Introduction to Computer Vision from Automatic Face Analysis Viewpoint[1]

Erno Mäkinen (etm@cs.uta.fi)
Department of Computer Sciences
University of Tampere, Finland

1 Computer Vision

The computer vision field is rather extensive. It has applications from industry to homes. However, many of the underlying processes and techniques are the same for all application areas. Next, an overview of these processes and techniques is given. Digital image acquisition and processing are the first topics since they form the basis for higher level processing, such as pattern recognition, when computer vision is considered. Machine learning and pattern recognition are the second two topics, since they are also applied extensively in computer vision. In fact, many machine learning and pattern recognition techniques, such as neural networks and support vector machines (SVM) are also used in many other fields than computer vision.

1.1 Digital Image Acquisition and Processing

Digital image acquisition is the first step in any computer vision system. There are several ways to acquire an image. The image may be acquired in visible, infrared, ultraviolet, x-ray, gamma-ray, and radio-wave bands or it may be formed from sound as in medical ultrasound imaging or from some other source. The visible band is a fairly natural choice for human-computer interaction. However, visible band is by no means the only medium for use in HCI. For example, infrared sensors are typically used in gaze tracking.

When a light source emits light, the light is partially absorbed and partially reflected from the objects in the scene. The camera senses a part of the light in the scene when the light passes through the camera lens and the lens refracts the light to the sensors that transform the sensed visible light (or some other energy) into electrical form, voltage. The intensity of the light determines the strength of the voltage.

[1] Modified from the original text published in: Mäkinen, E. (2007), Face Analysis Techniques for Human-Computer Interaction, *Phd Thesis*, University of Tampere.

Digital cameras have either CMOS (Complementary Metal–Oxide–Semiconductor) or CCD (Charge-Coupled Device) arrays that sense light and transform it into voltage. The array is a group of sensors arranged in a grid. The number of sensors in the array depends on the camera but a typical web camera has 640*480 size array and, for example, Canon PowerShot SD600 pocket camera has the largest image size of 2,816*2,112 pixels (6 Megapixels) meaning that the CCD sensor array is close to that size too.[2]

After the sensed light has been transformed to the voltage it is further quantized. Quantization means that voltages at a certain range are defined to have a certain same value. For example, there could be 256 distinct values for the quantized voltage. Finally, after quantization the image is in digital form and can be stored or further processed.

The low-level image processing may include filtering the image in spatial or frequency domain, doing histogram equalization for it or transforming its intensities by a log-transform, for example. The common purpose of low-level processing is to enhance the image (Gonzalez and Woods, 2002, pp. 25-28). There can also be some morphological processing and segmentation of the image parts. Not all this processing does need to happen before higher-level processing such as face recognition. Instead, the processes are usually interleaved. For example, faces can be detected from an image using a pattern recognition algorithm and after the faces have been detected histogram equalization can be performed for each of them.

Histogram equalization spreads the intensity values of the images over a larger range. It is used because it decreases the effect of different imaging conditions, for example different camera gains and it may also increase image contrast (Rowley et al., 1998). When analyzing faces it is important that there is as little variation due to external conditions (such as imaging conditions) as possible, so that the variations between the faces become more visible, that is the issue we are interested in.

The classifier and data representation used as input to a classifier determines if histogram equalization should be used. For pixel-based input it is often useful. Haar-like features (see Subsection 1.2) can also benefit from histogram equalization although they use intensity differences between pixels. The reason is that intensity differences for images with different intensity distributions produce different results.

[2] The CCD array in this case has actually more sensors than the maximum image size is. The reason for this is cheaper mass production.

Gabor features[3] on the other hand are robust against local distortions caused by variance in illumination (Shen and Bai, 2006), so histogram equalization is not necessary when using them.

Histogram equalization is simple to implement and computationally inexpensive. The function that maps image pixel intensity to a histogram equalized value is

$$s_k = \sum_{i=0}^{k} \frac{n_i}{n}, k = 0,1,2,\ldots,L-1$$

where s_k is histogram equalized intensity value for *k*th intensity value in the range *L* of total number of possible intensity values in the original and target image, *n* is the number of pixels in the original and target image, and n_i is the number of image pixels that have intensity value *i* in the original image.

Examples of histogram equalized face images with their original counter parts are shown in Figure 1. As can be seen, face intensities look more uniform and in one case contrast has improved dramatically.

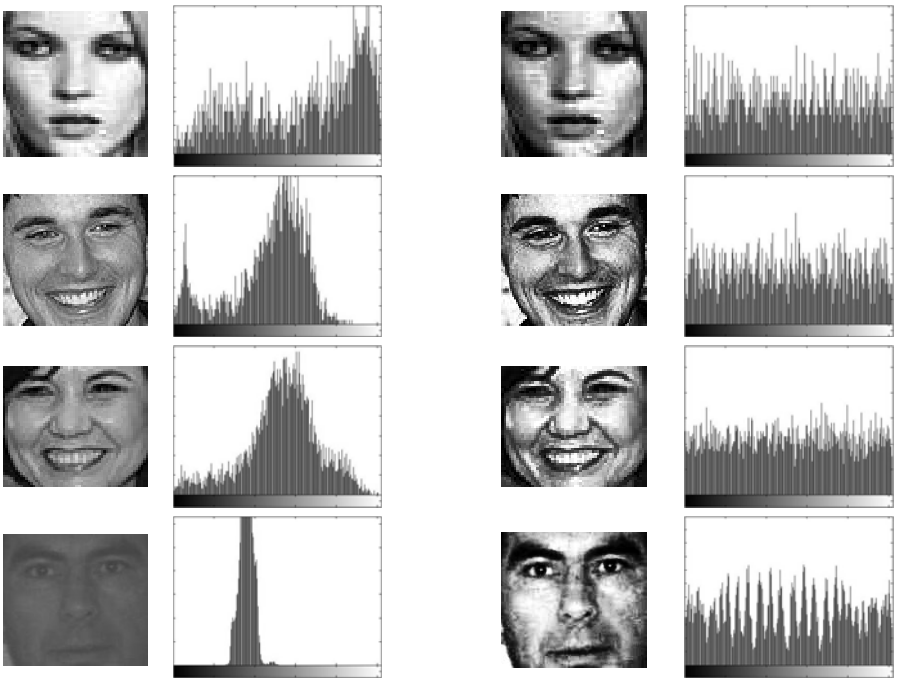

Figure 1. Original images are shown on the left and corresponding histogram equalized face images are shown at the right. The histogram of each face image is shown at the right side of the image.

[3] A comprehensive introduction to the Gabor features will be found in the thesis by Kämäräinen (2003)

Although histogram equalization usually produces good results, it is worth noting that in some rare cases it does not work well. Such example is shown in Figure 2. The face has strong shadows and after histogram equalization the right part of the face has burned out. To remove the effect, a method to remove shadows could be used (see Subsection 2.2) before doing the histogram equalization. Also histogram specification (also known as histogram matching) would work better in this case but, unlike histogram equalization, it requires manual parameter setup. From this it follows that histogram equalization is more useful in automatic face analysis systems.

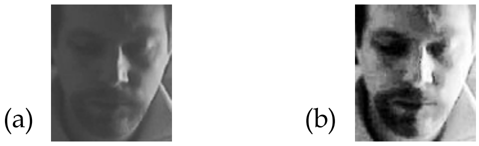

Figure 2. (a) Original image with strong shadows. (b) Image after histogram equalization. Histogram equalization does not remove shadows and the right side of the face has burned out.

Connected component labeling is a method used for finding regions from an image. It can be used in face detection to find skin colored regions from an image. After regions have been found further processing with some other method can be used to separate faces from other skin colored regions.

The algorithm for connected component labeling at the general level is shown in Figure 3.

1. Set criteria that define when two pixels are similar enough to belong to the same region.
2. Starting from the top left corner of the image, move pixel by pixel to the left and down until the bottom right corner of the image is reached.
3. For each pixel *p* check the nearest neighbor pixel above and to the left of the pixel *p*. If neither of the neighbors fulfills the defined criteria with the pixel *p*, give a new label for the pixel *p*. If only other neighbor pixel fulfills the defined criteria then give the pixel *p* the same label as the neighbor has. If both neighbors fulfill the criteria and they have same labels then give the label to the pixel *p*. If both neighbors fulfill the criteria and they have different labels then give the pixel *p* either label and add both labels to the equivalency class.
4. After the bottom right corner of the image has been reached go through the image second time. This time for each pixel *p* check if its label is in an equivalence class. If the label is in the equivalence class then common label is given to the pixel *p* with the rest of pixels which labels are in the equivalence class.

Figure 3. Algorithm for the connected component labeling.

There are many factors affecting the performance of the connected component labeling algorithm. Wu et al. (2005) studied ways to optimize it.

1.2 Machine Learning and Pattern Recognition

Duda et al. (2001, p. 1) define *pattern recognition* as follows: "the act of taking raw data and making an action based on the "category" of the pattern." The definition means that pattern recognition is a rather broad field. In the case of face analysis raw data means images (not only face images) and action can be, for example, classification determining whether there is a face or not in the image or whose face is in the image. To be successful in the action a pattern recognition system has to be constructed of appropriate parts. Usually the raw data has to be changed into another form that is used in classification. Raw data can be changed into another form using various digital image processing techniques and feature extraction methods. Machine learning techniques are used for the classification. In the case of face analysis *machine learning* means that the computer adapts to the classification problem so that it can distinguish faces from non-faces, identities of the faces, genders of the faces, and so on.

There are numerous feature extraction and machine learning techniques. For example, Gabor wavelets have a biological basis and also have many other properties that make them attractive for face analysis and computer vision in general.

The techniques presented differ a lot from each other. However, one thing common to all the machine learning techniques presented is that they learn from examples. This means that a set of positive and negative data examples, for example faces and non-faces, or female and male faces, are used in the learning. The classifier learns to discriminate classes from each other (faces from non-faces, females from males, and so on) when it is trained with the examples.

Neural networks have been used in various fields including computer vision. Multi-layer perceptrons, one type of neural networks, are possibly the most used type in computer vision problems. An example of the multi-layer perceptron is shown in Figure 4.

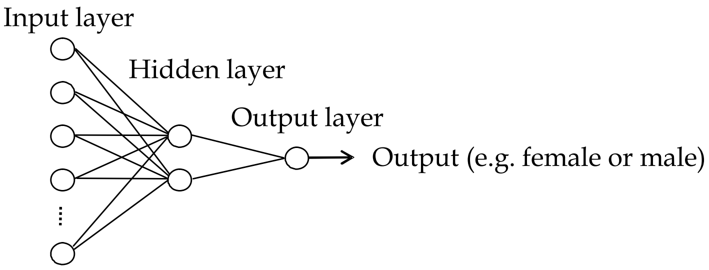

Figure 4. Example of a multi-layer perceptron with one hidden layer and one output node.

A multi-layer perceptron is a neural network with an input layer, an output layer, and with one or more hidden layers. Each layer has a set of nodes and there are connections between the nodes. Each connection is represented by a weight. The data to be classified is inputted to the network through the input layer. Each node at the input layer represents one data value. The input layer feeds the data forward to the nodes of the hidden layer. The hidden layer then feeds the input further to the other hidden layer or to the output layer. The output of the output layer is the classification result. When the data is fed through the network it is modified by the neuron activation functions. The functions modify the neuron inputs using the connection weights and then they output the modified input to the next layer.

The learning of the multi-layer perceptron is based on changing the connection weights between the nodes of the layers. A multi-layer perceptron is typically trained with the back-propagation algorithm. There is a set of the training data (for example, faces and non-faces) that is used for the training. The training proceeds so that the training data examples are fed one by one into the network and the network weights are modified based on the difference between the produced output and expected output. The name back-propagation comes from the fact that weight updating takes place by updating the output layer weights first and then proceeding layer by layer towards the input layer. The training takes place in rounds so that after all training examples have been inputted to the network the new training round starts by inputting the first example to the network.

In addition to the training images there is usually a set of validation images. They are used to avoid over-fitting to the training data. Over-fitting means that the neural network fits to the training data so well that the classification of the unseen data suffers. In practice, validation takes place so that the validation images are classified with the neural network after each training round and the classification error, the difference between the expected and real outputs is calculated. Training is stopped when validation error starts to increase, which means that the classification of the validation images begins to detoriate.

Adaboost (Freund and Schapire, 1997) is also a fairly popular method among face analysis methods. For example, the face detector by Viola and Jones (2001) had a cascade of discrete Adaboost classifiers using Haar-like features (see Figure 5).

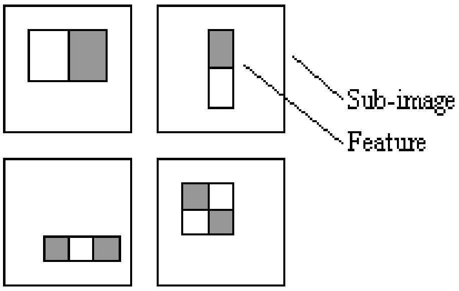

Figure 5. Haar-like features used with the cascaded face detector.

Haar-like features extract information from the raw image pixel data. Each Haar-like feature has a specific type and a specified location. Each type of the Haar-like features used with the cascaded face detector by Viola and Jones (2001) are shown in Figure 5. Each Haar-like feature has a value that is calculated as follows:

1. The Haar-like feature is placed on the specific location inside the face sub-image.

2. Pixel intensities inside the dark rectangle(s) are summed together. A sum is calculated similarly for the white rectangle(s).

3. The sum of the dark rectangle(s) is subtracted from the sum of the white rectangle(s).

The discrete Adaboost training algorithm is shown in Figure 6. The classification problem can be, for example, gender classification from faces in which case training examples are face images and classes are female and male. The training takes place in rounds. Each example face is given an equal weight at the start. This means that each face has an equal effect on feature selection. At each round each training data example is classified with each unselected weak classifier and feature. At each round a weak-classifier and the feature with the lowest classification error are selected and included in the resulting feature set. The example faces that are classified correctly with the selected feature are given lower weight for the next feature selection round so that the misclassified faces have more effect on the next selection. The number of training rounds determines the number of features that will be selected for the final classifier. The strong classifier is formed of all the selected features, and the features selected first tend to have more effect on the classification.

With the threshold weak classifiers each weak classifier has a threshold value that is determined during training. When an image is classified with the weak classifier the value calculated for the corresponding Haar-like feature is compared to the threshold. The weak classification is decided either as male or female depending on whether the calculated value is smaller or bigger than the threshold. As described above, the final classification is a result of the all weak classifications, so that the features selected earlier during training are weighted more.

Given example images (x_i, y_i); $i = 1, \ldots, n$ where $y_i = \pm 1$ for positive and negative examples respectively.

Initialize weights $w_{t,i} = 1/n$; $i = 1, \ldots, n$; $t = 1$

For $t = 1, \ldots, T$:

1. Normalize the weights so that the $\sum_{i=1}^{n} w_{t,i} = 1$.

2. For each feature j calculate the error $e_{t,j} = \sum_{i=1}^{n} w_{t,i} \left| h_j(x_i) - y_i \right|$, where h_j is the weak classifier for the feature j and it produces value -1 or 1.

3. Select the feature, j_t, with the lowest error $e_{t,j}$, and include it in the resulting feature set.

4. For each x_i classified correctly, $w_{t+1,i} = w_{t,i}\beta_t$, where $\beta_t = e_{t,j} / (1 - e_{t,j})$.

The strong classifier is $H(x) = \begin{cases} 1, if \sum_{t=1}^{T} \alpha_t h_t(x) \geq d \\ -1, otherwise \end{cases}$, where $\alpha_t = \log \frac{1}{\beta_t}$ and optimal d is near $\frac{1}{2} \sum_{t=1}^{T} \alpha_t$.

Figure 6. Training algorithm for the discrete Adaboost.

LUT Adaboost (Wu et al., 2003a) has, instead of the threshold, a lookup table (LUT) for each feature. The lookup table has a specified number of bins, for example five bins. Each bin corresponds to a certain range of possible feature values. The bin ranges are of equal size and the total range of bins includes all possible feature values. During training the feature values are calculated for each example. The counter of the bin that matches the calculated feature value is incremented by one. After training each bin contains the number of positive examples that had feature value within the range of that bin. The number of negative examples per bin is similarly stored. During classification feature value is calculated and the numbers of positive and negative examples inside the bin that correspond to the feature value are retrieved. If the number of positive examples is greater than the number of negative examples, then the classification result with the feature is positive and otherwise negative. The final classification result is calculated exactly as it is calculated with the threshold Adaboost and with the mean Adaboost. It is the weighted classification result formed of the all weak classifications.

Support Vector Machine (SVM) is a relatively new machine learning algorithm from 1992 (Boser et al., 1992). It has been successfully applied in many face analysis problems. SVM transforms the data, which may have been preprocessed, in high-dimensional feature space and classifies the data in that space to specified classes. The classes are separated from each other by maximizing the margin between the training examples of different classes in the space. The training examples that separate the classes from each other are also called support vectors. The data may be face images or, for example, hand images.

The transformation to the high-dimensional space with SVM is done using a kernel function. Linear, polynomial and radial basis function (RBF) and Gaussian kernels have been widely used in the applications. Each kernel has a different set of parameters and there are various algorithms for finding the optimal parameters for them. The RBF kernel has two parameters that determine the result of the training with the set of training examples used. These parameters are cost and gamma.

Local binary patterns (LBPs) (Ojala et al., 1996) are features the values of which are calculated from the image pixel intensities in a local pixel neighborhood. The basic idea is that as many binary values are created as there are pixels in the neighborhood of the center pixel and at the end these are concatenated to one binary value. The algorithm for calculating the value for the LBP-features is given in Figure 7.

1. For each pixel (g_p; p= 0, …, P-1; P is the size of the neighborhood) in the neighborhood of the center pixel g_c, calculate difference $x_p = g_p - g_c$.
2. The value of LBP is calculated with $LBP_{P,R} = \sum_{p=0}^{P-1} s(x_p) * 2^p$, where P is the size of the neighborhood and R is the radius from the center pixel. $s(x_p) = 1$ if $x_p \geq 0$, and otherwise $s(x_p) = 0$.

Figure 7. Algorithm for the calculation of LBP feature value.

An example of using the local binary pattern on a face image pixel neighborhood is shown in Figure 8. The example in Figure 8 is of the $LBP_{4,1}$-feature that has a radius of one pixel and 4 neighbor pixels next to the center pixel.

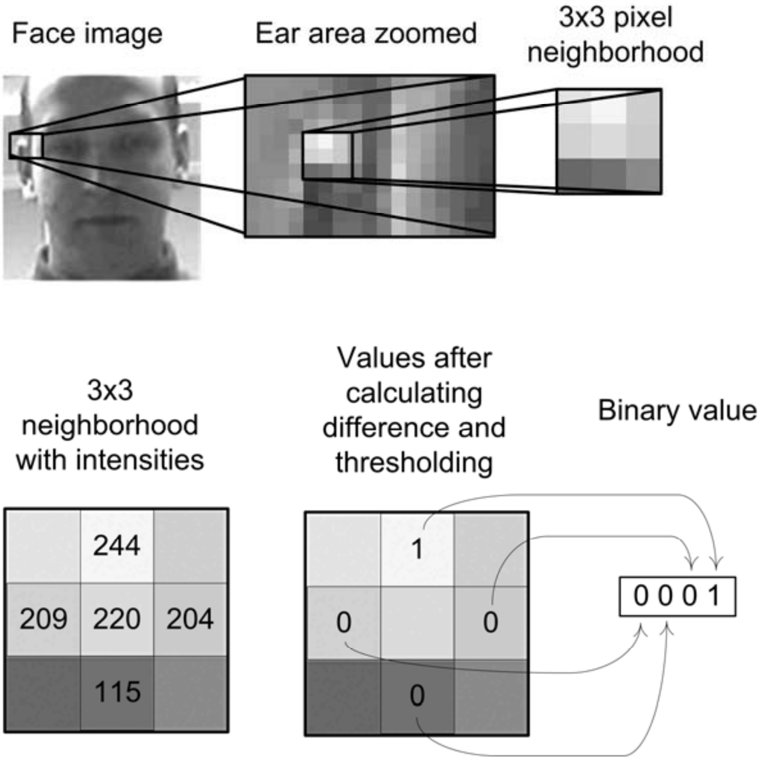
Face image
Ear area zoomed
3x3 pixel neighborhood
3x3 neighborhood with intensities
244
209
220
204
115
Values after calculating difference and thresholding
1
0
0
0
Binary value
0 0 0 1

Figure 8. An LBP$_{4,1}$-operator in use.

Originally LBP was defined for the 3*3 pixel neighborhood but later Ojala et al. (2002) extended it to different neighborhoods and introduced rotation invariant and rotation invariant uniform extensions to it. Rotation invariant uniform patterns solve the practical problem that some patterns may occur too rarely to create reliable statistics for specific analysis problem. The calculation with these rotation invariant uniform patterns proceeds first as with regular LBP. However, after a concatenated binary value has been created, it is rotated so that as many significant bits as possible are zero before the first occurrence of binary value one. For example, 01100000 would be rotated to 00000011. In addition, if the binary value has more than two bitwise transitions (from zero to one or from one to zero) then it is changed to a predetermined value that is the same for all such binary values that have more than two bitwise transitions. For example, binary values 01010000 and 10101010 would be changed to a predetermined value while the binary value 01100000 would only be rotated. The formula and a more detailed description for the basic LBP and rotation invariant uniform LBP calculation will be found in the journal article by Ojala et al. (2002).

2 Automatic Face Analysis

The face provides vast amount of information. We can tell a lot about the other person and his or her feelings just by seeing his or her face. Looking at another person's face and eyes is natural and usually expected when discussing face to face. On the other hand, communication between humans and computers still mostly happens using, keyboard, mouse, and a display. When more effective, versatile, and user friendly ways to use computers are developed automatic face analysis is one promising tool to be used.

There are many ways that automatic face analysis can be used. A person can be identified from the face and facial expressions can be analyzed so that the computer can adapt to the usage situation, for example.

Besides the applications, face analysis research is also interesting from other aspects. Because the face is a complex non-rigid object, the results of face analysis research can be useful in a wider range of object recognition tasks. For example, Yang et al. (2002) stated that most of the research on object recognition has been limited to rigid objects. They also stated that large sets of faces have been used in the experiments, which have been rare with other objects. Both these properties of face analysis research may help in other object recognition problems.

The relation of the automatic face analysis process to the other parts of the typical HCI system is shown in Figure 9 and the face analysis process is illustrated in Figure 10.

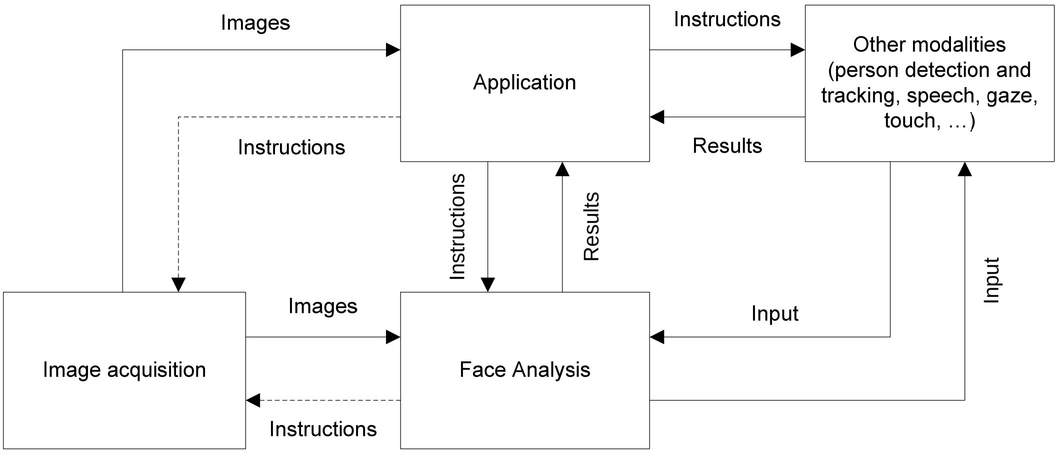

Figure 9. Face analysis related to the whole HCI system.

As shown in Figure 9 the application controls the whole system by giving instructions to the other components of the system. The other components produce results that the application utilizes. The image acquisition component provides images, for example by capturing them with a video camera and the application may choose to show them to a user if that fits the context of the application. The face analysis component analyzes images that are inputted to it by the image acquisition component and feeds the results into the application. The results may include but are not limited to the number and locations of the faces in an image at a certain moment, facial expressions, genders of the faces, and identities of the people whose faces have been detected in the image. Both the application and face analysis component may give instructions to the image acquisition component. The direction of the camera and camera parameters can be changed, for example. There may be other components such as speech and gaze tracking components that communicate with the face analysis component, so that the application receives multimodal feedback from them. For example, person identification can be based on both face and speech data, and gaze information can be combined with the facial expression data.

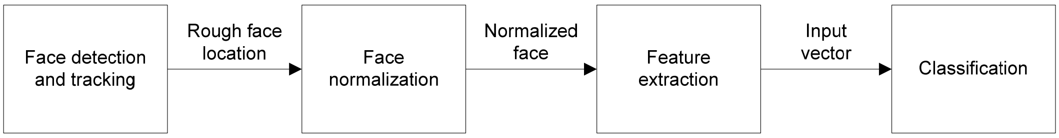

Figure 10. Face analysis in detail.

The first task in the face analysis process is to find the faces as shown in Figure 10. Depending on the application the faces may be tracked over time or detected from a single image (or from video image but without tracking). Yang et al. (2002) define *face detection* as follows: "Given an arbitrary image, the goal of face detection is to determine whether or not there are any faces in the image and, if present, return the image location and extent of each face." *Face tracking* means estimating the location of the face continuously from video image (Yang et al., 2002). In addition to face detection and tracking facial features may be detected and tracked. *Facial feature detection* means finding the locations of the features such as eyes, nose and mouth from a face.

If person detection is used in addition to face detection then person detection might be done first or it might happen at the same time with face detection but this is not a necessity. In addition, in some specific applications face images are inputted to the system and faces need not to be detected or tracked.

The results of the face detection and tracking phase may be useful as such for the application. However, usually there is need for further analysis of the faces found. Typically some kind of normalization is done before features are extracted for the classification phase. The most typical normalizations are alignment and normalization of the illumination. Various methods can be used for both of these. After normalization the face data is transformed to the form that can be used in the classification phase. This transformation is often called feature extraction. It may be very simple such as using image pixels directly as an input vector for a classifier, or it may be a rather complicated process such as filtering the face with Gabor wavelets.

The classification phase may include several tasks. For example, the gender, identity, age, facial expressions or ethnicity of the person can be determined. One classification task may precede the other or they can be fully separate. The benefit of preceding one task with another is that the reliability of the latter classification may be enhanced (Saatci and Town, 2006). If gender, age or ethnicity classification precedes face recognition then the recognition speed may also be increased because the preceding classification reduces the set of candidate faces for face recognition.

The actual implementation of the face analysis component is affected by the application where face analysis is used. Many of the HCI applications require continuous tracking of the faces from real-time video image. This means that computationally expensive methods cannot be used unless enough efficient hardware is available. Even though the faces were tracked continuously it may be enough to do a classification of the face gender, age, or identity from only one video frame. However, the reliability can be improved if several video frames are used for the classification (Castrillón-Santana et al., 2006; Wu et al., 2003b).

Face analysis from a video image has both benefits and disadvantages. In addition to the increased number of face images on which the classification is based, the segmentation of the moving face (or person) from the video image using motion information can be used. On the other hand, occlusions and other challenges such as varying illumination, low image resolution and changing face pose may increase the difficulty of face analysis.

2.1 FACE DETECTION AND TRACKING

As stated above, face detection means finding all the faces from an image, and face tracking means that the faces are detected over time from video image and each face is matched between the images. One should note that there could be more than one face in the image. If we can assume that the image contains only one face then the task is simpler and the term to use is *face localization*.

Face detection is a challenging task since there are many conditions that may vary. Each person has a unique face, meaning that each face looks different. Even the face of the same person looks different depending on the time when the image is taken. For example, the age of the person, eyeglasses, beard, moustache and make-up make a difference. Pose and the orientation of the faces in relation to the camera vary. Facial expressions affect the look of the face. Face may be partially occluded by some other object, also by another face. Even the imaging conditions vary. The image may be taken outdoors in daylight, indoors in fluorescent light, or in other lighting conditions. An image may be gray scale or color image and resolution can vary. Image may be taken with a web camera, with a high quality frame grabber or with a 3D scanner. The effects of various conditions are shown in Figure 11.

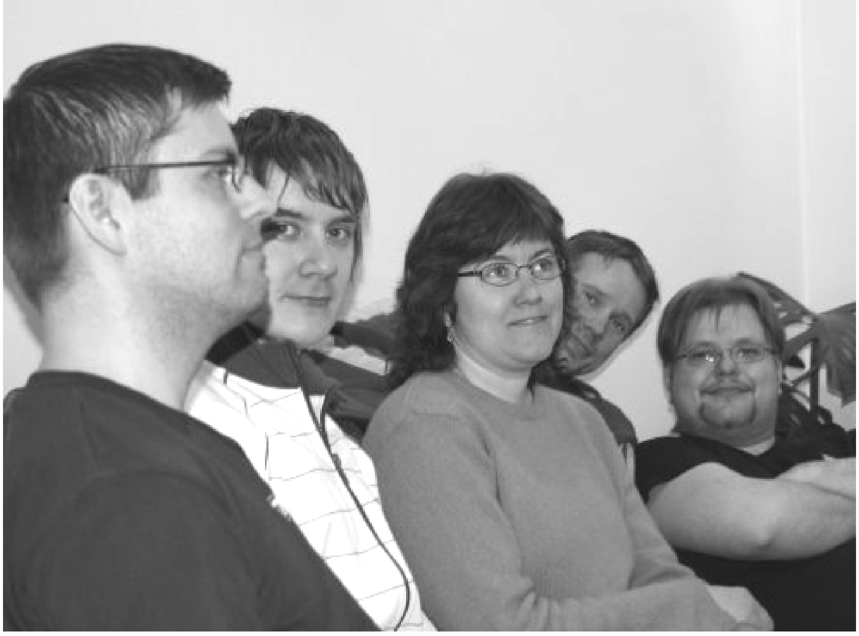

Figure 11. Examples of possible causes of problems in face detection and tracking. Faces with various orientations and poses, some occluding the others.

Face detection is the first step in a fully automatic face analysis system and thus it is one reason why it has received a lot of attention from researchers. For example, it is needed in fully automatic face recognition applications.

The evaluation of the face detection methods is important (Yang et al., 2002) as it is with the other face analysis tasks. There are public databases that can be used to evaluate face detection methods. Different metrics have been defined for the evaluation of face detectors, for example: detection accuracy, detection speed, required training time, the number of training samples required for the training, and the memory requirements during training and use.

The detection rate and false alarm rate determine the detection accuracy. The *detection rate* can be defined as the ratio between the number of correctly detected faces and the number of faces in the image. For example, if there are 10 faces in the image and 8 faces are correctly detected by the face detector then the detection rate is 80%. The *false alarm* rate on the other hand determines the number of detected faces that actually are not faces. It is possible to report it as a ratio between the number of false detections and the number of face locations searched for from the image. However, the ratio depends on the face locations searched for and the number of locations searched varies between methods. Usually, the interesting fact, especially from the viewpoint of the application that uses face detection, is the actual number of false detections. Therefore, when reporting experimental results the number of false detections should be used. In addition to the detection rate and false alarm rate there are two related terms: false positive and false negative. *False positive* means that a non-face object has been detected as a face and *false negative* means a face that has not been detected.

It is also good practice to report ROC (Receiver Operating Characteristics) curves for the methods. A ROC curve for a hypothetical face detector is shown in Figure 12. The y-axis defines the detection rate and the x-axis the number of false alarms. The points in the curve determine detection rate with a certain number of false alarms. The bigger the area under the ROC curve is the better, because the closer the curve is to the upper left corner the higher the detection rate is while number of the false alarms is not increased. The perfect curve would be such that it goes from the lower left corner to the upper left corner and from there to the upper right corner.

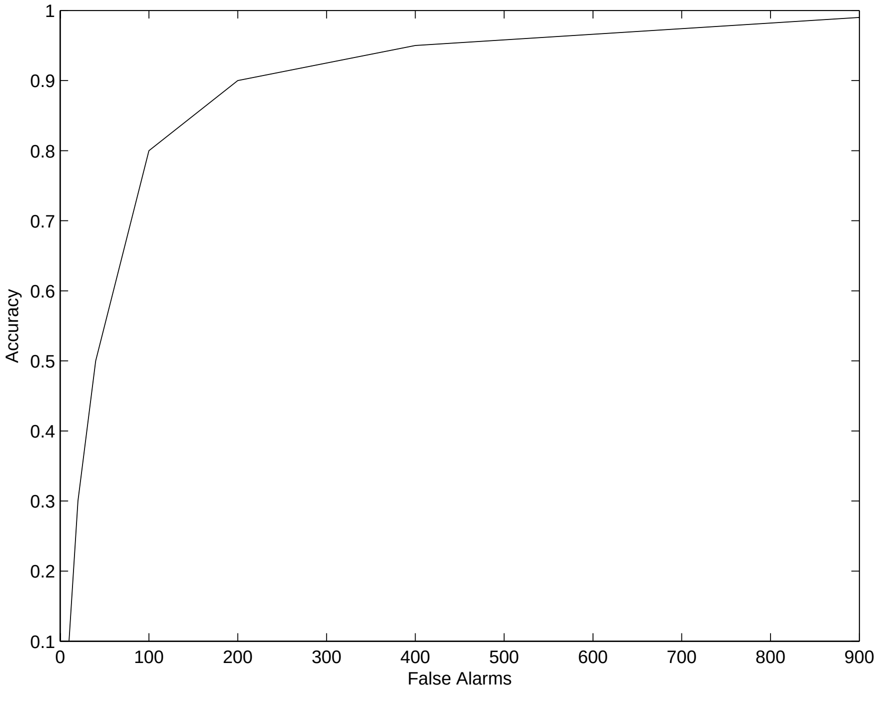

Figure 12. A ROC curve for a hypothetical face detector.

A high detection rate and only few or no false detections are often preferable, but some applications may have other criteria. Yang et al. (2002) mention face validation application. In that case the detected face is used for identity verification. The detected face is matched to the face of the person in the database that the person being verified claims to be. The false detections will be rejected by the face verifier because they hardly match the face in the database. Therefore, false detections are not that big a problem in this case.

The detection speed is usually an important factor. There are great differences in detection speeds between detection methods. There are also some other factors than the method which determine the detection speed. For example, the image size affects the speed. Naturally, the larger the image is the longer time is required to detect faces from it. Hardware is another important affecting factor. The methods that work in "real-time" with a standard PC are the most useful in typical HCI applications since users usually expect immediate feedback on their actions. Such face detection methods are, for example, the cascaded face detector by Viola and Jones (2001) and the rotation invariant multi-view face detector by Huang et al. (2007). Naturally, as hardware is constantly becoming more powerful even such methods that have been computationally too expensive to use become usable at some point.

The number of face samples needed in training is important when considering the memory requirements for the system where training takes place. Memory may also be a problem when the trained detector is in use. The required training time and work needed to train the detector may be an issue, for example, when selecting a detection method for a commercial product or even when doing academic research. A long training time may also be a problem with an application that uses real-time detection and on-line training would be needed (Yang et al., 2002).

One thing to consider is how precisely the face has to be located that the detection is considered correct. If the application is interested only of the number of faces in the image or just the rough location is needed, then all the detections for the face can be considered correct. However, if further classification is done for the detected face then badly located faces may become a problem even if face alignment is used. As good data as possible should be provided to the classifiers and alignment does not necessarily work if initial face location is very inaccurate.

Two extensive surveys have been published on face detection (Hjelmås and Low, 2001; Yang et al., 2002). These surveys also handled facial feature detection. Face detection methods can be categorized in several ways. Yang et al. (2002) divided face detection approaches into four categories: knowledge-based, feature invariant, template matching and appearance-based. Hjelmås and Low (2001) had two main categories: feature-based approaches and image-based approaches. These two categories were further subdivided into different categories.

Knowledge-based approaches use rules that are defined by a human. Feature invariant methods are based on the idea that facial features are detected from the image before the face. After the features have been detected they are grouped together to form a face. The facial features can be low-level or high-level. Examples of low-level features are points, edges, color, and intensity. High-level features are for example eyes, nose, and mouth. In the template matching methods some sort of template is compared to image regions and the regions that correlate well with the template are considered as a face. In appearance-based methods a face model is learned by the face detector so that it is trained with a large set of face and often non-face examples. The detector can use a neural network or several neural networks, SVM, Adaboost or some other machine learning method.

The cascaded face detector by Viola and Jones (2001) is a very well known appearance-based method. As its name suggests, the detector is constructed of a cascade of classifier layers. Each layer has a set of simple classifiers trained with the Adaboost algorithm. The detector searches faces from an image by starting from the top left corner of the image and ending to the bottom right corner of the image (see Figure 13). The image where faces are detected is searched through several times with a different sub-image size each time. However, the image does not have to be resized when different size faces are searched for from the image. Instead, Haar-like features (see Figure 5) and integral images that are used with the detector make it possible to resize the features instead and still use the same number of calculations with all feature sizes.

When the image is scanned through, the sub-images are passed to the first layer of the face detector cascade that contains a set of Haar-like features to determine whether the sub-image contains a face or not (see Figure 14). If the first layer classifies the sub-image as a face then the sub-image is passed to the next layer. This is continued until the sub-image is passed through all layers or discarded in a layer. If it is passed successfully through all layers then the final classification is a face and otherwise a non-face.

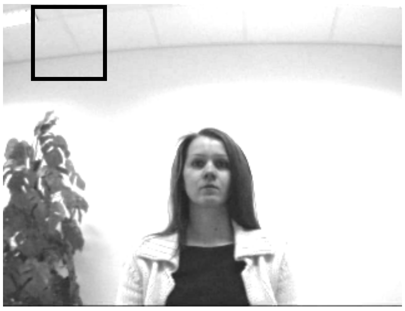

Figure 13. Each image is scanned from top left corner to bottom right corner using sub-images.

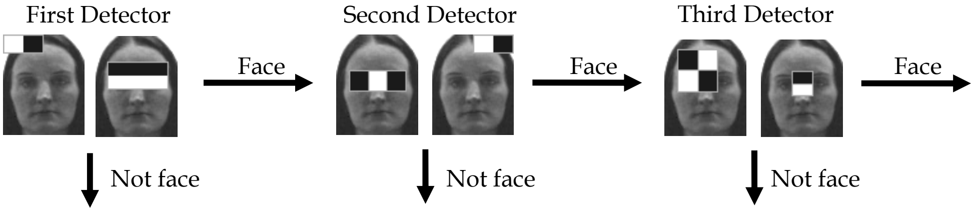

Figure 14. Face detector cascade. Two features are shown for each layer.

Each layer is an Adaboost classifier that makes the detection decision based on the set of Haar-like features on that layer (see Figure 14 and Figure 5). The main idea of the cascade is that easily recognizable non-face sub-images can be rejected at the earlier layers which have fewer features than the later layers. The original face detector by Viola and Jones (2001) had 32 layers and the first layer had 2 features while the last 20 layers had 200 features. This cascaded structure brings efficiency because it takes less time to process a sub-image in the earlier layers and most of the sub-images are rejected at the first layers. The detector can reliably detect frontal faces in gray scale images with over 90% detection rate at the frequency of 15 frames per second with a typical PC.

Face tracking means following a face or faces for a certain period of time. Many of the algorithms that can be used for object and person tracking can also be used for face tracking. Before a face can be tracked it needs to be detected at least once. However, after the initial detection, detection and tracking can also take place simultaneously. For example, the system by Yang et al. (2006) had separate face detection and tracking modules. The system switched between the modules when needed to improve robustness of tracking.

Facial feature detection means the detection of such high-level features as eyes, nose, chin, mouth, mouth corners, and lips or low-level features such as facial edges or points, for instance.

Facial feature detection is often an integral part of face detection. Those knowledge-based face detection methods that use rules based on facial feature locations require that facial features are detected before the face location is determined. However, facial feature detection can also be separate from face detection.

There are variations in facial feature appearances between people, which cause challenges for facial feature detection and tracking. In addition, especially mouth appearance changes a lot when a person speaks and shows expressions. Eye appearance is also changed within expressions but also when we open and close the eyelids.

2.2 FACE NORMALIZATION AND ALIGNMENT

After or even before the face and possibly facial features have been detected some kind face normalization is usually needed. Most systems include some sort of image intensity normalization. The face images may have been captured in different lighting conditions and with different camera parameters and normalization improves the robustness of the system. Histogram equalization described in Subsection 1.1 can be used for minimizing the differences between the imaging conditions. In addition to histogram equalization there are other methods that can be used for intensity normalization. For example, the cascaded face detector by Viola and Jones (2001) used variance normalization to create unit variance for the sub-window intensities during image scanning. To make the normalization faster it was done for the feature values instead of the pixels and the variance was calculated using the integral images. In addition to histogram equalization Sung and Poggio (1998) used illumination gradient correction to reduce the effect of heavy shadows. The illumination gradient correction was done so that the intensity plane that best fitted the face image was subtracted from the image. Choi et al. (2007) presented a method for shadow compensation that was specifically designed for faces. It used knowledge of how shadows appear on faces when the direction of the light changes. Other methods have also been proposed for shadow compensation.

The location of the face can also be normalized so that the face sub-images that are used as input to the following face classification tasks are more consistent in location and face shape. Most of the face detection methods find the rough location of the face and face alignment can be useful in these cases. One possibility is to use located facial features. For example, faces can be rotated so that the eyes are vertically aligned. Furthermore, faces can be resized and the place of the sub-images translated so that eyes are at the same location in all sub-images. The shape of the face can also be modified, for example, by stretching or squeezing the face so that, in addition to the eyes, the mouth is also at the same location in the sub-images. Naturally, for the alignment to be useful, it has to be exact. Otherwise it just makes the face analysis slower while the classification rates remain unchanged or even get worse. For example, Active Appearance Models (AAM) (Cootes and Taylor, 2001) can be utilized for face alignment.

2.3 FACE RECOGNITION AND VERIFICATION

In addition to face detection the other face analysis topic that has received a lot of attention from researchers is face recognition. Zhao et al. (2003) defined *face recognition* in their survey as follows: "given still or video images of a scene, identify or verify one or more persons in the scene using a stored database of faces."

Although even commercial systems on face recognition exist, the face recognition field still faces many challenges. Zhao et al. (2003) mentioned that major challenges with face recognition are "illumination, pose and recognition in outdoor imagery."

The history of computer based face recognition goes back to the 1970s (Kelly, 1970; Kanade, 1977). Since then numerous studies have been carried out and several face recognition literature surveys have been written (Samal and Iyengar, 1992; Chellappa et al., 1995; Zhao et al., 2003; Scheenstra et al., 2005; Tan et al., 2006).

The reason why face recognition has been under intense research in recent years is partly in the amount of commercial applications. One application area for face recognition is security and surveillance and some commercial systems already exist. The other reasons for popularity are that hardware has become efficient enough and the importance of security related applications has increased (Zhao et al., 2003). The complexity of the face as an object to be analyzed makes face analysis, including face recognition, interesting from the research point of view, as stated above.

In the face recognition field, there have been efforts to achieve comparable results of different methods. Probably the best known public database for face recognition is the FERET database (Phillips et al., 1998). It contains 14,051 face images of 1,199 individuals. Depending on the person there are images with different facial expressions and poses. In addition, there are images where lighting conditions have been changed or images were taken at different dates. The FERET database has later been updated with a color version containing 11,338 images of 994 individuals and is largely same as the original FERET database but for color format.

There have also been several Face Recognition Vendor Tests (FRVTs) in years 2000, 2002, and 2006 (Blackburn et al., 2001; Phillips et al., 2003; Phillips et al., 2007) carried out by the US government organization National Institute of Standards and Technology (NIST) that evaluated commercial and prototype face recognition technologies. As a part of the latest evaluation, the US government provided Biometric Experimentation Environment (BEE) that made it easier for an experimenter to evaluate the methods and for researchers to prepare their methods for the evaluation.

As another example, The CSU Face Identification Evaluation System (CSU, 2003) provides implementations of some well known algorithms (PCA, PCA combined with LDA, Bayes, and Elastic Bunch Graph Matching with Gabor jets) and protocols to carry out face recognition experiments.

Although there are challenges in face recognition the technology is constantly improving. The face recognition accuracy in the last FRVT 2006 evaluation was ten times better than in the FRVT 2002 evaluation four years earlier (Phillips et al., 2007).

2.4 GENDER CLASSIFICATION

The neurophysiological and psychological research on gender classification started before the computer vision research on the topic. The computer vision research on the topic started at the beginning of the 1990's. The research has been partly motivated by psychology but it has applications in other fields including HCI.

The very first computer vision studies on gender classification were reported simultaneously by Cottrell and Metcalfe (1990) and by Golomb et al. (1990). The studies were quite similar. Auto-associative networks and perceptrons were used in both and the aim was to have a working gender classifier. In addition, Cottrell and Metcalfe (1990) also considered face and facial expression classification. After that various methods have been proposed for the task. Recently, Baluja and Rowley (2007) and Mäkinen and Raisamo (2008) have compared performances of gender classifiers in various conditions.

2.5 FACIAL EXPRESSION AND GESTURE CLASSIFICATION

Emotions greatly affect human behavior and it has been shown that emotions affect our memory and other mental processes (Lewis and Critchley, 2003). For example, humans cannot make decisions if their emotional functions have been damaged (Damasio, 1994). Emotions and facial expressions also have an extremely important role in communication between humans. Facial expressions mirror our emotions, mental activities, and physiological activities (Fasel and Luetin, 2003), and help other people to understand us.

When humans interact with computers emotions are also present. When there are some problems with the application we are using we may become annoyed and eventually even angry. There are even extreme cases where users have broken their computers after getting angry. Naturally, when humans are communicating with each other through computers emotions and facial expressions are also involved.

There have been efforts to define facial expressions precisely. Ekman and Friesen (1978) developed the Facial Action Coding System (FACS) that has become the most well known system for describing facial expressions. It defines 46 facial action units (AUs) that are based on facial muscle movements. All facial expressions can be defined using this system.

Ekman and Friesen (1971) defined six basic emotions and their corresponding facial expressions that seem to be universal among all cultures in the world. These emotions are: happiness, sadness, fear, disgust, surprise, and anger. However, in addition to these basic expressions there are a lot of expressions that do not belong to the basic ones (Tian et al., 2001) and there are differences among cultures in showing and interpreting facial expressions (Fasel and Luetin, 2003; Matsumoto, 1993). Facial expressions can also be caused on purpose, in which case they can be considered gestures.

Pantic and Rothkrantz (2000) and Fasel and Luetin (2003) have reported surveys on facial expression analysis. In addition to the general challenges in face analysis including lighting conditions, occlusions and so on, there are challenges specific to automatic facial expression classification. In the case of facial expressions we are measuring intra-personal variations in face. However, there are still differences in facial expressions and their intensities between different individuals. Because there are about 7,000 combinations of AUs (Ekman, 1982) it is also hard to develop a system that can take all these combinations into account. Since facial expressions are dynamic they have temporal characteristics. Each expression has onset (attack), apex (sustain), and offset (relaxation). These can be analyzed from an image sequence but not from a single image.

Fasel and Luettin (2003) pointed out that the automatic facial expression classification field has still a lot of research to do. Most of the existing systems classify expressions directly to the six basic emotions (Fasel and Luettin, 2003). However, in practice expressions are almost always more complex and many expressions are caused by physiological activities or are used as gestures. It is also possible to classify expressions using the AUs. In this case the classified AUs are interpreted using a facial expression dictionary. One example of such a dictionary is the Facial Action Coding System Affect Interpretation Dictionary (FACSAID) by Ekman et al. (1998). Only a few existing systems use these dictionaries in the classification.

Most of the systems assume frontal faces and allow only small head movements between the video frames, and manual initialization is often needed. In addition, there are only a few systems (Lien, 1998; Lisetti and Rumelhart, 1998; Fasel and Luettin, 2000) that classify intensities of facial expressions.

2.6 Age Classification

The specific challenges with age classification are that the age of the person is hard to predict exactly because facial appearance changes slowly when a person is aging and this change in appearance is somewhat person dependent. Further, bone structure and wrinkles are also affected by other factors than age alone. For example, identity and ethnicity affect the bone structure and wrinkles that appear within aging can also be caused by facial expressions.

Kwon and Lobo (1999) used facial feature detection and wrinkle detection to classify age to the three age groups: babies, young adults and seniors. They carried out experiments with the faces of 5 babies, 5 young adults, and 5 seniors. Using the locations between detected facial features and the number of wrinkles on the face they determined the age group of the face. Classification was successful for all 15 faces. Ueki et al. (2006) presented a classifier based on two phases using 2D-LDA and LDA to classify age. The benefit of their classifier is that it is robust under various lighting conditions. Ueki et al. (2006) experimented by using age ranges of 5 years, 10 years, and 15 years. The respective classification rates for each range were 46.3%, 67.8%, and 78.1%. Besides classification to age groups it is also possible to estimate the exact age of the person. The studies by Lanitis (2002) and by Lanitis et al. (2004) achieved roughly a 5-year mean error in the experiments where they used face images of people aged between 0 and 35 years.

There is also a database specifically intended for research on face based age classification: FG-NET Aging database (FG-NET, 2007).

3 Conclusions

Computer vision is a challenging research field that has many application areas. This tutorial considered computer vision from face analysis viewpoint. Some state of the art methods, algorithms and studies were presented that serve as a good starting point for further research on the face analysis field.

References

Baluja, S. & Rowley, H. A. (2007), Boosting sex identification performance, *International Journal of Computer Vision* **71**(1), 111-119.

Blackburn, D., Bone, J. & Phillips, P. (2001), FRVT 2000 evaluation report, *Technical Report*, National Institute of Justice, USA.

Boser, B. E., Guyon, I. M. & Vapnik, V. N. (1992), A training algorithm for optimal margin classifiers, *in Proceedings of the 5th Annual Workshop on Computational Learning Theory* (COLT'92), ACM Press, New York, NY, USA, pp. 144-152.

Castrillón-Santana, M., Déniz-Suárez, O., Lorenzo-Navarro, J. & Hernández-Tejera, M. (2006), Gender and identity classification for a naive and evolving system, *in 2nd Workshop on Multimodal User Authentication* (MMUA'06).

Chellappa, R., Wilson, C. & Sirohey, S. (1995), Human and machine recognition of faces: A survey, *Proceedings of the IEEE* **83**(5), 705-741.

Choi, S., Kim, C. & Choi, C. (2007), Shadow compensation in 2D images for face recognition, *Pattern Recognition* **40**(7), 2118-2125.

Cootes, T. & Taylor, C. (2001), Statistical models of appearance for medical image analysis and computer vision.

Cottrell, G. W. & Metcalfe, J. (1990), EMPATH: Face, emotion, and gender recognition using holons, *in* Richard Lippmann, John E. Moody & David S. Touretzky, ed., *Proceedings of the Advances in Neural Information Processing Systems* 3 (NIPS), Morgan Kaufmann, pp. 564-571.

CSU (2003), The CSU face identification evaluation system.

Damasio, A. (1994), *Descartes' Error*, Grosset/Putnam.

Duda, R. O., Hart, P. E. & Stork, D. G. (2001), *Pattern Classification*, A Wiley-Interscience Publication.

Ekman, P. (1982), *Handbook of Methods in Nonverbal Behaviour Research*, Cambridge University, chapter: Methods for measuring facial actions, pp. 45-90.

Ekman, P. & Friesen, W. V. (1978), *Facial Action Coding System: A Technique for the Measurement of Facial Movement*, Consulting Psychologists Press.

Ekman, P. & Friesen, W. V. (1971), Constants across cultures in the face and emotion, *Journal of Personality and Social Psychology* **17**(2), 124-129.

Ekman, P., Rosenberg, E. & Hager, J. (1998), Facial action coding system affect interpretation dictionary (FACSAID), http://face-and-emotion.com/dataface/facsaid/description.jsp.

Fasel, B. & Luettin, J. (2003), Automatic facial expression analysis: A survey, *Pattern Recognition* **36**(1), 259-275.

Fasel, B. & Luettin, J. (2000), Recognition of asymmetric facial action unit activities and intensities, *in Proceedings of the 15th International Conference on Pattern Recognition* (ICPR’00), pp. 1100-1103.

FG-NET, FG-NET Aging database, 2007.

Freund, Y. & Schapire, R. E. (1997), A decision-theoretic generalization of on-line learning and an application to boosting, *Journal of Computer and System Sciences* **55**(1), 119-139.

Golomb, B. A., Lawrence, D. T. & Sejnowski, T. J. (1990), SEXNET: A neural network identifies sex from human faces, *in Proceedings of the Advances in Neural Information Processing Systems* 3 (NIPS), Morgan Kaufmann, pp. 572-579.

Gonzalez, R. C. & Woods, R. E. (2002), *Digital Image Processing*, Prentice Hall.

Hjelmås, E. & Low, B. K. (2001), Face detection: A survey, *Computer Vision and Image Understanding* **83**(3), 236-274.

Huang, C., Ai, H., Li, Y. & Lao, S. (2007), High-performance rotation invariant multiview face detection, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **29**(4), 671-686.

Kanade, T. (1977), Computer recognition of human faces, *Interdisciplinary Systems Research* **47**.

Kelly, M. D. (1970), Visual identification of people by computer, *PhD Thesis*, Stanford University.

Kwon, Y. H. & da Vitoria Lobo, N. (1999), Age classification from facial images, *Computer Vision Image Understanding* **74**(1), 1-21.

Kämäräinen, J. (2003), Feature extraction using Gabor filters, *PhD Thesis*, Lappeenranta University of Technology.

Lanitis, A. (2002), On the significance of different facial parts for automatic age estimation, *in Proceedings of the 14th International Conference on Digital Signal Processing* (DSP'02), pp. 1027-1030.

Lanitis, A., Draganova, C. & Christodoulou, C. (2004), Comparing different classifiers for automatic age estimation, *IEEE Transactions on Systems, Man and Cybernetics* **34**(1), 621-628.

Lewis, P. A. & Critchley, H. D. (2003), Mood-dependent memory, *Trends in Cognitive Sciences* **7**(10), 431-433.

Lien, J. J. (1998), Automatic recognition of facial expressions using hidden Markov models and estimation of expression intensity, *PhD Thesis*, Robotics Institute, Carnegie Mellon University.

Lisetti, C. & Rumelhart, D. (1998), Facial expression recognition using a neural network, *in Proceedings of the 11th International Flairs Conference*.

Matsumoto, D. (1993), Ethnic differences in affect intensity, emotion judgments, display rule attitudes, and self-reported emotional expression in an American sample, *Motivation and Emotion* **17**(2), 107-123.

Mäkinen, E. & Raisamo, R. (2008), Evaluation of gender classification methods with automatically detected and aligned faces, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **30**(3), 541-547.

Ojala, T., Pietikäinen, M. & Harwood, D. (1996), A comparative study of texture measures with classification based on featured distributions, *Pattern Recognition* **29**(1), 51-59.

Ojala, T., Pietikäinen, M. & Mäenpää, T. (2002), Multiresolution gray-scale and rotation invariant texture classification with local binary patterns, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **24**(7), 971-987.

Pantic, M. & Rothkrantz, L. J. (2000), Automatic analysis of facial expressions: The state of the art, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **22**(12), 1424-1445.

Phillips, P., Grother, P., Micheals, R., Blackburn, D., Tabassi, E. & Bone, J. (2003), FRVT 2002 evaluation report (NISTIR 6965), *Technical Report*, National Institute of Standards and Technology, USA.

Phillips, P. J., Scruggs2, W. T., O'Toole, A. J., Flynn, P. J., Bowyer, K. W., Schott, C. L. & Sharpe, M. (2007), FRVT 2006 and ICE 2006 large-scale results (NISTIR 7408), *Technical Report*, National Institute of Standards and Technology, Gaithersburg, USA, MD 20899.

Phillips, P. J., Wechsler, H., Huang, J. & Rauss, P. J. (1998), The FERET database and evaluation procedure for face recognition algorithms, *Image and Vision Computing* **16**(5), 295-306.

Rowley, H. A., Baluja, S. & Kanade, T. (1998), Neural network-based face detection, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **20**(1), 23-38.

Saatci, Y. & Town, C. (2006), Cascaded classification of gender and facial expression using active appearance models, *in Proceedings of the 7th*

IEEE International Conference on Automatic Face and Gesture Recognition (FG'06), pp. 393-400.

Samal, A. & Iyengar, P. A. (1992), Automatic recognition and analysis of human faces and facial expressions: A survey, *Pattern Recognition* **25**(1), 65-77.

Scheenstra, A., Ruifrok, A. & Veltkamp, R. (2005), A survey of 3D face recognition methods, *in Proceedings of the 5th International Conference on Audio- and Video-Based Biometric Person Authentication* (AVBPA'05), pp. 891-899.

Shen, L. & Bai, L. (2006), A review on Gabor wavelets for face recognition, *Pattern Analysis and Applications* **9**(2), 273-292.

Sung, K. & Poggio, T. (1998), Example-based learning for view-based human face detection, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **20**(1), 39-51.

Tan, X., Chen, S., Zhou, Z. & Zhang, F. (2006), Face recognition from a single image per person: A survey, *Pattern Recognition* **39**(9), 1725-1745.

Tian, Y., Kanade, T. & Cohn, J. (2001), Recognizing action units for facial expression analysis, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **23**(2), 97-115.

Ueki, K., Hayashida, T. & Kobayashi, T. (2006), Subspace-based age-group classification using facial images under various lighting conditions, *in Proceedings of the 7th IEEE International Conference on Automatic Face and Gesture Recognition* (FG'06).

Viola, P. & Jones, M. (2001), Rapid object detection using a boosted cascade of simple features, *in Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (CVPR'01), pp. 511-518.

Wu, B., Ai, H. & Huang, C. (2003a), LUT-based Adaboost for gender classification, *in Proceedings of International Conference on Audio and Video-Based Biometric Person Authentication* (AVBPA'03), pp. 104-110.

Wu, B., Ai, H. & Huang, C. (2003b), Real-time gender classification, *in Proceedings of the 3rd International Symposium on Multispectral Image Processing and Pattern Recognition*, pp. 498-503.

Wu, K., Otoo, E. & Shoshani, A. (2005), Optimizing connected component labelling algorithms, *in Proceedings of the SPIE, Medical Imaging 2005: Image Processing*, pp. 1965-1976.

Yang, M., Kriegman, D. & Ahuja, N. (2002), Detecting faces in images: A survey, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **24**(1), 34-58.

Yang, T., Li, S., Pan, Q., Li, J. & Zhao, C. (2006), Reliable and fast tracking of faces under varying pose, *in Proceedings of the 7th IEEE International Conference on Automatic Face and Gesture Recognition* (FG'06), pp. 421-426.

Zhao, W., Chellappa, R., Phillips, P. & Rosenfeld, A. (2003), Face recognition: A literature survey, *ACM Computing Surveys* **35**(4), 399-458.