# Internship Presentation

## Semantic Saliency of Video Content: Application to Quality Assessment and Efficient Video Retrieval

Ho Tien Lam

Master of Software Engineering (2009-2011)
Pôle Universitaire Français - Ho Chi Minh City
LaBRI - University of Bordeaux 1
Supervisor: Prof. Jenny Benois-Pineau

February 13, 2012

# Outline

## Internship

- Time: 6 months (July 2011 – December 2011)
- Place: LaBRI

## LaBRI (Laboratoire Bordelais de Recherche en Informatique)

- A French research laboratory in field of computer science
- 6 departments:
  - Combinatorics and Algorithms
  - Image and Sound
  - Languages, Systems and Networks
  - Formal Methods
  - Models and Algorithms for Bio-informatics and Data Visualization
  - Supports and Algorithms for High Performance Numerical Applications

## Video Analysis and Indexing Group

- I worked as a trainee in Image and Sound team.
- 4 research groups inside Image and Sound team:
  - Structuring and analysis of images
  - 3D modeling, visualization and interaction
  - Video analysis and indexing
  - Modeling of sound and music
- My internship took place in the Video Analysis and Indexing group (for short AIV group)

## Fact

- Human beings do not pay equal attention to all exposed visual information.
- They only focus on certain areas known as focus of attention (FOA) or saliency regions.

## Context of project

The LaBRI's researchers have proposed a novel objective quality assessment metric using spatio-temporal saliency map.

## Aim of my internship

To help in studying if a semantic saliency, based on faces, could improve this metric.

# Process Overview

- To enhance the existing method, I have studied how to create semantic saliency maps.
- The semantic saliency map is generated in two ways: manual way and automatic way.

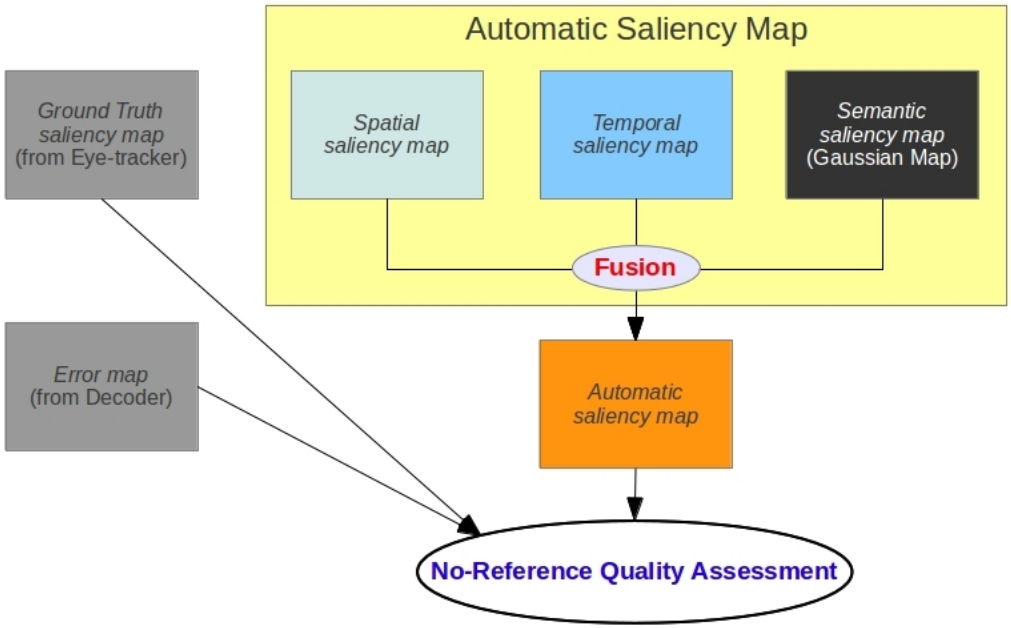



Figure: Process Overview

# Saliency Map

## Definition

A saliency map is used for representing the saliency areas in an image and guiding the selection of attended locations, based on the spatial distribution of saliency.

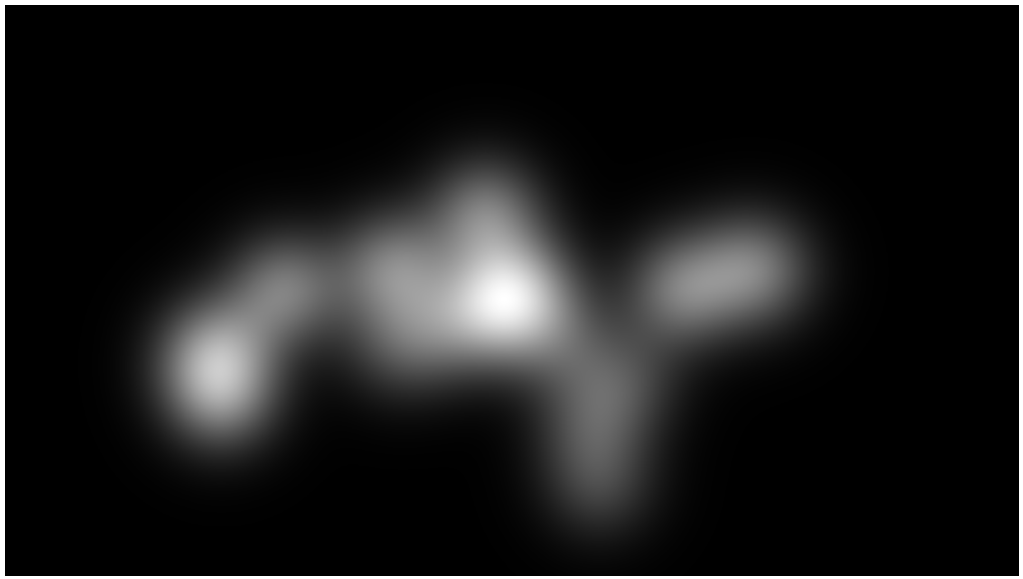


Figure: Example of Saliency map

# Gaussian Map

## Definition

A Gaussian map is a gray-scale saliency map which assigns a saliency value to each pixel. This value is calculated by using the formula of Gaussian function.

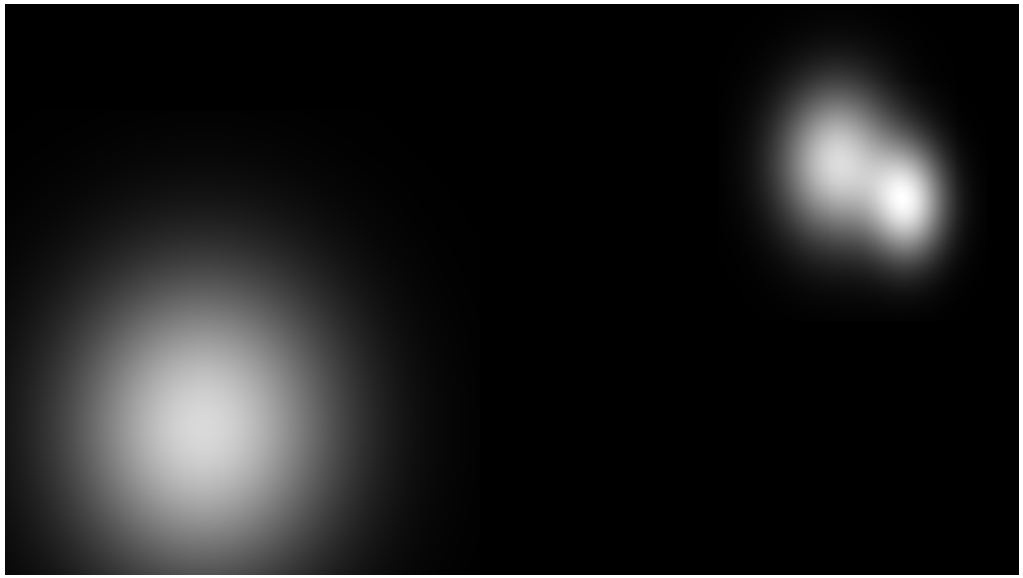


Figure: Example of Gaussian map

# Bounding Box

## Definition

A bounding box is a rectangle which surrounds an object of interest. It is used for representing a semantic saliency region in an image. These boxes may be generated manually by using BBBox Annotation Software or automatically by face detection algorithm.


Figure: Example of Bounding box

# Goal & Tasks

## Goal of my training placement

To create "semantic saliency" maps of HD video which express the presence of a semantic object–"face".

# Goal & Tasks

## Goal of my training placement

To create "semantic saliency" maps of HD video which express the presence of a semantic object–"face".

## Tasks

1. Adapt BBBox annotation software for annotating HD videos.

# Goal & Tasks

## Goal of my training placement

To create "semantic saliency" maps of HD video which express the presence of a semantic object–"face".

## Tasks

1. Adapt BBBox annotation software for annotating HD videos.
2. Annotate the ground truth: trace the bounding boxes of faces and store their coordinates observed in video frames by human.

# Goal & Tasks

## Goal of my training placement

To create "semantic saliency" maps of HD video which express the presence of a semantic object–"face".

## Tasks

1. Adapt BBBox annotation software for annotating HD videos.
2. Annotate the ground truth: trace the bounding boxes of faces and store their coordinates observed in video frames by human.
3. For manually annotated bounding boxes of faces, we compute the saliency maps with the help of Gaussian filtering.

# Goal & Tasks

## Goal of my training placement

To create "semantic saliency" maps of HD video which express the presence of a semantic object–"face".

## Tasks

1. Adapt BBBox annotation software for annotating HD videos.
2. Annotate the ground truth: trace the bounding boxes of faces and store their coordinates observed in video frames by human.
3. For manually annotated bounding boxes of faces, we compute the saliency maps with the help of Gaussian filtering.
4. Detect faces automatically by using Viola-Jones face detector of OpenCV library.

- The BBBox software is developed in LaBRI on the basis of Viola–Jones object detection algorithm.
- Written in C++, and it uses Qt[1] framework to build graphical user interface (GUI).
- The aim of this task is to adapt software GUI for annotating HD videos.

[1] http://qt.nokia.com

# Functionality of BBBox software

- BBBox is used for annotating the human faces in each frame of a video.

# Functionality of BBBox software

- BBBox is used for annotating the human faces in each frame of a video.

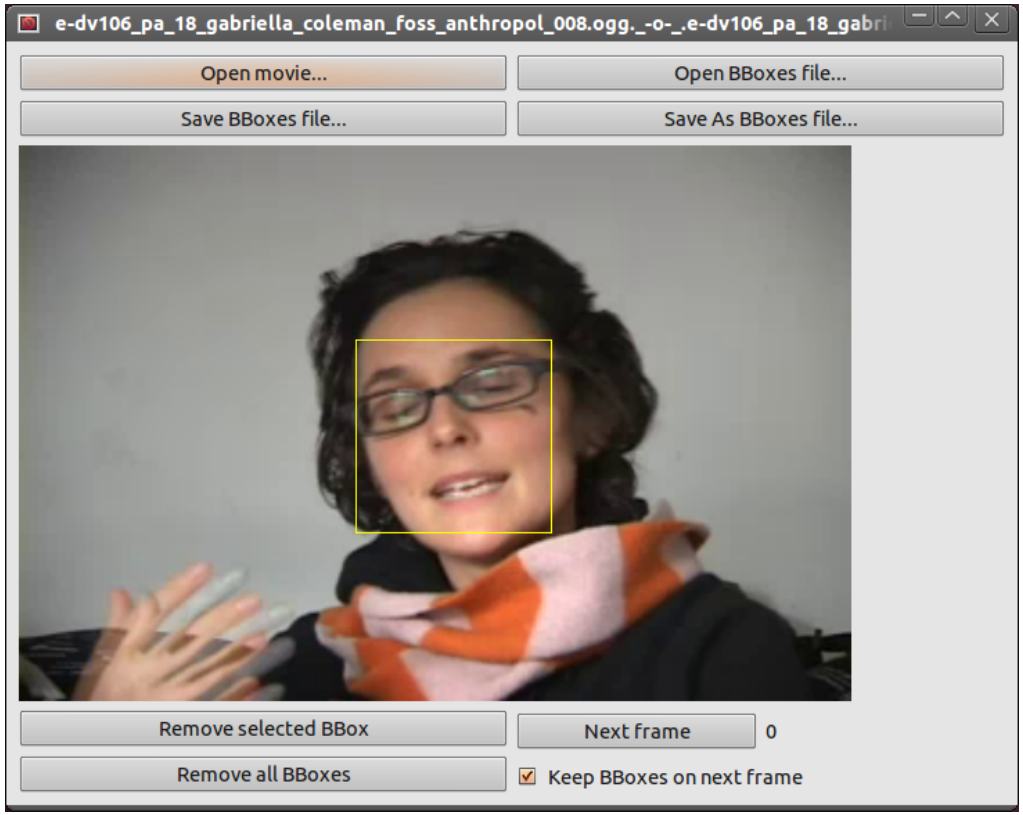



Figure: Screenshot of BBBox

# Output of BBBox software

## BBoxes file

- After annotating some frames of the video, user can save the current result to a file called BBoxes file.
- BBoxes file is just a text file which contains one line per processed frame.

# Output of BBBox software

## BBoxes file

- After annotating some frames of the video, user can save the current result to a file called BBoxes file.
- BBoxes file is just a text file which contains one line per processed frame.
- Each line of text has the following format.

$$frameNb \quad nbBBoxes \quad xMin_1 \quad yMin_1 \quad width_1 \quad height_1 \ldots xMin_n \quad yMin_n \quad width_n \quad height_n$$

| | |
|---|---|
| $frameNb$ | The frame number containing bounding boxes |
| $nbBBoxes$ | The number of boxes |
| $xMin_1, yMin_1$ | The position of first box's top-left corner |
| $width_1, height_1$ | The dimension of first box |

# Missing Features of BBBox software

1. When user opens a video with resolution bigger than that of desktop screen, the actual version of BBBox cannot adjust the window size to fit well in desktop screen.

# Missing Features of BBBox software

1. When user opens a video with resolution bigger than that of desktop screen, the actual version of BBBox cannot adjust the window size to fit well in desktop screen.
2. The software does not scale the movie scene up or down when a user resizes the window.

# Missing Features of BBBox software

1. When user opens a video with resolution bigger than that of desktop screen, the actual version of BBBox cannot adjust the window size to fit well in desktop screen.
2. The software does not scale the movie scene up or down when a user resizes the window.

The $1^{st}$ problem was corrected completely.
The $2^{nd}$ problem was partially solved. It was possible to scale the image up, while the act of scaling down was not successful.

# Hierarchy of Classes in BBBox software

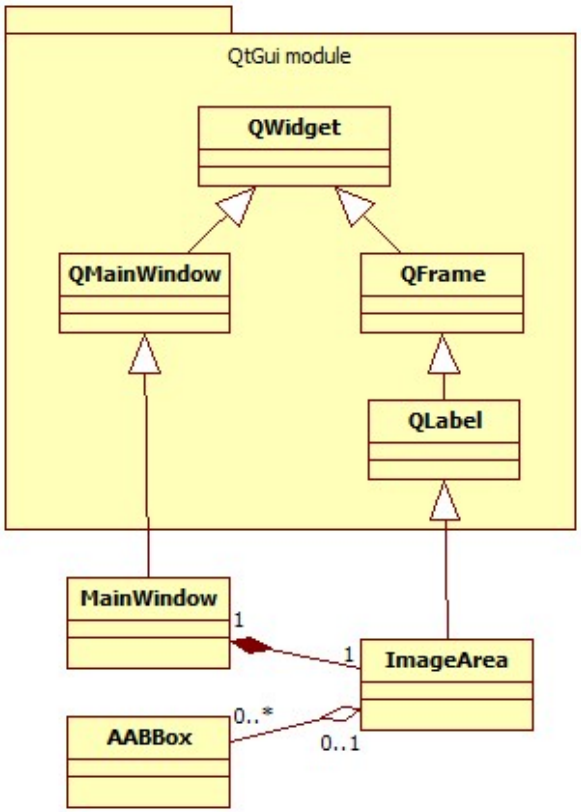



Figure: Class diagram of BBBox

# Hierarchy of Classes in BBBox software

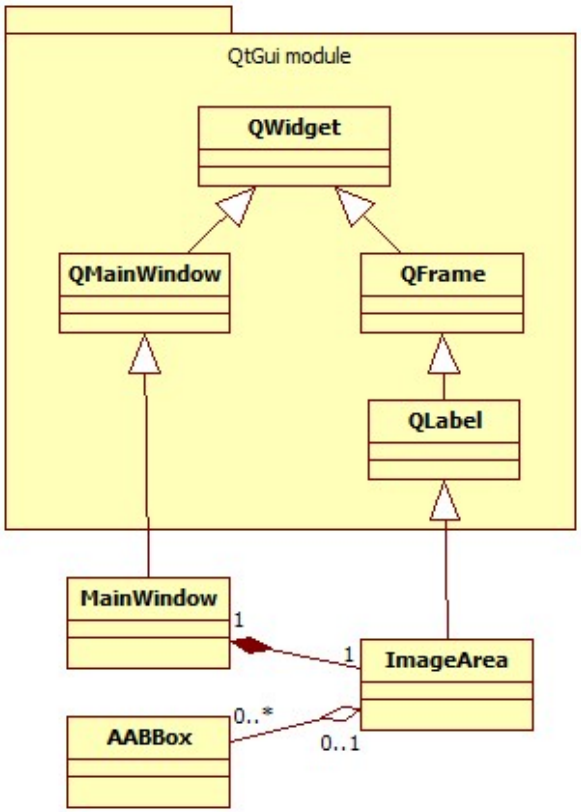



Figure: Class diagram of BBBox

## QWidget

The base class of all user interface objects.

## QMainWindow

Provides a main application window.

## QFrame

The base class of widgets that can have a frame.

## QLabel

Provides a text or image display.

# Hierarchy of Classes in BBBox software

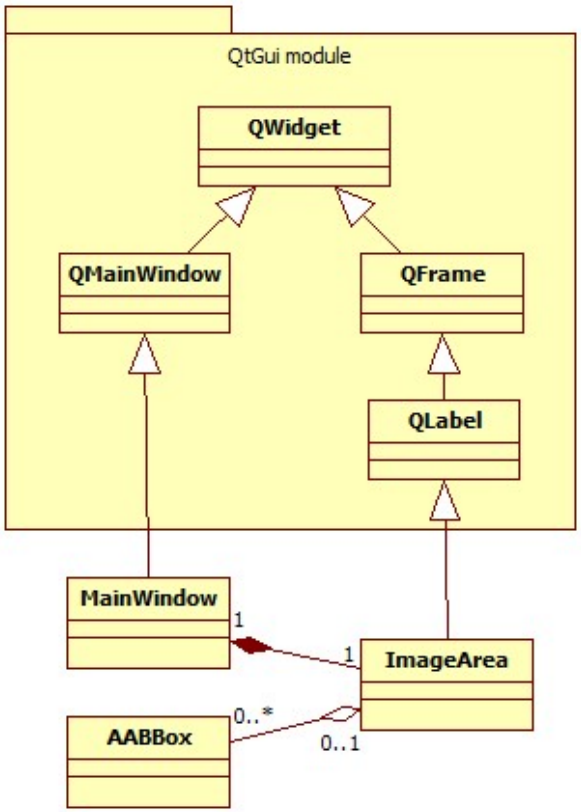



Figure: Class diagram of BBBox

## MainWindow

Takes charge of building BBBox GUI, receiving events, and event-handling.

## ImageArea

Takes charge of displaying each video frame on main window and dealing with events.

## AABBox (Axis-Aligned Bounding Box)

This class defines the attributes of a bounding box and some utility methods.

# Modification of BBBox software

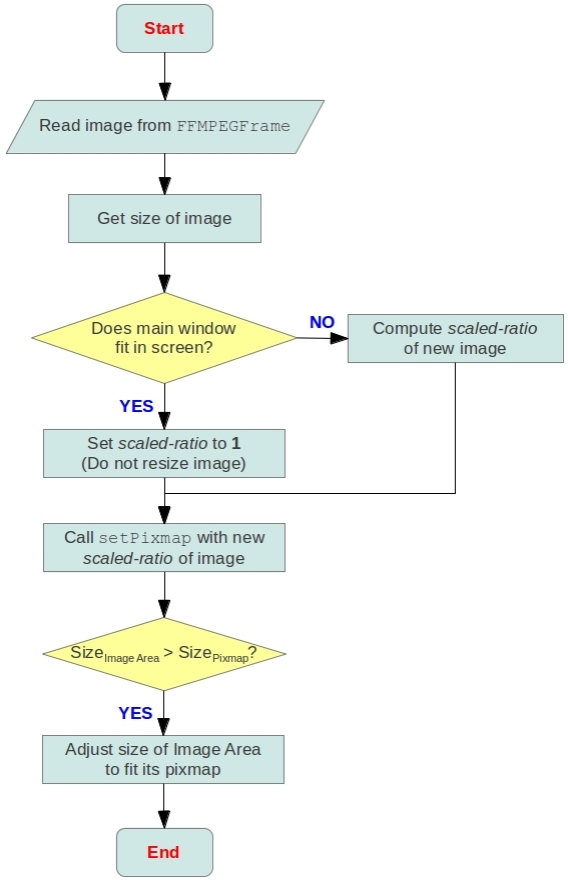



Figure: Flowchart for solving the first problem

# Modification of BBBox software

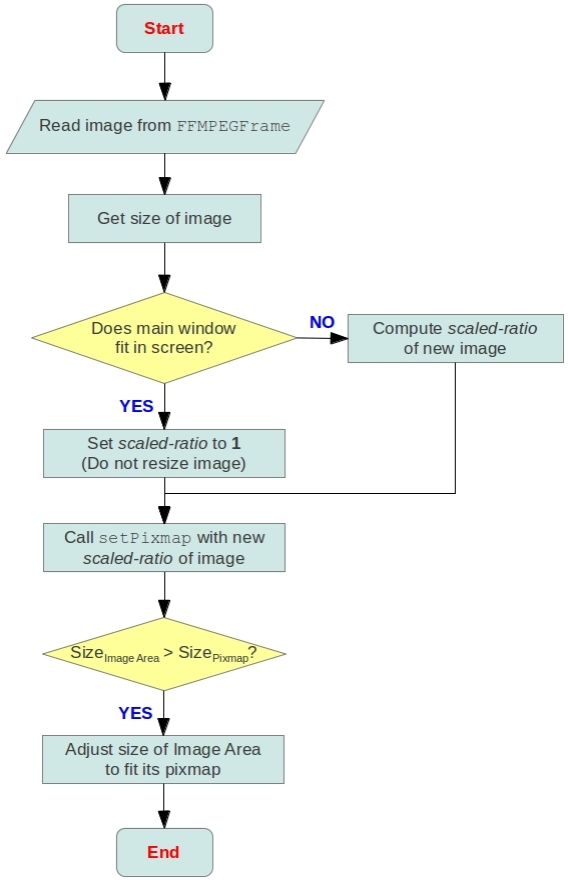



Figure: Flowchart for solving the first problem

## Difficulty

We have to calculate main window size before calling `setPixmap()` method on `ImageArea`.

## Solution

Compute approximately the main window size. This computation is done with dimension of displayed image, button height, height of title bar, spacing and margin values of layout.

# Modification of BBBox software

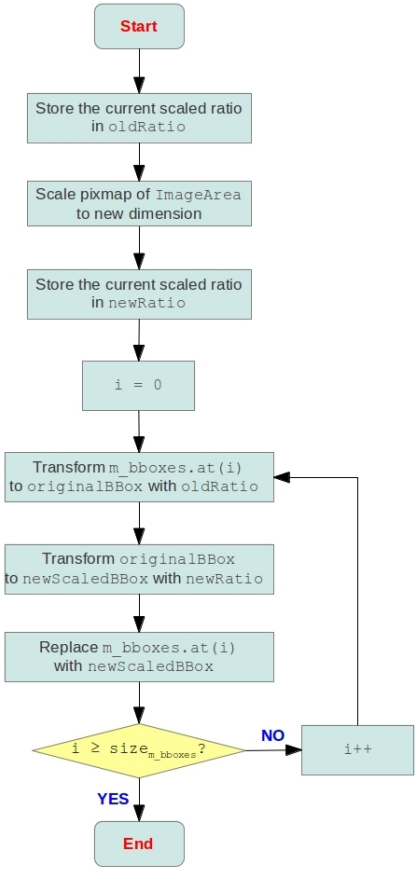



Figure: Flowchart for solving the second problem

# Modification of BBBox software

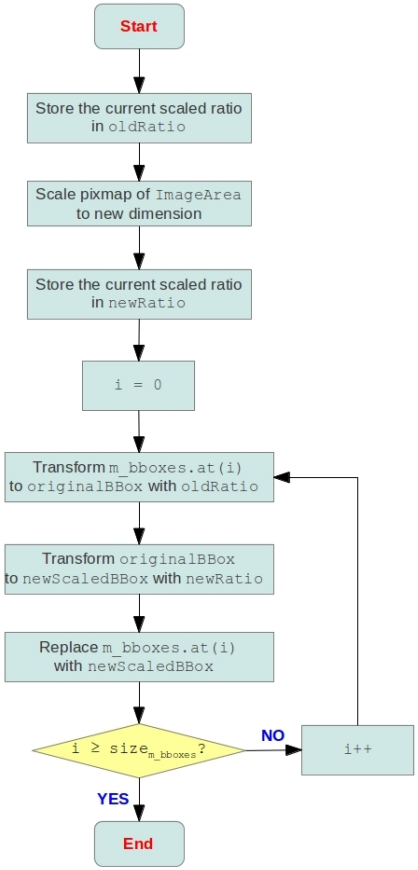



Figure: Flowchart for solving the second problem

We need to scale not only the `pixmap` of `ImageArea` but also the bounding boxes displayed on this `pixmap`.

The reason is that there are **2 different coordinates of a bounding box**. One coordinate is used for displaying on the scaled `pixmap`, and the other is used for storing in BBoxes file.

- Gaussian map, one kind of saliency map, is generated by using the results of annotating videos and Gaussian function. (Review the "Goal & Tasks" slide by clicking here)
- Use OpenCV library in processing images.
- OpenCV[2] (Open Source Computer Vision) is an open-source library that includes hundreds of computer vision algorithms.

[2] http://opencv.willowgarage.com/wiki/

## Annotate Sources of Videos

- The aim of this task is to simulate a perfect face detection.
- We annotate several sources of videos by using the modified version of BBBox software.
- The result of annotating videos is 1 BBoxes file per video.

## Annotate Sources of Videos

- The aim of this task is to simulate a perfect face detection.
- We annotate several sources of videos by using the modified version of BBBox software.
- The result of annotating videos is 1 BBoxes file per video.

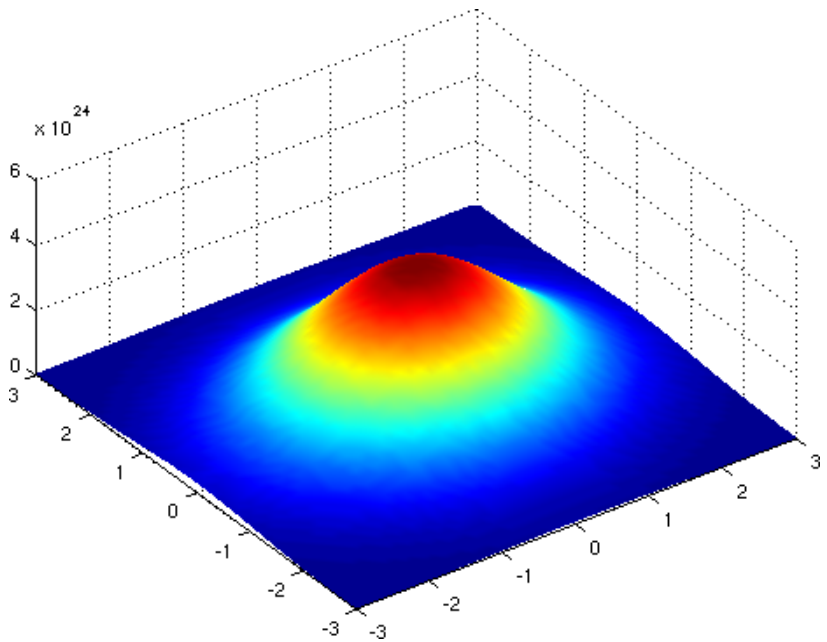
## Generate Gaussian Map

- Gaussian functions are widely used in statistics where they describe the normal distributions, in signal processing where they serve to define Gaussian filters.
- For creating Gaussian map, we used a particular form of two-dimensional Gaussian function.

# 2D Gaussian functions

## Formula

$$f(x,y) = Ae^{-\left(\frac{(x-x_0)^2}{2\sigma_x^2}+\frac{(y-y_0)^2}{2\sigma_y^2}\right)} \quad (1)$$




In (1), the coefficient $A$ is the amplitude, $(x_0, y_0)$ is the center coordinate and $\sigma_x, \sigma_y$ are the x and y dimension of the blob.

Figure: Example of 2D Gaussian functions

# Processing Steps

Use BBBox software to annotate faces in video frame

# Processing Steps

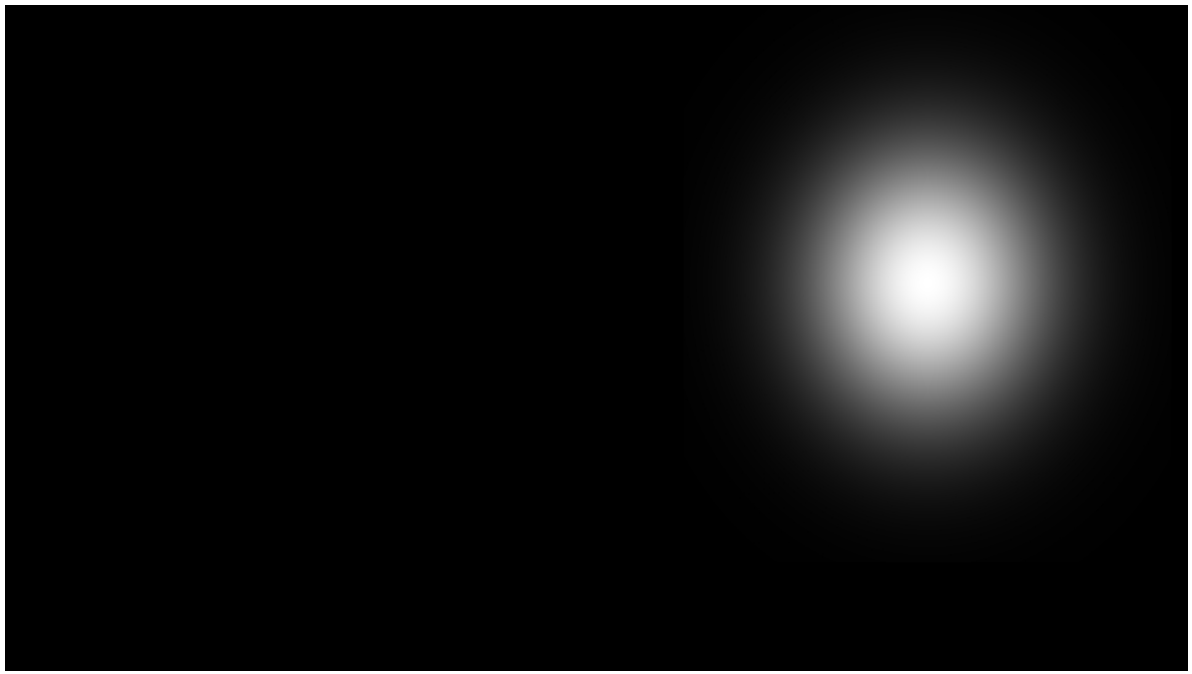
Use BBBox software to annotate faces in video frame

# Processing Steps

Use 2D Gaussian function to create the gray-scale saliency map

# Processing Steps

Use 2D Gaussian function to create the gray-scale saliency map



- The purpose of this task is to add automatic face detection for the automatic semantic map construction.
- Use Viola–Jones object detection algorithm in this task.
- This algorithm is implemented in an example program coming with OpenCV library.
- With the result of automatic face detection, we also construct Gaussian maps after generating BBoxes file from detection result.

## Conclusion

The goal of my internship has been achieved by:

- Modifying BBBox software
- Annotating videos
- Generating Gaussian maps
- Automatic Face Detection

## Future Work

- We plan to improve the automatic detection by using face tracking algorithm.
- Evaluate if the automatically-produced saliency maps, including semantic information, allow better video quality assessment.

Thank you for your attention!

**Q & A**