Ego-Motion Compensated Face Detection on a Mobile Device

Björn Scheuermann Arne Ehlers Hamon Riazy Florian Baumann Bodo Rosenhahn
Institut für Informationsverarbeitung
Leibniz Universität Hannover, Germany
secondname@tnt.uni-hannover.de

Abstract

In this paper we propose face tracking on a mobile device by integrating an inertial measurement unit into a boosting based face detection framework. Since boosting based methods are highly rotational variant, we use gyroscope data to compensate for the camera orientation by virtual compensation of the camera ego-motion. The proposed fusion of inertial sensors and face detection has been tested on Apple's iPhone 4. The tests reveal that the proposed fusion provides significant better results with only minor computational overhead compared to the reference face detection algorithm.

1. Introduction

In the last decade, much work on face detection has been done [15] and a variety of different approaches have been developed. Yang et al. [14] classify face detection methods into four categories: knowledge-based methods, feature invariant approaches, template matching methods and appearance-based methods. They characterize appearance-based methods as algorithms that learn models from a set of training images. Zhang et al. [15] recapitulate that in general appearance-based methods have shown superior performance to algorithms belonging to the remaining three categories. One of the most famous approaches belonging into that category is the object detection framework presented by Viola and Jones [12] that utilizes the machine learning algorithm Adaboost [5] in the classifier training. They boost weak classifiers based on so called Haar-like features and use an intermediate image representation, the integral image, for efficient feature computation. The drawback of many boosting based face detectors, like the framework presented by Viola and Jones, is that they are highly rotational variant.

In practice, however, it is possible that the faces are not upright in the image and boosting based methods, trained on upright faces, fail to detect the face. This situation occurs if either the people rotate their faces while taking an image or, more often, if the camera is not upright. In this paper we focus only on the situation, where the camera is not upright. We will use inertial sensors of the device to virtually align the image and run Viola and Jones detector on the rotated image. The rotation will make sure, that the face will become upright in the rotated image and thus the algorithm is able to detect the face.

(a) (b)

Figure 1: Comparison between standard face detection using the OpenCV implementation of Viola and Jones (a) and the proposed integration of inertial sensors to compensate the camera ego-motion (b).

1.1. Prior Work

In recent years several approaches have been proposed to achieve a rotation invariant detection based on the framework of Viola and Jones. In [4], Du et al. introduce a new set of $\pm$ 26.565° rotated Haar-like features that can be efficiently computed. Utilizing the new features they divide the 360° plane into 12 orientations on which faces are searched for. Although the rotated features can be calculated in an efficient way this method has the drawback that in principle 12-times as much features have to be evaluated, which overcharges the processing power of most mobile devices.

In [13], Wu et al. proposed a fast rotation invariant face detection, also based on a variant of Adaboost and Haar-like

features. They divide human faces according to the variant appearance due to different view points. Because of the running time of 250 ms for a 320×240 image on a Pentium 4 2.4 GHz PC, their rotation invariant multi-view face detection is also not applicable for a mobile device.

Ren et al. [11] proposed several optimizations for the Viola and Jones face detection algorithm on mobile devices. The optimizations are categorized into three classes: data reduction, search reduction and numerical reduction. Reducing the amount of data e.g. by spacial subsampling lead to a smaller detection rate. Also the proposed search reduction will lead to a smaller detection rate. Since our mobile device, Apples iPhone 4, has a full floating point unit the numerical reduction gained no improvement in running time.

Inertial sensors are getting major attention over the last years in the computer vision community. The main reason is, that it is a comparable cheap sensor which is readily available on mobile devices and it can act complementary to the visual cues provided from image data: In [9], Pons et al. used inertial sensors to stabilize markerless human motion capturing. Also in the problem of reconstructing 3D structure and camera motion from a sequence of images, known as structure from motion (SfM), additional inertial sensors can provide valuable information. In [6] Labrie et al. combined a camera with an inertial sensor to estimate the camera translations between the frames of a sequence. By this means they improved the efficiency of the required process of matching between the images. Clipp et al. [3] make use of consumer inertial sensors and GPS system to build a mobile reconstruction system for large scale urban scenes. In [10], Ramachandran et al. simplified the SfM problem by utilizing the measurements of inertial sensors.

1.2. Contribution

We propose the fusion of inertial sensors and the boosting based face detection by Viola and Jones. To overcome the effect that a face is not detected if the camera rotates, we utilize inertial sensors to compute a virtual upright image. Using the virtual upright image, instead of the original image, for face detection lead to a huge improvement for the situation, that the camera is rotated. Since there is no benchmark available for the fusion of inertial sensors and face detection, we will make our test sequences including the sensor information publicly available. To summarize, our contribution are:

- We ported the OpenCV [2] implementation of the Viola and Jones algorithm to Apple's iPhone 4.
- By utilizing inertial sensors we compensate camera ego-motion.
- The computational overhead is insignificant.

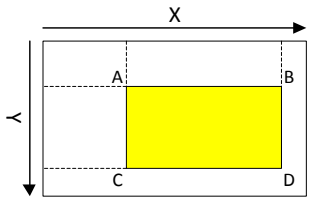

Figure 2: Rectangle in an integral image. The sum of the pixels in the rectangle edged by the points A,B,C and D can be calculated as $D + A - B - C$

- We will provide the video sequences and sensor data for the scientific community.

2. Boosting based face detection

Adaptive Boosting, Adaboost, is a machine learning algorithm proposed by Freund and Schapire [5]. During a round-based training phase it automatically composes a strong classifier as a linear combination of some weak classifiers. In each round one weak classifier is selected from a given set of classifier, that classifies a labeled training set with minimal error. That classifier is added to the linear combination and thereby weighted based on its classification error. The key point of Adaptive Boosting is that also weights are assigned to the elements of the training set and the classification error is calculated from the weights of the wrongly classified elements. After each training round these weights are adapted such that incorrectly classified elements obtain higher influence on the classification error. In that way Adaboost concentrates on the challenging elements of the training set during the classifier selection.

In [12] Viola and Jones proposed the application of Adaboost in object detection. They employed a set of Haar-like features as the basis for the classifier selection and introduced integral images to allow for their efficient calculation. Haar-like features describe rectangular shapes in the way that the difference between the sums of pixel intensities in rectangular image areas is calculated. In the integral image representation each pixel is calculated as the sum of the pixels above and to the left, including itself. Therefore the sums of rectangular regions in Haar-like features can be computed from integral images by evaluating only the corner pixels of the rectangles, see Fig. 2. Obviously this property is only valid for rectangles whose edges are parallel to the image axes. Hence in the framework proposed by Viola and Jones all Haar-like features are aligned in that way.

The training phase of the object detection framework is computational expensive as the selection of the weak classifiers is performed by an exhaustive search. This process often takes several hours or days to complete. But the strong

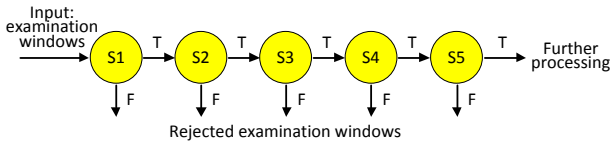

Figure 3: Attentional cascade consisting of 5 stages. The classifier in each stage assigns the examination windows to True or False class and thus decides if they are passed to the next stage.

classifier formed by the Adaboost algorithm can be applied in real-time because it is, although complex, a linear combination of classifiers based on Haar-like features. Additionally Viola and Jones set up their detector in an attentional cascade as shown in Fig. 3. In the cascade structure earlier stages contain small and efficient boosted classifiers, which are trained to reject a significant amount of negative sub-windows while preserving almost all positive examples. The following stages consist of increasingly complex classifiers that achieve lower false positive rates. As most of the examined sub-windows are rejected in the early stages only few are passed to the slower complex classifiers. Therefore the attentional cascade of classifiers can be processed very efficiently.

During training the strong classifier has been adapted to the task of classifying the training set of positive and negative example images. To enable Adaboost the creation of a classifier capable to describe a high variety of unseen objects the positive training set is usually designed on the one hand to contain enough variation. On the other hand the positive set should be constrained to reduce the complexity of the classification problem. For this reason many face detectors, like in OpenCV [2], have been trained solely on upright faces yielding a highly rotational variant classifier.

Because of the alignment constraints on the Haar-like features these face classifiers cannot be rotated without strong decreases in performance and thus rotated faces are not detected.

3. Inertial Measurement Unit

The mobile devices used in our work, Apples iPhone 4 and the 4G iPod Touch, provide an inertial measurement unit (IMU) consisting of a three axis accelerometer and a three axis gyroscope [1]. To feature a full nine degree-of-freedom (9-DoF) inertial sensing device, Apple combined both MEMS (Micro-Electro-Mechanical Systems) with an electronic compass. In this work only the 3-DoF gyroscope-data is used to determine the relevant orientation of the camera.

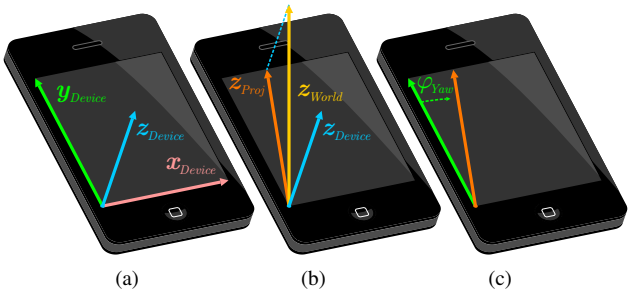

Figure 4: (a) Camera-coordinate system. (b) Projection of z_{world} onto the screen surface spanned by the vectors $(x_{\text{device}}, y_{\text{device}})$. (c) To compute the virtual upright image the yaw-angle describes the rotation angle.

3.1. World-coordinate System

The attitude, or spatial orientation of the device is measured in relation to the so called world-coordinate system. This reference system is established automatically by the device when an application starts. The world-coordinate system defined by the IMU is spanned by the axes $(x_{\text{world}}, y_{\text{world}}, z_{\text{world}})$, where

$$z_{\text{world}} = -\frac{a_{\text{gravitation}}}{||a_{\text{gravitation}}||}, \tag{1}$$

meaning that the z-axis is always the negative direction of gravity measured by the internal accelerometer. The x-axis and y-axis are always defined orthogonal to gravity to define right-hand space. To provide a constant assignment during a session, the x-axis and y-axis are internally defined according to the device orientation. Apple provides a proprietary algorithm that can accurately calculate absolute orientations relative to the static world coordinate system.

3.2. Camera-coordinate system

Fig. 4a shows the camera-coordinate system defined by the basis-vectors $(x_{\text{device}}, y_{\text{device}}, z_{\text{device}})$. In contrast to the world-coordinate system, the camera-coordinate system is well defined. The z-axis is defined to be orthogonal to the devices screen surface, y-axis is parallel to the longitudinal side and the x-axis to the transverse side. Analogue to the world-coordinate system the basis defines a right-hand space. For our problem we are especially interested in the orientation of the defined camera-coordinate system in relation to z_{world}. Therefore we need to compute the angle between a projection of z_{world} onto the device screen surface and y_{device}.

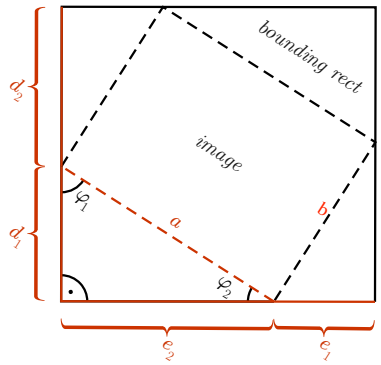

Figure 5: Bounding Rectangle. Having a yaw-angle φ and rotating the image increases its dimensions.

4. Mapping the Two Coordinate Systems

Apples proprietary algorithm provides a stream of quaternions [8] (or rotation-matrices) that define, at every frame, the map or coordinate transformation from the initial world-coordinate system to the device's current reference frame [1]. Using quaternions, we are able to compute the camera-coordinate system. Having a quaternion $q = (q_{re}, q_{im})$ describing the absolute orientation of frame t relative to the initial reference frame we get:

$$(0, i_{\text{device},t}) = q(0, i_{\text{world}})q^{-1}\,, \tag{2}$$

for $i = (x_{\text{device},t}, y_{\text{device},t}, z_{\text{device},t}) \in \mathbb{R}^3$. $(0, i_{\text{world}})$ describes the reference world-coordinate system and $(0, i_{\text{device},t})$ describes the current camera-coordinate system at frame t. Now we want to compute a mapping to align y_{device} with the negative direction of gravity z_{world}. For this, we use the projection $z_{\text{proj},t}$ of the axis z_{world} on the devices screen surface at frame t.

$$z_{\text{proj},t} = z_{\text{world}} - \langle z_{\text{world}} | z_{\text{device},t} \rangle \cdot z_{\text{device},t}\,, \tag{3}$$

as illustrated in Fig. 4b.

If $z_{\text{proj}} = y_{\text{device}}$ holds, the devices y-axis is perfect aligned with the negative direction of gravity and we can use the current frame directly to detect a face using the Viola and Jones face detector. On the other hand, if $z_{\text{proj}} \neq y_{\text{device}}$ the device y-axis is not aligned with the negative direction of gravity and we compute the so called yaw-angle according to (see Fig. 4c):

$$\cos(\varphi_{\text{yaw}}) = \pm \langle y_{\text{device},t} | z_{\text{proj},t} \rangle / ||z_{\text{proj},t}||\,. \tag{4}$$

We choose the positive sign, if $\langle y_{\text{device}} \times z_{\text{world}} | z_{\text{device}} \rangle > 0$ and else the negative sign. This gives us the angle $\varphi_{\text{yaw}} = \measuredangle y_{\text{device}}, z_{\text{proj}}$, that we need for transforming the current image and align it with z_{world}.

4.1. Rotating the Image

Having the yaw-angle, we rotate the image and produce an virtual upright image for the Viola and Jones face detector. The first step is to compute the so called bounding rectangle, since we do not want to loose information. The bounding rectangle is defined according to Fig. 5 and the equations:

$$\begin{aligned} d &= d_1 + d_2 = a\cos\varphi_{\text{yaw}} + b\sin\varphi_{\text{yaw}} \\ e &= e_1 + e_2 = a\sin\varphi_{\text{yaw}} + b\cos\varphi_{\text{yaw}} \end{aligned} \tag{5}$$

This implies that the aligned frame dimension gets bigger according to φ_{yaw}. Since the additional pixels are defined to be white, we use OpenCV's option Canny pruning [2] to skip these region in the face detection algorithm.

5. Experiments

For face detection using inertial sensors to compute a virtual upright image, there is no standard test set available. Therefore we provide detection results on 3 video sequences recorded with an iPhone 4. For all sequences we logged the angle of rotation φ_{yaw} according to Equation 4. All sequences were recorded in a public environment with the constraint, that a person is in front of the camera.

To show, that the rotation dependency of the Viola and Jones face detector has been resolved in case of a device rotation, we provide experiments on 3 video sequences with different situations. The individual cases are classified according to Table 1. The results presented in the diagrams for each of the situations show if either the face has been detected (1) or not (0). The x-axes in the diagrams state the yaw-angle of the camera ego-motion from $-180°$ till $180°$.

Because of the additional rotation of every frame and the different image size we further analyzed the running time of both methods on an iPhone 4. For the face detection we used the OpenCV implementation [2], which is an improved version of the Viola and Jones algorithm, implemented by Lienhart [7].

Using an image size of 192×144 and a minimum patch size of 20 pixels, the running time of Viola and Jones face detector (OpenCV implementation) on an iPhone 4 is about 80 ms. Assume a $45°$ rotation, resulting in an image size of $\approx 238 \times 283$, the running time of Viola and Jones face detector is about 150 ms plus 12 ms needed for the rotation. Increasing the image size to 480×360, which correspond to a rotated image size of 594×594, the running times are 210 ms and 350 ms plus 70 ms for the image transformation. This means that the running time is, in the worst case, doubled. It seems that the OpenCV implementation of Canny pruning does not skip the additional regions. Skipping these regions, since there is no face, would speed up the presented approach.

(a) (b) (c) (d)

Figure 6: Example frames of all sequences: (a) Seq. 1: upright camera and upright face. Both algorithms detect the face; (b) Seq. 2: upright camera and rotated face. Both algorithms performed comparably; (c+d) Seq. 3: rotated camera and upright face. The proposed fusion outperformed the standard algorithm.

	CAMERA	FACE	SYMBOLIC
Sequence 1	straight	straight	
Sequence 2	straight	rotated	
Sequence 3	rotated	straight	

Table 1: Evaluation of test cases

Sequence 1, as shown in Figure 6a, analyzes whether the alteration and extension of the framework has a negative impact for the standard case. This is a situation of ideal conditions for the unmodified face detector. The face is almost perfectly aligned upright and thus corresponds to the expected value of the algorithm. The yaw-angle computed is almost zero, so that the individual frames are rotated with a small angle. The experimental video sequence consists of 165 frames and yaw-angles between $\approx -9°$ and $\approx 11°$.

Figure 7 shows, that the extension causes no loss of

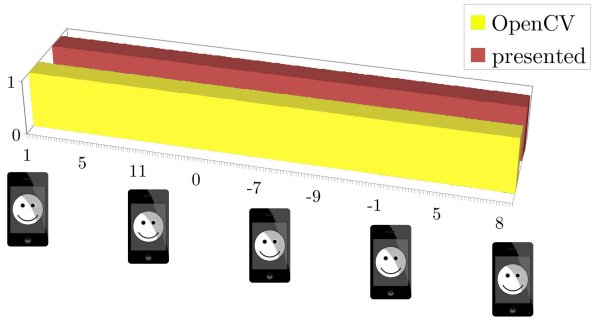

Figure 7: Detection-rate on video sequence 1. The camera and the face are almost perfectly aligned upright. Both algorithms detect the face in every frame. The y-axis states weather a face is detected (1) or not (0). The x-axis depicts the yaw-angle of the camera ego-motion.

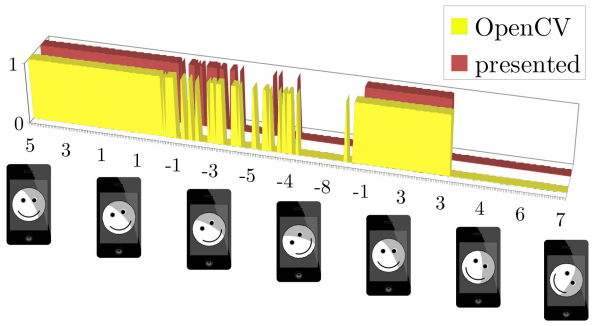

Figure 8: Detection-rate on video sequence 2. The camera is almost perfectly aligned upright and the face rotates to the left and to the right. Both algorithms perform comparably. The y-axis states weather a face is detected (1) or not (0). The x-axis depicts the yaw-angle of the camera ego-motion.

quality compared to the conventional algorithm, since both methods detect the face in every frame.

Similar to the first sequence, we did not change the direction of the camera orientation in sequence 2. The face, however, turns from the ideal position to a rotated one. As expected and shown in figure 6b, both algorithms fail to recognize the face after a certain amount of rotation. Since the camera was not perfectly aligned, the original algorithm performed slightly better. The experimental video sequence consists of 352 frames, a camera yaw angle between $\approx -8°$ and $\approx 8°$ and a face rotation around z_{world} between $\approx -30°$ and $\approx 45°$.

The graph in Figure 8 shows, that both algorithms achieve a similar performance in this case.

The third sequence analyzes the main weakness, which was the motivation for this paper. An upright face in front of the camera and the camera being rotated, see Figure 6c. For the evaluation we recorded a sequence with 472 frames

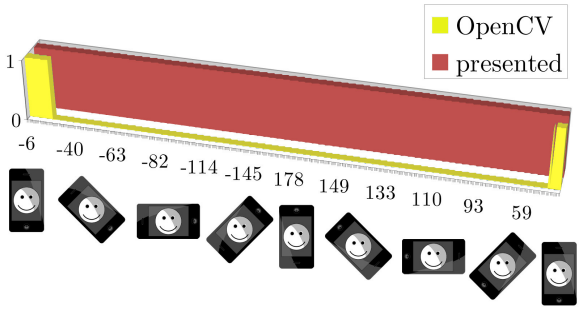

Figure 9: Detection-rate on video sequence 3. The face in front of the camera is upright and the camera performs a full 360°rotation. This is the main motivation of our paper. The proposed fusion of Viola and Jones face detector and inertial sensors to compensate the ego-motion of the camera clearly outperforms the standard approach. The y-axis states weather a face is detected (1) or not (0). The x-axis depicts the yaw-angle of the camera ego-motion.

	♯ FRAMES	OPENCV	FUSION
Sequence 1	165	100 %	100 %
Sequence 2	352	≈ 54 %	≈ 52 %
Sequence 3	472	≈ 7 %	100 %

Table 2: Comparison of detection rates using the OpenCV implementation of Viola and Jones and the proposed fusion with inertial sensors.

and the camera did a full 360° rotation. For the traditional Viola and Jones algorithm this means, that the face is no longer in the expected orientation and the detection fails, as shown in Figure 9. Using the inertial sensors to compute the camera ego-motion and align the face virtually however performed flawlessly and managed to find the face in 100% of the tested frames. For this case the integration of an IMU into face detection outperforms the standard algorithm.

Table 2 gives a complete overview of the detection rates on the tested sequences.

6. Conclusion

We presented an efficient fusion of an inertial measurement unit and boosting based face detection by Viola and Jones. Using the internal IMU, in our case the gyroscope of Apple's iPhone 4, we compute a virtual upright image and align the vertical axis. Running Viola and Jones face detector on the virtual upright image instead of the original gained much better results. In our experiments, we have shown that the proposed fusion is efficient and the computational overhead is insignificant. We showed, that our fusion clearly outperforms the standard approach if the camera rotates, while the face stays upright in front of the camera. Compared to other algorithms, which claim to be rotation invariant, our method is fast enough to run on a mobile device like Apple's iPhone 4. Since there is no test set available for the integration of inertial sensors into face detection algorithms we will publish our sequences and sensor data.

References

[1] Apple Inc., Cupertino, CA. *Event Handling Guide for iOS - Data Management: Event Handling*, 2010. 68, 69

[2] G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000. 67, 68, 69

[3] B. Clipp, R. Raguram, J.-M. Frahm, G. Welch, and M. Pollefeys. A mobile 3d city reconstruction system. Proc. IEEE VR workshop on Cityscapes, 2008. 67

[4] S. Du, N. Zheng, Q. You, Y. Wu, M. Yuan, and J. Wu. Rotated haar-like features for face detection with in-plane rotation. In *VSMM*, pages 128–137, 2006. 66

[5] Y. Freund and R. E. Schapire. Experiments with a new boosting algorithm. *Machine Learning: Proceedings of the Thirteenth International Conference*, pages 148–156, 1996. 66, 67

[6] M. Labrie and P. Hebert. Efficient camera motion and 3d recovery using an inertial sensor. In *Fourth Canadian Conference on Computer and Robot Vision, CRV '07.*, pages 55 –62, 2007. 67

[7] R. Lienhart and J. Maydt. An extended set of haar-like features for rapid object detection. *International Conference on Image Processing*, 2002. 69

[8] R. Murray, Z. Li, S. Sastry, and S. Sastry. *A mathematical introduction to robotic manipulation*. CRC, 1994. 69

[9] G. Pons-Moll, A. Baak, T. Helten, M. Müller, H.-P. Seidel, and B. Rosenhahn. Multisensor-fusion for 3d full-body human motion capture. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010. 67

[10] M. Ramachandran, A. Veeraraghavan, and R. Chellappa. A fast bilinear structure from motion algorithm using a video sequence and inertial sensors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(1):186 –193, 2011. 67

[11] J. Ren, N. Kehtarnavaz, and L. Estevez. Real-time optimization of viola-jones face detection for mobile platforms. In *Circuits and Systems Workshop: System-on-Chip-Design, Applications, Integration, and Software, 2008 IEEE Dallas*, pages 1–4, 2008. 67

[12] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2001. 66, 67

[13] B. Wu, H. Ai, C. Huang, and S. Lao. Fast rotation invariant multi-view face detection based on real adaboost. In *Sixth IEEE International Conference on Automatic Face and Gesture Recognition, 2004. Proceedings.*, pages 79–84, 2004. 66

[14] M.-H. Yang, D. Kriegman, and N. Ahuja. Detecting faces in images: a survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(1):34 –58, 2002. 66

[15] C. Zhang and Z. Zhang. A survey of recent advances in face detection. Microsoft Research Technical Report, MSR-TR-2010-66, 2010. 66