Face Detection Using Multiple Cues

Thomas B. Moeslund, Jess S. Petersen, and Lasse D. Skalski

Laboratory of Computer Vision and Media Technology
Aalborg University, Denmark

Abstract. Many potential applications exist where a fast and robust detection of human faces is required. Different cues can be used for this purpose. Since each cue has its own pros and cons we, in this paper, suggest to combine several complimentary cues in order to gain more robustness in face detection. Concretely, we apply skin-color, shape, and texture to build a robust detector. We define the face detection problem in a state-space spanned by position, scale, and rotation. The state-space is searched using a Particle Filter where 80% of the particles are predicted from the past frame, 10% are chosen randomly and 10% are from a texture-based detector. The likelihood of each selected particle is evaluated using the skin-color and shape cues. We evaluate the different cues separately as well as in combination. An improvement in both detection rates and false positives is obtained when combining them.

1 Introduction

The "Looking at people" research field covers applications where cameras observe humans. This ranges from surveillance, HCI, to motion capture and analysis of athletes' performance. All applications require the humans to be segmented in the image and a tremendous amount of research has been conducted in this field due to the potential applications [11]. For some applications only one or a few body parts are required. For example the human face for applications such as identity recognition, facial expression recognition, head pose estimation or simply to detect the presence of a person. The core technology for such applications is first of all to detect human faces in an image or video sequence. Different methods for doing this have been suggested and they can roughly be grouped according to the type of data they operate on.

Many operate by finding skin pixels in the image and group then into head-shaped objects [6,10,11]. Such methods are sensitive to other skin-color objects present in the scene. A different approach is to look for head-shaped objects in either a silhouette or edge version of the input image [7,9,10,11,18]. But again, background clutter can disrupt this data type. Yet another approach is to find features inside the face, e.g., eyes, nose, mouth, hair-line, cheeks, etc., and build a classifier based on their structural relationships [10,14,17]. Such approaches are indeed affected by background clutter and therefore work best as a verification tool for possible head candidates [10]. Furthermore they tend to operate best on frontal images and can be quite heavy computational-wise. A related approach

B.K. Ersbøll and K.S. Pedersen (Eds.): SCIA 2007, LNCS 4522, pp. 51–60, 2007.
© Springer-Verlag Berlin Heidelberg 2007

is to apply the appearance of the entire face, for example using the texture of the face [10,16,17].

No matter which cues one uses background clutter and other noise sources will challenge the detector. A combination of multiple cues can therefore be applied to make a detector more robust. Often this is done by using either a skin-color detector followed by a verification using facial features [6,10] or using an appearance-based detector followed by a verification using facial features [10,14].

In this work we do a parallel fusion of multiple cues in other to benefit directly the complimentary characteristics of the different cues. This is similar to approaches followed in other domains. For example in [13] where persons are detected using color and texture, and in [15] where hands are detected using color and motion, and in [1] where arms are detected using color, motion and shape. Concretely we combine color, shape and texture to build a robust and fast face detector. Furthermore, since we are interested in detecting faces in video, we apply the temporal context to improve the detections. In section 2, 3, and 4, we describe the shape, color and texture cues, respectively. In section 5 we describe how they are integrated and how the temporal context is applied. In section 6 results are presented and in section 7 a conclusion is given.

2 Shape-Based Detection

Shape-based detection is based on the fact that the contour of the head is a rather distinct feature in a standard image and the face can hence be found by finding this shape.

Shape-based detection (and recognition) is based on as least two key elements: 1) a definition and representation of the shape and 2) a measure for the similarity between the shape model projected into the current image and the edges found in the current image.

2.1 Shape Model

A correct model of human heads and their variations can be trained and modeled using for example a Point Distribution Model. However, for the purpose of detecting the human face a rough model will suffice. A rough and very simple model is an ellipse, which in many cases is a rather accurate match, see figure 1. The elliptic shape is not necessarily unique, i.e., other objects in a scene might have this shape. It has therefore been suggested to enhance the uniqueness by including the shoulder-profile [18], see figure 1. While the contour of a head seldom deform this is not the case for the shoulder and the head-shoulder profile is therefore often modeled using some kind of dynamic contour represented by a Spline. Due to the extra parameters such methods require extra processing and can be sensitive to arm/shoulder movements. We therefore use an elliptic model.

We note that the neck can be hard to identify resulting in a poor match for the lower part of the ellipse. We therefore only apply the elliptic arc seen in figure 1.

Fig. 1. An illustration of matching different shape types with the human head in the image

2.2 Shape Matching

Matching is here based on edges extracted from the images. When matching a shape to an image, it is desired that a smooth search space is present. This ensures that not only a perfect match results in a high similarity measure but also solutions in the proximity. This is vital since a perfect match is virtually impossible due to noise and an imperfect shape model. We apply Chamfer matching [2] to generate a smooth search space from an edge image.

Chamfer matching can be used to find occurrences of a shape in an image based on edges extracted from the image. The matching is based on a distance map, which is created from the edge image using a distance transformation. This distance map is an image, in which each pixel contains the distance to the nearest edge. The matching is done by projecting the shape into the distance map, and sum the values of the overlapping pixels in the distance map. The sum is normalized by the number of pixels resulting in the average distance, $\overline{d}$, for a particular shape x. This average distance is converted into a likelihood measure as

$$P_{Shape}(x) = \begin{cases} 0, & \overline{d} \geq 10; \\ 1 - \frac{\alpha \cdot \overline{d}}{10}, & \text{Otherwise.} \end{cases} \quad (1)$$

where α is a constant learned during training.

3 Color-Based Detection

Detecting a face based on color relays on the notion that skin-color is a rather distinct feature in a standard image. Skin color is in fact a strong cue for finding

faces, but obviously flawed by the fact that other skin regions, e.g., hands and arms, are often present. As for the shape cue, this cue also requires the choice of an appropriate representation and matching scheme.

We have assessed different color representations (spaces) and matching schemes and found the one suggested by [8] to be the best in terms of sensitivity. Besides, it operates in the RGB color space, which means that there is no computational overhead in transforming the colors from the input image (R,G,B). Furthermore, the method is chosen, because it is able to detect skin color in indoor scenes with changing illumination. The matching scheme is shown below. The likelihood measure for the color cue is implemented as an AND operation

Table 1. Skin classification rules from [8]

Lighting conditions	Uniform daylight	Flashlight or lateral daylight
Skin color classification rule	R > 95, G > 40, B > 20 Max{R,G,B}-Min{R,G,B} > 15 |R-G| > 15, R > G, R > B	R > 220, G > 210, B > 170 |R-G| ≤ 15, B < R, B < G

between the thresholded and filtered skin-color image and an ellipse representing a candidate face, see figure 2. When representing a face using an ellipse, not all face pixels will be classified as skin color, due to eyes, hair, and mouth. The likelihood measure is therefore calculated by counting the number of skin pixels within the ellipse and dividing by β of the total number of pixels within the ellipse.

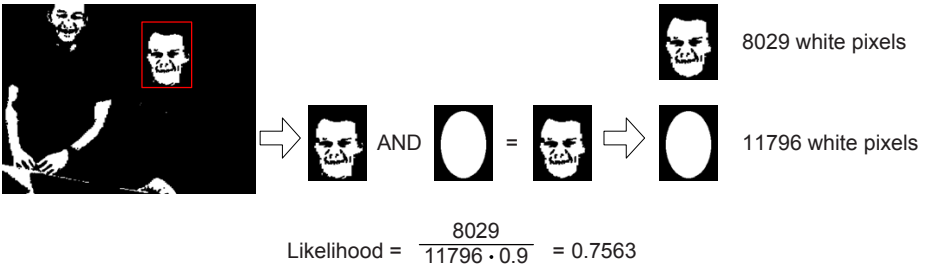

Fig. 2. An example of how the skin color likelihood is calculated for $\beta = 0.9$

4 Texture-Based Detection

A texture-based detector looks for templates having face-like appearance. In its most simple form template matching is applied. However, in recent years a different strategy has been very successful, namely to use a number of simple

and generic templates as opposed to merely one specific template. The basic idea was first proposed in [16], where two key ideas are presented: 1) create a face detector based on a combination of weak classifiers and combine them to a boosted classifier using machine learning, and 2) create a cascade of boosted classifiers resulting in the final face detector.

A weak classifier is constructed of a single Haar-like feature, a boosted classifier is a weighted combination of weak classifiers, and a cascaded classifier is a sequence of boosted classifiers as illustrated in figure 3.

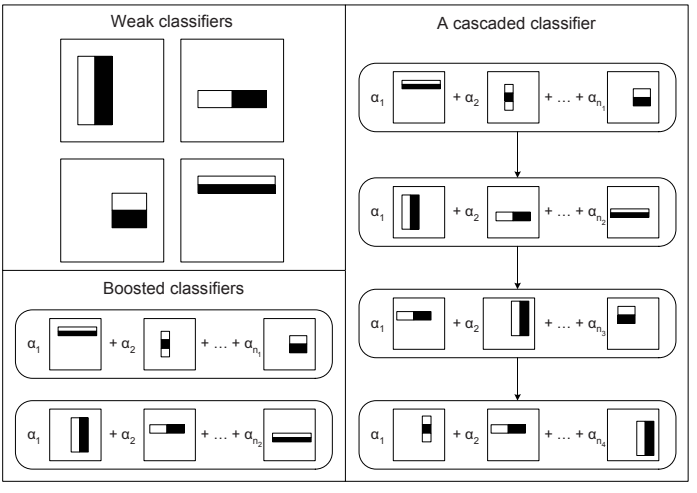

Fig. 3. Examples of weak classifiers, boosted classifiers and a cascaded classifier

To detect faces in images a subwindow is moved across the image in different scales. Each subwindow is then processed by the cascaded classifier. Each boosted classifier in the cascade is denoted a layer, for each of these the subwindow is evaluated by the corresponding boosted classifier. If the subwindow is classified as a face, it is passed to the next layer. A subwindow must be classified as a face by all layers of the cascade to be detected. A cascaded classifier is trained to consists of increasingly more complex boosted classifiers. Each boosted classifier has a very high detection rate and a moderate false acceptance rate, e.g. 99% detection rate and 50% false acceptance rate. This enables the first few layers of the cascaded classifier to reject a majority of the non-face subwindows in the input image. In this way more computation time is spent on more difficult samples.

The detector can process an image relatively fast due to the simple nature of the features and the clever invention of an integral image [16]. However, training the detector takes a very long time due to the massive amount of possible features, positions and scales and a large training set of normally several 1000s positive and negatives samples.

We apply a subwindow size of $24x24$ pixels resulting in around 160.000 possible features. We use the AdaBoost algorithm [16] to do the training using around

5000 face images and 7000 non-face images (which took approximately one month of constant processing!) resulting in a detector consisting of 11 boosted classifiers with a total of 587 weak classifiers.

The output of the detector is a number of subwindows likely to contain a face. Overlapping subwindows are merged into only one output.

5 Combining the Cues

The different detectors can each analyze a particular position, scale, and rotation in the image. In order to represent all possible solutions we define a state-space spanned by the different degrees-of-freedom. These are the two translations in the x- and y-direction, rotation in the image plane and scale. The first two have a resolution of one pixel and are limited to the image plane. The rotation parameter has a resolution of 15° degrees and is limited to ±30° [12]. The scale parameter is linked to the two primary axes of the ellipse in the following manner.

The size of an ellipse is defined by the major and minor axes. In order to limit the number of parameters we use the 2006 anthropometric data from NASA [12] to find the ratio between these axes for average humans: Male: 25,7cm(Height) / 16,5cm(Width) = 1,558. Female: 24,3cm(Height) / 16,8cm(Width) = 1,446. Based on this we define a general average ratio of 1.5 and use that to scale the ellipse. The resolution of the scale-factor is 5 and it is limited by height $\in [48; 144]$ pixels, corresponding to $1m - 3m$ from the camera.

The state-space is spanned by four axes and each point in the space corresponds to one particular position, rotation and scale of the head. A detector can now operate by trying each possible state and see how well it matches the current image using the detectors described above. If a state has a high match then a face is located. Due to the size and resolution of the state-space such a brute force approach is not realistic due to the heavy processing.

For most applications where face detection is required the movement of people between frames will typically be limited, hence, temporal knowledge can be used to reduce the state-space. A well-documented framework for this is a Particle Filter [11].

When a particle filter is used to reduce the state-space, the space is limited to the states, denoted particles, with a high likelihood in the previous frame. This allows for approximating the entire state-space using several magnitudes of fewer possible solutions. A low number of particles may however cause problems if new persons enter the scene or if a person is not detected in all frames. We therefore only sample 80% of the particles from the state-space in the previous frame and as suggested in [4] we randomly sample 10% of the particles to cover random events. The last 10% are sampled from the output of the texture-based detector. This means that the likelihood function used to evaluate each particle will only be based on color and shape. We have tried different combinations and found this to be the most suitable solution since the texture-based detector is the best stand-alone detector and tends to produce many false positives making it suitable to detect new objects and lost objects. Furthermore, the texture-based

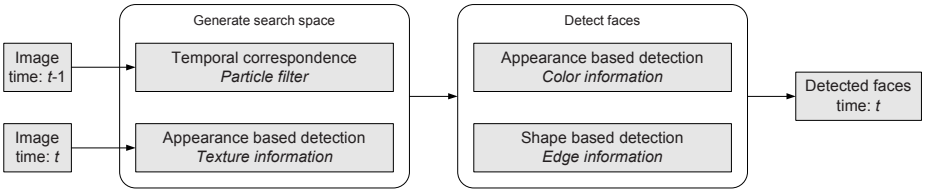

Fig. 4. An illustration of the cue integration

detector only estimates three of the four state-space parameters. In figure 4 the cue integration is illustrated.

To summarize, the particle filter has two main steps for each frame. First a search space is generated, i.e., likely particles are defined. 80% are predicted from the previous frame using Random Walk. 10% are chosen randomly, and the last 10% are taking from the output of the texture-based detector and diffused to create slight variations. This results in N particles likely to contain a correct state of a face in the current image. Each of these N states are now evaluated regarding color and shape using both the color-based detector and the shape-based detector. The output will be a state-space where each entry contains a likelihood of this particular state being present in the current image. By finding the maximum peaks the faces are detected. Since the particle filter is defined in a Bayesian framework the peaks are equal to the MAP (maximum a posteriori).

6 Results

We define a correct detection as a situation where minimum 50% of the face is inside the ellipse defined by a state and minimum 50% of the pixels inside the ellipse are skin-pixels from the face. A face is defined as the visible region of the head with hair, but without the neck. We use 250 manually annotated frames from a complicated scene containing background clutter, non-human skin color objects and motion in the background. The algorithm is evaluated using 1000 particles corresponding to 0.2% of the total number of possible solutions in the state-space. The framerate is 5.1Hz on a Pentium 1300MHz Centrino.

Five different evaluations are performed:

A: Texture-based detection
B: Particle filter using color detection, and 10% randomly sampled particles in each image
C: Particle filter using shape detection, and 10% randomly sampled particles in each image
D: Particle filter using color and shape detection, and 10% randomly sampled particles in each image
E: Particle filter using color and shape detection, and 10% randomly sampled particles in each image, and 10% samples distributed using the texture-based detection

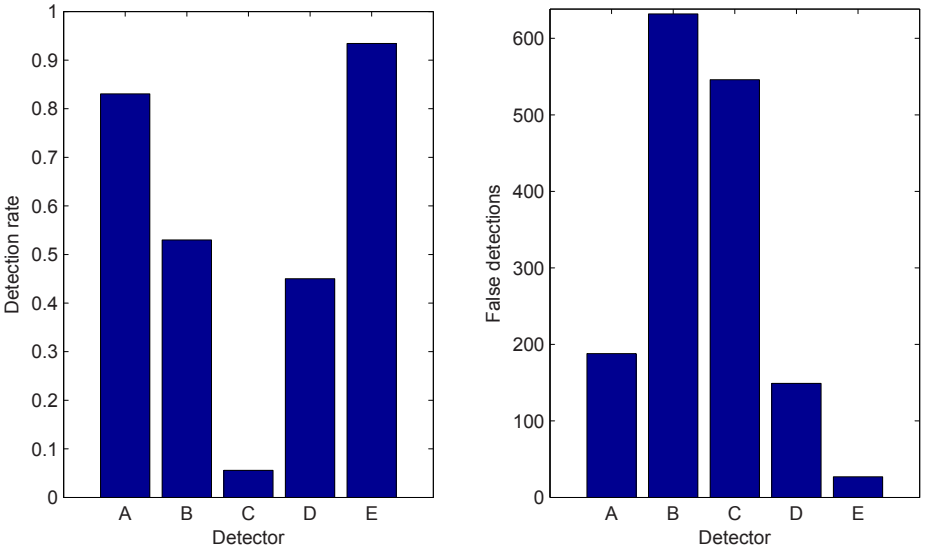

Fig. 5. The results of evaluating the five detectors

The particle filter is initialized by random sampling all particles. However, in evaluation **E** the particle filter is initialized by randomly sampling 50% of the particles and distributing the remaining 50% based on the detections from the texture-based detection. In figure 5 the detection rates and the number of false detections for each of the detectors are shown.

The results show that detector **E** has the highest detection rate and has significantly fewer false detections than the other detectors, i.e., using multiple cues outperforms any of the individual detectors. Detector **A** has significantly higher detection rate than **B**, **C**, and **D**, which underline the current trend in computer vision of using machine learning based on massive training data. The detection rate of detector **B** is slightly higher than the detection rate of detector **D** which is unexpected. However, the number of false detections using detector **D** is remarkably lower than using detector **B** meaning that detector **C** is too sensitive as a detector (primarily due to background clutter), but can contribute to eliminating false detections as it compliment detector **B**. Comparing detector **D** and **E** shows the significance of introducing particles selected by the texture-based detector **A**. Detector **A** does introduce more false detections, but using the color and shape cues allow us to prune non-supported states resulting in relative few false detections. Figure 6 shows example images containing for detector **E**.

7 Discussion

The tests clearly show the benefit of combining several complimentary cues. Most of the errors of detector **E** are very close to the actual face, see figure 6, meaning that the detection rate can be increased and the false detections lowered

Fig. 6. Example images containing both true and false detections using detector **E**

if the definition of detection in section 6 is relaxed. Another possible improvement is to increase the number of particles, but experiments show that significantly more particles are required resulting in a somewhat slower system. Other alternatives are either some kind of postprocessing, e.g., a Mean Shift tracker [3] or to make each particle converge to a local maximum see e.g., [5]. But again, such improvements will reduce the speed of the system.

References

1. Azoz, Y., Devi, L., Yeasin, M., Sharma, R.: Tracking the Human Arm Using Constraint Fusion and Multiple-Cue Localization. Machine Vision and Applications 13(5-6), 286–302 (2003)
2. Borgefors, G.: Hierarchical Chamfer Matching: A Parametric Edge Matching Algorithm. IEEE Transaction on Pattern Analysis and Machine Intelligence 10(6), 849–865 (1988)
3. Cheng, Y.: Mean Shift, Mode Seeking, and Clustering. IEEE Transactions on Pattern Analysis and Machine Intelligence 17(8), 790–799 (1995)
4. Davis, L., Philomin, V., Duraiswami, R.: Tracking Humans from a Moving Platform. In: International Conference on Pattern Recognition, Barcelona, Spain (September 3-8, 2000)
5. Deutscher, J., Reid, I.: Articulated Body Motion Capture by Stochastic Search. International Journal of Computer Vision 61(2), 185–205 (2005)
6. Hsu, R.L., Abdel-Mottaleb, M., Jain, A.K.: Face Detection in Color Images. IEEE Transactions on Pattern Analysis and Machine Intelligence 24(5), 696–706 (2002)
7. Hu, W., Hu, M., Zhou, X., Tan, T., Lou, J., Maybank, S.: Principal Axis-Based Correspondence between Multiple Cameras for People Tracking. IEEE Transactions on Pattern Analysis and Machine Intelligence 28(4), 663–671 (2006)
8. Kovac, J., Peer, P., Solina, F.: 2D Versus 3D Colour Space Face Detection. In: EURASIP Conference on Video / Image Processing and Multimedia Communications EC-VIP-MC'03, Zagreb, Croatia (July 2-5, 2003)

9. Lanitis, A., Taylor, C.J., Cootes, T.F.: Automatic Interpretation and Coding of Face Images Using Flexible Models. IEEE Transactions on Pattern Analysis and Machine Intelligence 19(7), 743–756 (1997)
10. Kriegman, D.J., Yang, M.H., Ahuja, N.: Detecting Faces in Images: A Survey. IEEE Transactions on Pattern Analysis and Machine Intelligence 24(1), 34–58 (2002)
11. Moeslund, T.B., Hilton, A., Kruger, V.: A Survey of Advances in Vision-Based Human Motion Capture and Analysis. Computer Vision and Image Understanding 104(2-3), 90–126 (2006)
12. NASA - National Aeronautics and Space Administration. Anthropometry and Biomechanics,`http://msis.jsc.nasa.gov/sections/section03.htm` (2006)
13. Okuma, K., Taleghani, A., de Freitas, N., Little, J., Lowe, D.: A Boosted Farticle Filter: Multitarget Detection and Tracking. In: European Conference on Computer Vision, Prague, Czech Republic (May 11-14, 2004)
14. Santana, M.C., Suárez, O.D., Artal, C.G., González, J.I.: Cue Combination for Robust Real-Time Multiple Face Detection at Different Resolutions. In: International Conference on Computer Aided Systems Theory, Las Palmas de Gran Canaria, Spain (February 7-11, 2005)
15. Stenger, B.: Template-Based Hand Pose Recognition Using Multiple Cues. In: Asian Conference on Computer Vision, Hyderabad, India (January 13-16, 2006)
16. Viola, P., Jones, M.: Rapid Object Detection using a Boosted Cascade of Simple Features. In: Computer Vision and Pattern Recognition, Kauai Marriott, Hawaii (December 9-14, 2001)
17. Wu, J., Pedersen, J.M., Putthividhya, P., Norgaard, D., Trivedi, M.M.: A Two-level Pose Estimation Framework Using Majority Voting of Gabor Wavelets and Bunch Graph Analysis. In: ICPR workshop on Visual Observation of Deictic Gestures, Cambridge, UK (August 22, 2004)
18. Zhao, T., Nevatia, R.: Stochastic Human Segmentation from a Static Camera. In: MOTION '02: Proceedings of the Workshop on Motion and Video Computing, Orlando, Florida, USA (December 5-6, 2002)