

**Silesian University of Technology**

**Faculty of Automatic Control, Electronics   
and Computer Science**

##### Final Project

##### (choose appropriate)

Implementation of face detection algorithms in video sequences

Author: Wojciech Mazur

Supervisor: PhD Michał Staniszewski

Consultant: PhD Name Surname

Gliwice, January 2019Załącznik Nr 2 do Zarz. Nr 97/08/09

**Oświadczenie**

Wyrażam zgodę/nie wyrażam\* zgody na udostępnienie mojej pracy dyplomowej/rozprawy doktorskiej\*

…………….………., dnia …………………………

……………………………………..……………….……

*(podpis)*

……………………………………………………………

*(poświadczenie wiarygodności podpisu przez Dziekanat)*

*\* właściwe podkreślić*

**Oświadczenie promotora**

Oświadczam, że praca „Implementation of face detection algorithms in video sequences” spełnia wymagania formalne pracy dyplomowej inżynierskiej.

|  |  |
| --- | --- |
| Gliwice, dnia ……………………… | ………………..……………….……  *(podpis)* |

Contents

[1. Introduction 3](#_Toc526891482)

[2. [Problem analysis] 5](#_Toc526891483)

[3. Requirements and tools 13](#_Toc526891484)

[4. External specification 16](#_Toc526891485)

[5. Internal specification 19](#_Toc526891486)

[6. Verification and validation 22](#_Toc526891487)

[7. Conclusions 25](#_Toc526891488)

[Bibliography **Error! Bookmark not defined.**](#_Toc526891489)

[List of abbreviations and symbols iii](#_Toc526891490)

[Contents of attached CD-ROM iv](#_Toc526891491)

[List of Figures v](#_Toc526891492)

[List of Tables vi](#_Toc526891493)

# Introduction

This chapter contains following elements:

* introduction into the problem domain,
* settling of the problem in the domain,
* objective of the thesis,
* scope of the thesis,
* short description of chapters,
* clear description of contribution of the thesis’s author – in case of more authors table with enumeration of contribution of authors.

According to Maslow’s Hierarchy of needs, safety is one of our most fundamental needs. Without it, it is hard to think about friends, relationships, accomplishments, or self-fulfillment. Over the last few years it can be observed, that the biological identification is rising in its popularity. There are new ways found to utilize human biological footprints. What can be noticed, is that most of the use cases revolves around security, and for a good reason. Fingerprints are used to unlock locks for decades. Fingerprint scanners have also found a way to consumer electronics. World with securing our smartphone or a notebook with just a password can no longer be imagined. Fingerprint scanners have expended rapidly when it comes to smartphones and made our lives better.

But it’s not the only way people are trying to secure our privacy, including our increasingly valuable resource, which is data. There are ways developed to recognize people by other features like iris, or even from a face in general, using different kinds of sensors. Cameras, infrared light sensors to depth detectors that remember given points on a face. And everything happens within a blink of an eye.

What people want to do apart from protecting our data, is something that humanity crave to do since beginning of itself. To protect ourselves from external harm. Feel safe and comfortable, which is, in fact, required, as stated previously.

One of the main issues arising in recent years, is public safety and detection of dangerous people. Increasing activity of terrorism in a lot of European countries can be observed. The awareness of people is also increasing, it is being learned that prevention is better that treatment. National safety agencies recognize most of European countries at least on moderate threat of terrorism. A lot of countries are considered even as a high threat for regular travelers. This is something that needs to be addressed as quickly as possible, and it is needed to work on the technology, that can be effectively applicated for new solutions.

## Goal of the thesis

Main goal of this thesis is to implement face detection algorithms, both with the face recognition. This could provide us with useful tools to analyze and examine, for example video footages that are collected every day on thousands of city cameras around the globe. That existing architecture can be used, not only to look for people that have already broke the law, but also to prevent crime. Simple camera system doesn’t provide us with the possibility of detecting the face. It would be needed to do that manually which is a complicated and tedious work, that not every person is able to do. The main focus is to accomplish filtering dangerous, or at least for some reason crucial for safety people, implementing algorithms that detect and recognize their faces.

## Scope of work

Main scope of the work contains a few elements that needed to be combined to give full spectrum of the problem. First of all, I needed to choose programming language, that would be easy to use, yet contain all the elements needed to process images, like easily accessible and usable libraries that would greatly accelerate the progress of work. Then, analysis of current solutions, and choice of a few algorithms that comply with what this thesis is trying to achieve and comparison of their effectiveness. Next thing was the actual implementation of the solution in the programming language of choice, but also processing the testing material consisting of video sequences that were analyzed and graphically modified that the human can easily observe effects of the detection and recognition in real time.

# [Problem analysis]

This chapter contains following elements:

* problem analysis,
* state of the art, problem statement,
* literature research (all sources in the thesis have to be referenced [1, 2, 4, 3]),
* description of existing solutions (also scientific ones, if the problem is scientifically researched), algorithms, location of the thesis in the scientific domain.

## Face detection methods

There are a lot of ways to detect given features, including face features, that would allow us to recognize position of the face on a given media sample. Although one that got very popular since its release is based on machine learning approach, from the work of P. Viola and M. Jones [1] [2]. Their work is described as extremely rapid in terms of effectiveness, and this is something that is crucial for effective and real time face recognition. The original study was performed on Intel Pentium III clocked on 700MHz, which is a rather old processing unit considering performance of newest units. This would give over 15fps on 384x288 pixel images. Similar CPUs will be compared using tool UserBenchmark [3]. Comparing closest processor that this tool provides us, which is a successor to Pentium III, Pentium 4. Its performance can be taken as comparable, especially in units that were on similar clock speeds [4]. Unit that was used for calculations in this thesis was Intel Core i5 4690k, which is according to UserBenchmark, over ten times faster on average, often going up to twenty times faster and above [3]. This gives us the opportunity to take higher resolution of video samples and photos, and still achieve a very satisfying frames per second rate.

Idea is that if given element, item or other object has a unique shape, it can be assumed, that it also has unique properties when it comes to light that is projected onto that object. This gives us a tool to check whether chosen parts of an image have lighting that is changing in a way that was previously seen and is recognize for example as a face.

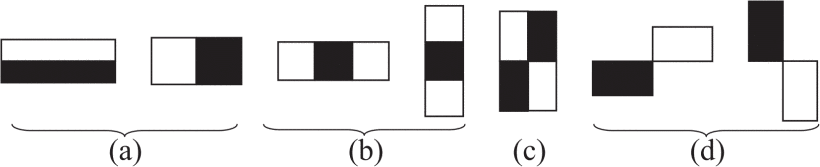


Figure 1Basic Haar-like features [1]

On the Figure 1 is can be seen what features are used to distinguish between light and dark spots on a picture. Although this can be misleading, what should be really be seen if understanding of the problem is a priority, is to see the image in a way shown in Figure 2.



Figure 2Haar-like feature on face [5]

The colors itself are not important, but the pixel values within. For example, if face is to be found, it can be safely assumed, that forehead pixels will be on average brighter than pixels that are within eyes region, due to the shadow dropping from the eyebrows. The same goes mostly for things like nose, where either vertical line, that is bright and has darker surroundings due to the shadows. But this is also something that needs to be taken with care, due to the changing light conditions. Compensation can be done, by looking for white line that has a darker region only on one side.

There were taken many approaches to improve initial Viola and Jones, including the Viola himself. One of the first successful improvements were made only one year later, by the R. Lienhart and J. Maydt [6]. They were dealing with two main problems of original approach, by adding 45 rotated features and adding new optimization procedures for improved performance.

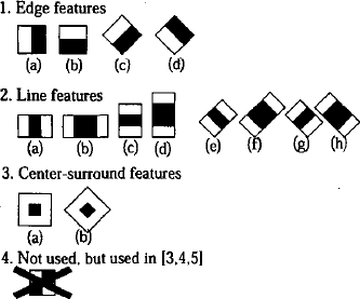


Figure 3Extended features proposed by Lienhart and Maydt [6]

Further attempts of improvements can be observed. T. Mita, T. Kaneko and O. Hori proposed and derived method of detecting co-occurrence of features. One of the things that are addressed by those authors, is that in original solution, after detecting one feature, detecting of proceeding ones comes with much higher error rate [7].

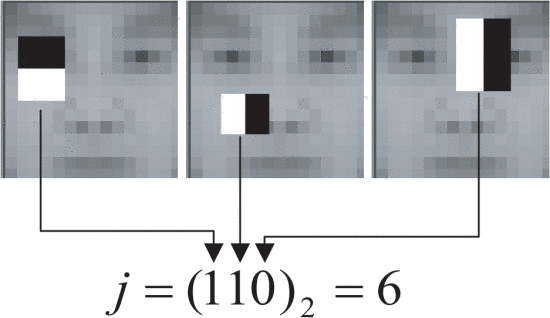


Figure 4 Joint Haar-Like feature

This approach allows to capture more of a structural similarity of faces. Experiment results give reduced error rates and improved performance, even when considering bigger number of features, than in Viola and Jones solution.

## Face recognition methods

Face recognition is a topic that is highly researched for decades, first by manual labor, where for example physical photos were projected onto photomultiplier matrix, then small motors were turned according to illumination [8], which is in fact a machine learning approach. Since then, great amounts of methods were developed to recognize patterns, that includes faces [9]. Numerous approaches that are holistic, hybrid, feature-based, artificial neural networks, fuzzy-based, generic-algorithms based and more can be found [9], where variance in implementation can mixing different approaches can also differ. Some ways of recognition would estimate face position, for instance, to be able to transform image in given space, to obtain more accurate results.

It is also required, for a reasonable efficiency to optimize and normalize dataset from which the algorithm will be built upon. The significant matter is to provide good quality images, for face recognition straight face is a suggested look, without rotation and with the same scale as the rest of the photos, also with an equal lighting [10]. But it is not always possible to provide such resources. Good practice would be to implement image processing that equalize the appearance of the given object.

Next general task in facial recognition is to extract features [9]. Dimensionality of the image is needed to be reduced, so that the data that needs to be processed is not that enormous, and yet the important features are conserved.

### Holistic approaches

Holistic ways of recognizing faces are dealing with the problem with comprehensive approach to processing facial verification, taking data from image as a whole block. They consider small number of features and try to envelop components of an image. Later, these components are used to verify similar shapes on other data [11].

#### Eigenfaces

When considering sets of images, it can be seen that great amount of data can easily arise. Reasonable and natural step would be to reduce its size. The advantage of this approach is that big dataset is not needed, although the bigger is usually the better. Eigenface is a method that is extracting features into vectors, and then represent then in a form of covariance matrix. Having those vectors, it can be calculated how distant they are [12] [11].

However, in this approach training data set is required to be processed, otherwise results will be not satisfactory. It is needed to equalize lighting, and align image, it is also preferred to remove background and other noise that could have negative influence. This algorithm is using PCA (Principal Component Analysis) to reduce dimensions and find vectors. Vector defines subspace – face space, and training set is projected, to find weights, or similarities in other words.

#### Fisherfaces

Fisherfaces are similar to Eigenfaces. Method uses both principal component analysis and linear discriminant analysis and produce a subspace projection matrix. It is minimizing variation within classes and maximizing separation. It should provide better results when it comes to different facial expressions and light variance. Although computation time is more complex, and time needed is greater [13].

### Hybrid approaches

These methods are based on mixing holistic approach and feature-based matching. These two methods have their advantages and disadvantages, but the idea is that combination of those two might give a better result and eliminate each other disadvantages [14] [15].

A lot of hybrid approaches draw from the Gabor wavelets that were acknowledged as a reliable local feature extraction method, due to its sturdiness when it comes to light, distortions, translation [15].

### Feature-based

Every face has some distinctive features related to some regions that can be extracted. Eyes, nose, mouth, cheeks can be used to classify a face. Complexity of these solutions are often hidden under recognizing particular parts of the face. For example, eye processing could be interrupted by reflexes on the iris, that should be eliminated, eyebrows are often described as a parabola approximation. Nose is one of the simplest properties that can be used as a base, where gray levels contrasts are great within neighbor regions. Mouth shape can be also described as a function. Knowing that many facial points chin location can be also estimated. Approach like this was described in [16] where efficiency was often exceeding 90 percent.

# Requirements and tools

This chapter contains following elements:

* functional and nonfunctional requirements,
* use cases (UML diagrams),
* description of tools,
* methodology of design and implementation.

.

# External specification

This chapter contains following elements:

* hardware and software requirements,
* installation procedure,
* activation procedure,
* types of users,
* user manual,
* system administration,
* security issues,
* example of usage,
* working scenarios (with screenshots or output files).

The entire document should contain references to the illustrations contained therein (Fig. 4.1).

|  |
| --- |
|  |
| Fig.4.1. *The variation funkstioni* |

# Internal specification

This chapter contains following elements:

* concept of the system,
* system architecture,
* description of data structures (and data bases),
* components, modules, libraries, resume of important classes (if used),
* resume of important algorithms (if used),
* details of implementation of selected parts,
* applied design patterns,
* UML diagrams.

A short code insertion in the text line is possible, e.g. class Main. Longer fragments should be written in *Courier* or *Courier New* font size 10 in frames (Listing 4.1) with a space between the lines of the value 1. All lines of code should be numbered so that they can be referenced in the text of the document.

Listing 1. Generating random numbers

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17 | **package** polsl.iinf.lab;  **import** java.util.Random;  **public** **class** Main {  **public** **static** **void** main(String[] args) {  Random r = **new** Random();  // drawing a number from the range 1..10  **int** a = r.nextInt(10 + 1);  System.*out*.println(a);  // drawing a number from the range -5..15  System.*out*.println(r.nextInt(21) - 5);  }  } |

# Verification and validation

This chapter contains following elements:

* testing paradigm (eg. V model),
* test cases, testing scope (full / partial),
* detected and fixed bugs,
* results of experiments (optional).

# Conclusions

This chapter contains following elements:

* achieved results with regard to objectives of the thesis and requirements,
* path of further development (eg. functional extension . . . ),
* encountered difficulties and problems.

# Bibliography

|  |  |
| --- | --- |
| [1] | P. Viola i M. Jones, „Rapid object detection using a boosted cascade of simple features,” w *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, Kauai, 2001. |
| [2] | P. Viola i M. Jones, „Robust Real-Time Face Detection,” *International Journal of Computer Vision,* tom 57, nr 2, p. 137–154, 2004. |
| [3] | "userbenchmark," [Online]. Available: https://cpu.userbenchmark.com/Compare/Intel-Pentium-4-160GHz-vs-Intel-Core-i5-4690K/m15237vs2432. [Accessed 29 12 2018]. |
| [4] | justinwl, "ancientelectronics," [Online]. Available: https://ancientelectronics.wordpress.com/tag/pentium-4-vs-pentium-iii/. [Accessed 29 12 2018]. |
| [5] | N. Ipe, "N Recursions," 16 10 2018. [Online]. Available: http://nrecursions.blogspot.com/2018/10/a-better-tutorial-on-haar-features-used.html. [Accessed 29 12 2018]. |
| [6] | R. Lienhart i J. Maydt, „An extended set of Haar-like features for rapid object detection,” w *Proceedings. International Conference on Image Processing*, Rochester, 2002. |
| [7] | T. Mita, T. Kaneko i O. Hori, „Joint Haar-like features for face detection,” w *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, Beijing, 2005. |
| [8] | W. K. Taylor, „Machine learning and recognition of faces,” IET, London, 1967. |
| [9] | M. &. B. D. Roomi, „A Review Of Face Recognition Methods,” *International Journal of Pattern Recognition and Artificial Intelligence,* tom 27, nr 4, 2013. |
| [10] | T. Shakunaga i K. Shigenari, „Decomposed eigenface for face recognition under various lighting conditions,” w *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Kauai, 2001. |

# List of abbreviations and symbols

|  |  |
| --- | --- |
| *DNA* | deoxyribonucleic acid |
| *MVC* | model–view–controller |
| *N* | cardinality of data set |

# Contents of attached CD-ROM

The thesis is accompanied by a CD-ROM containing:

* thesis (PDF file),
* source code of applications,

# List of Figures

# List of Tables