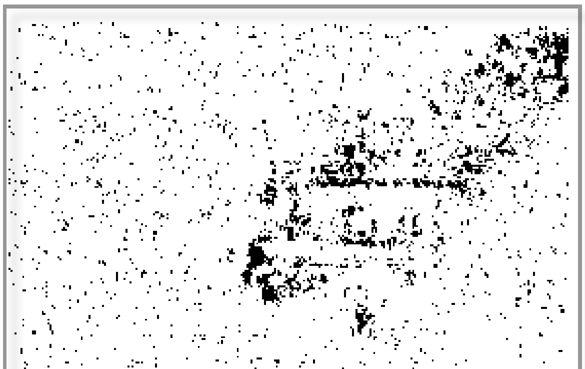


Figure 8 .2: Black pixels are in motion and white pixels are not

In Figure 8.2, the pixels in black are the pixels in motion. But, it is clearly seen in Figure 8.2, right-upper corner is the part of the image where the real motion is appear. The other parts are the noisy parts of the image.

8.3 Morphological Operations

In this step, our aim is to clear the noisy pixels without losing any real data. Morphological methods erosion and dilation are appropriate to clear the noisy pixels. Erosion method makes the edges smaller. Thus, small edges including noisy pixels are cleared. After applying erosion we apply dilation method to resize the edges to original size. Applying morphological methods to Figure 8.2 gives the output of Figure 8.3.

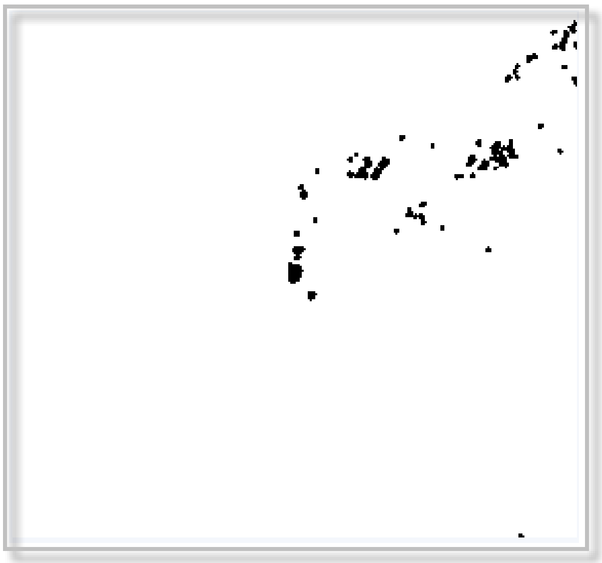


Figure 8 3: After the morphological methods are applied to image, noisy pixels are cleared.

8.4 Connected Components and Blob Filtering

After morphological operations are applied to the image, the pixels in motion are handled clearly. Some of the edges have connectivity relations between each other. They are called as *connected components* or *blobs*. After we determine all the blobs in the binary image, we delete some of these blobs which are less size than we expected. After removing them, bigger sized blobs are seen in the binary image. Blobs are being followed by some criteria. We suppose that the biggest blob is the one which we want to follow. The biggest blob is shown in red color; other blobs are shown in black and the pixels that are not in motion shown in white color.

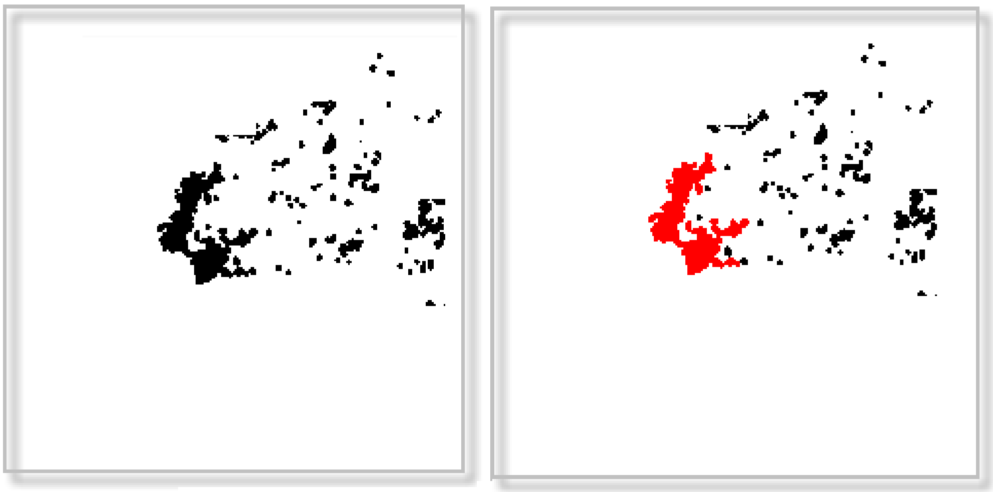


Figure 8.4 : First box represents the blobs left after the small sized blobs are cleared. Second box shows the biggest blob in red.

8.5 Following the Blob

After the biggest blob is handled, we start to follow of it. The biggest and the smallest values' of that blob's mean value gives us the blobs exact point. We assume that the exact point of the prompter is known. Creating a relation with these exact points helps us to turn the prompter's screen in an angle. Our experimental outputs about our software can be seen in appendices.

CHAPTER 9

CONCLUSIONS AND FUTURE WORK

9.1 Conclusion

During long time presentations, a simple device called prompter is used not to lose eye to eye connection between the presenter and the audience. But this simple device restricts the motion of the presenter because two prompters are located statically to the right and left side of the presenter. With the help of a camera, software and a revertible screen, this moving restriction could be disposed. If the software can truly processes the images captured from the camera and it can determine the exact point of the presenter, the screen of the prompter can be rotated to the presenter. Thus, presenter obtains a walking capability on the area where s/he realizes her/his presentation.

In this project, we compare the instant images with a reference image. This reference image is created when there is no motion exists on the area where the presentation will be realized. After comparing these two images, core motions could be handled. But some noisy pixels are included to these core motions. We use morphological methods to remove these noisy pixels. After removing noisy pixels, we detect the blobs on the image by using the “connected components” algorithms. We determine the specific blob which we will follow and we calculate the angle between the exact point of this blob and the prompter.

After we calculate the angle between the teleprompter and the presenter, we simulate the behavior of the prompter's screen.

9.2 Future Work

9.2.1 Background Image Update

During the detection process, some unexpected objects may be inserted to the background. Our system senses such objects in motion but these objects may not be in motion. Example: When the presenter realizes her/his speech, a table could be placed to the area. If the table will stay until the end of the speech then the system must perceive such changes and not detect as a motion. If an object has been placed to the area and do not moves for a long time, that object should be added to the background image and the background image must be updated. Background image should improve and renew itself. Such permanent changes should be added to the background image.

9.2.2 Motion Detection of the Presenter

We assume the biggest blob in the area is the presenter and we follow the motion of it. Of course, in real life such a system does not work correctly. Detection process must be depending on some criteria. Example: We may define the color of the clothes of the presenter and the detection process is based on these colors. In other words, we eliminate the blobs according the color of it and we follow the color defined before the detection process started.

Another option is determining the area where the presenter is located initially. An editor may select a local area manually from the main area and the blob occurred in that area is taken as the object to be followed.

Motion process does not jump from some local area to another instantly. Example: Suppose that we are following an object located at the right corner of the main area. If the system detects a motion at the left corner of the main area for a short time, we can eliminate that motion because we do not expect a motion at the other places of the main area.

9.2.3 Analyzing the Motion If the Blobs Crash

Suppose that two blobs which have same criteria to be followed walks on the area and passes each other very close.

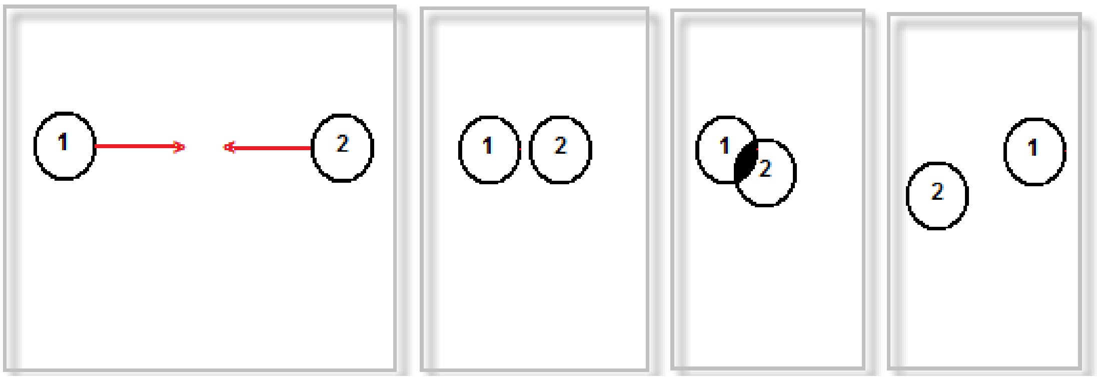


Figure 9.1: If the objects pass very close to each other

As seen in figure 9.1, the blobs 1 and 2 have same tracking criteria. In the box one, the blob 1 walking to right side and the blob 2 to left side. In the box two, they come together and crash each other which showed in the box three. If such an event happens, the system should not lose or confuse the blob that it follows. If the system was following blob 1, then it should continue to follow the blob 1 in the 4th box. Therefore, the system must analyze the direction of the motion in real time. In such an event, if the system was following the blob 1, it had to know that the blob one is walking to right side. After crashing, system has to analyze the direction of the blobs of 1 and 2, and since it knows the direction of blob 1, it must ignore the blob 2 after the crash.

REFERENCES

[1] http://en.wikipedia.org/wiki/Teleprompter

[2] http://software.intel.com/en-us/articles/intel-integrated-performance-primitives-intel-ipp-open-source-computer-vision-library-opencv-faq/

[3] http://www.microsoft.com/downloads/Browse.aspx?displaylang=tr&productID=DE7BB609-3FD0-4B0F-865D-5ED2463AD5D0

[4] *Open Source Computer Vision Library Reference Manuel*, Order Number: A77028-004

[5] **R. C. Gonzalez and R. E. Woods**, 2002. *Digital image processing, 2nd ed. Upper Saddle River*, N.J.: Prentice Hall.

[6] **Gibbons, Alan**, 1985.*Algorithmic Graph Theory*, Cambridge University Press.

[7] **T. Asano, Y. Igarashi , H. Nagamochi, S. Miyano, S. Suri** , 1996. 7th International Symposium ISAAC 96 , Osaka , Japan , December

[8] http://en.wikipedia.org/wiki/Connectedness

[9] H**. Samet and M. Tamminen** . 1988. *Efficient Component Labeling of Images of Arbitrary Dimension Represented by Linear Bintrees*. TIEEE Trans. Pattern Anal. Mach. Intell .

[10] http://en.wikipedia.org/wiki/Blob_extraction

[11] **V. Smirnov, I.V. Dunin-Barkovskii**, 1969. "*Mathematische Statistik in der Technik*" , Deutsch. Verlag Wissenschaft.

APPENDIX

OUTPUT OF SOFTWARE DEVELOPED

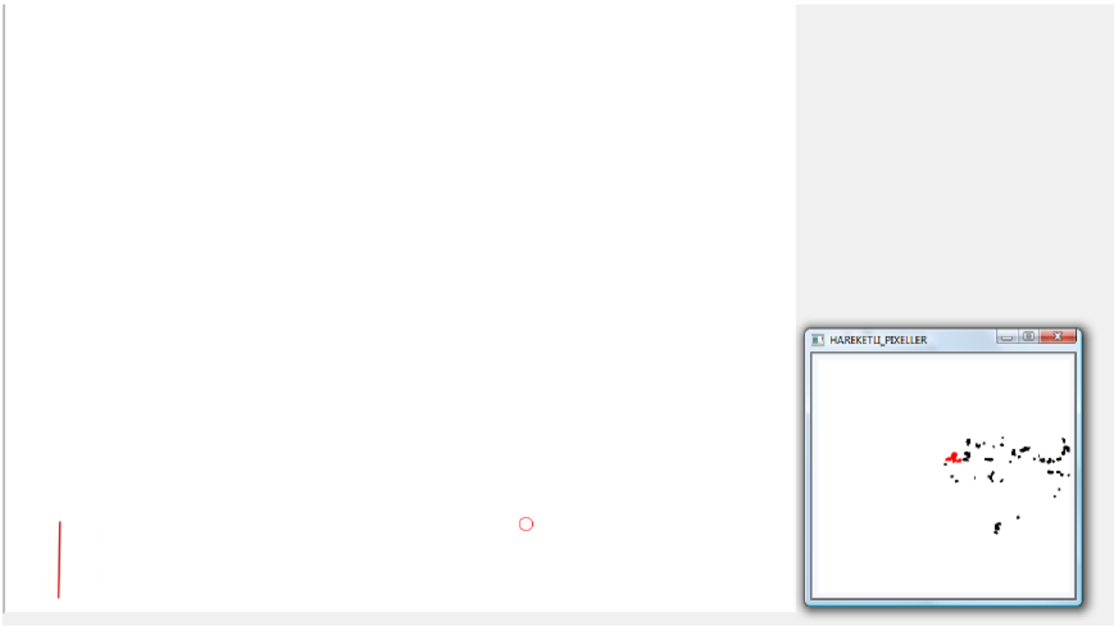


Figure A.1: The initial direction of the prompter. Detection process does not start yet.

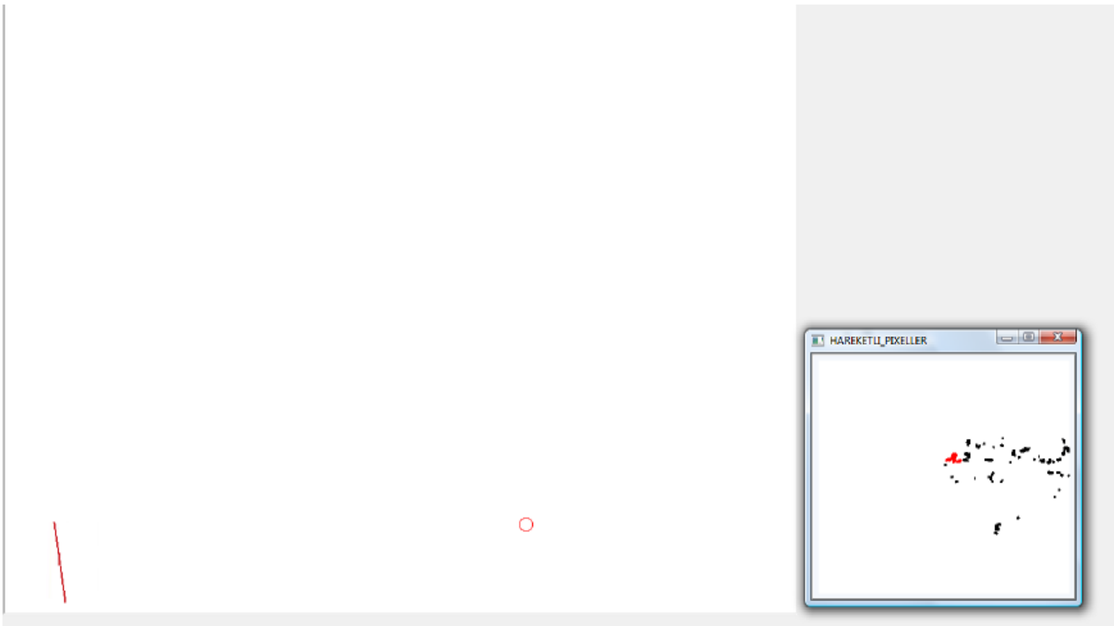
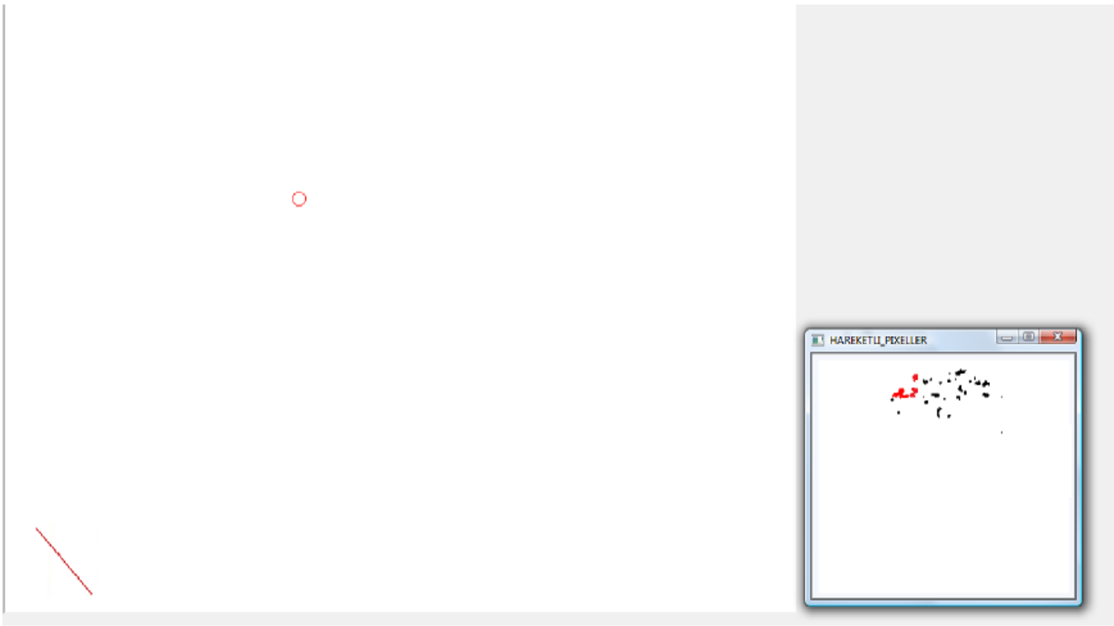
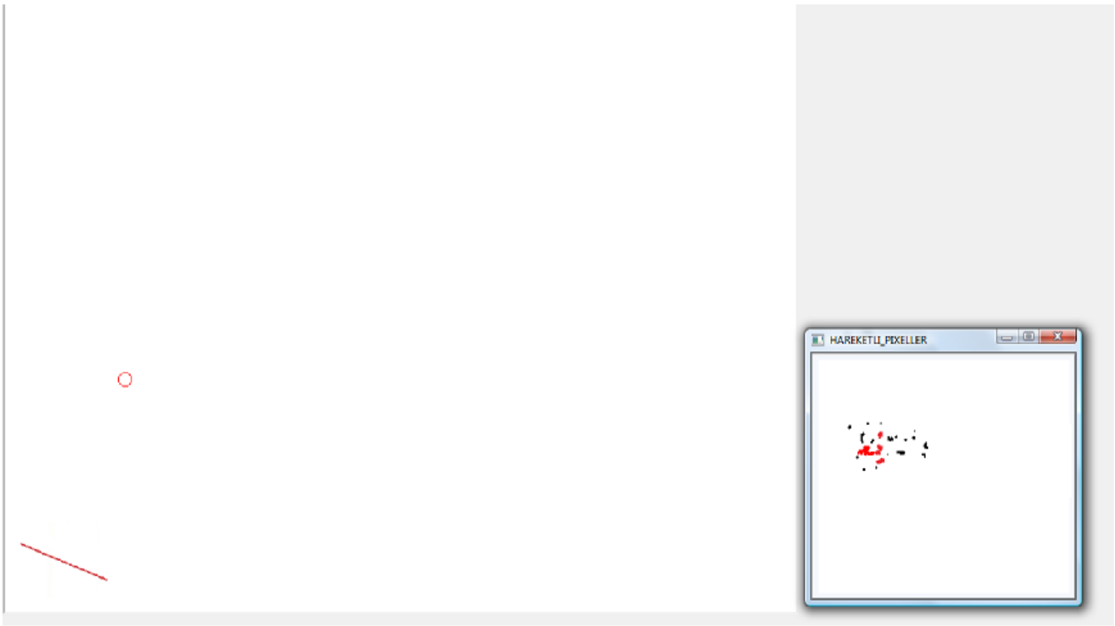
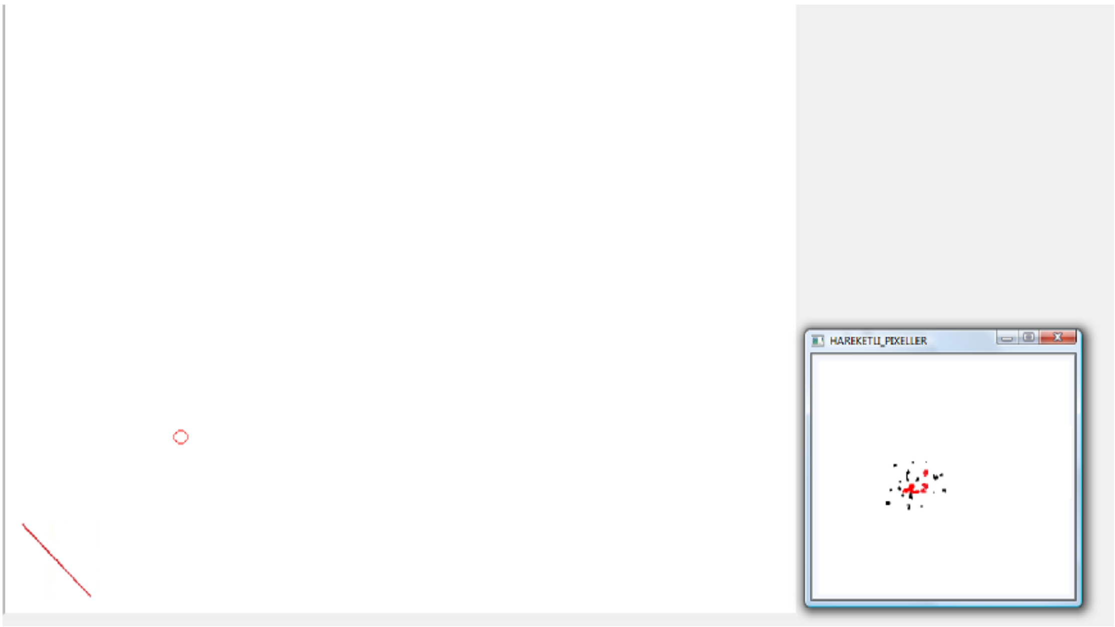


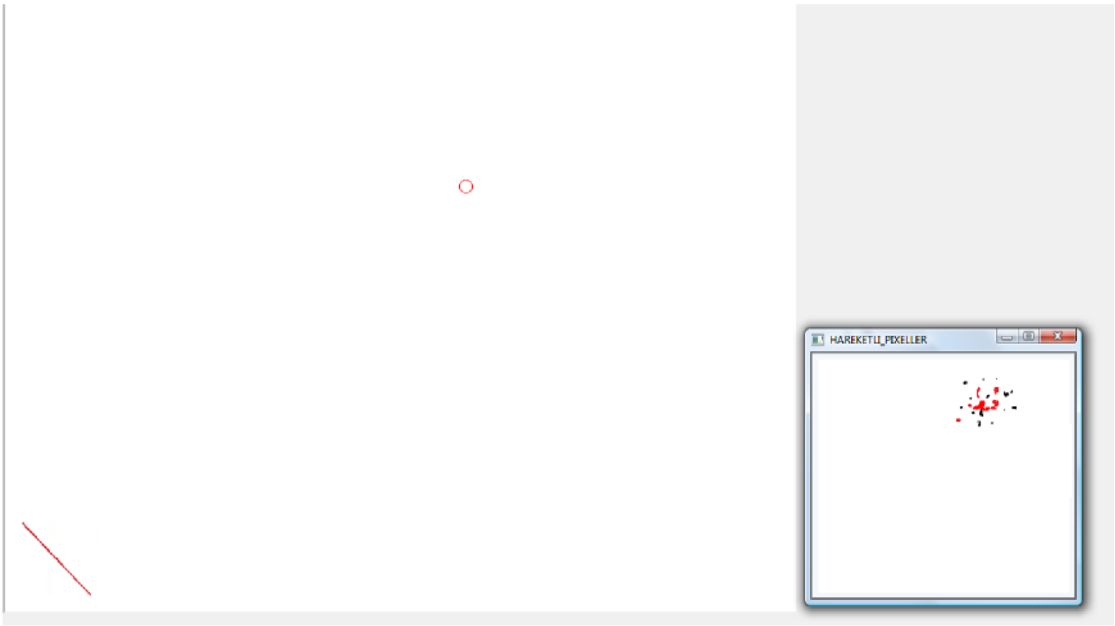
Figure A.2: When the system perceives the motion.

The figures shown below represent some prompter's screen direction at any instant times.




HAREKETLI_PIXELLER


HAREKETLI_PIXELLER


HAREKETLI_PIXELLER