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
Outline

- Motivation
- Boosting Based Face Detection
- Inertial Measurement Unit
- Proposed Method
- Experimental Results
- Conclusion
P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. CVPR, 2001
Viola and Jones Face Detection

- well known appearance-based approach
- utilizes Adaboost in the classifier training
  - weak classifiers based on Haar-like features
- integral image for efficient feature computation
- standard in current commercial cameras

P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. CVPR, 2001
Viola and Jones Face Detection

- well known appearance-based approach
- utilizes Adaboost in the classifier training
  - weak classifiers based on Haar-like features
- integral image for efficient feature computation
- standard in current commercial cameras

- pros:
  - general object detection framework
  - good detection rates
  - fast detection
- drawbacks:
  - slow training
  - highly rotational variant

P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. CVPR, 2001
Viola and Jones
Face detection:
Motivation

Viola and Jones
Face detection:
Viola and Jones
Face detection:
Motivation

Viola and Jones
Face detection:
Viola and Jones
Face detection:
Rotation invariant detection based on Viola and Jones?

- Du et al.: set of rotated Haar like features
- Wu et al.: divide human faces according to different view points
Rotation invariant detection based on Viola and Jones?

- Du et al.: set of rotated Haar like features
- Wu et al.: divide human faces according to different view points

Computational too expensive for mobile devices (8-12 times slower)
Rotation invariant detection based on Viola and Jones?
- Du et al.: set of rotated Haar like features
- Wu et al.: divide human faces according to different view points

Computational too expensive for mobile devices (8-12 times slower)

Inertial sensors getting major attention
- markerless human motion capturing
- structure from motion
- ...

Inertial sensors are available on most current mobile devices
- mobile phones (iPhone, HTC Desire, ...)
- iPod touch, iPad, ...
Viola and Jones face detection framework (rotational variant)
Viola and Jones face detection framework (rotational variant) mobile devices with an IMU (inertial measurement unit)
Viola and Jones face detection framework (rotational variant)

mobile devices with an IMU (inertial measurement unit)

fusion?!
Adaboost is an algorithm for constructing a „strong“ classifier as a linear combination of simple „weak“ classifiers $h_t(x)$:

$$f(x) = \sum_{t=1}^{T} \alpha_t h_t(x)$$

final „strong“ classifier/hypothesis: $H(x) = \text{sign}(f(x))$
Haar-like Features

- Easy computation (scalable and changeable in size and position)
- Fast and efficient computation with integral-image

\[
I_\Sigma(x, y) = \sum_{i=0}^{x} \sum_{j=0}^{y} I(i, j)
\]

\[
A + D - (C + B)
\]
Inertial Measurement Unit

iPhone 4, 4G iPod Touch, iPad 2

two hardware sources:  
• three axis accelerometer  
• three axis gyroscope
World-coordinate System (defined by the IMU)

spanned by the axes: \( (x_{\text{world}}, y_{\text{world}}, z_{\text{world}}) \)

\[
z_{\text{world}} := - \frac{a_{\text{gravitation}}}{\|a_{\text{gravitation}}\|}
\]

\( x_{\text{world}}, y_{\text{world}} \) defined orthogonal to gravity (right-hand space)
World-coordinate System (defined by the IMU)

spanned by the axes: \((x_{\text{world}}, y_{\text{world}}, z_{\text{world}})\)

\[ z_{\text{world}} := -\frac{a_{\text{gravitation}}}{\|a_{\text{gravitation}}\|} \]

\(x_{\text{world}}, y_{\text{world}}\) defined orthogonal to gravity (right-hand space)

Apple provides a proprietary algorithm to calculate absolute orientations relative to the static world-coordinate system.
Camera-coordinate system:

We are interested in the rotation around the z-axis (yaw-angle)
Proposed Method
Proposed Method

Apples proprietary algorithm gives:

\[(0, i_{\text{device},t}) = q(0, i_{\text{world}})q^{-1} \quad (i \in \mathbb{R}^3)\]
Apples proprietary algorithm gives:

\[(0, i_{\text{device}}, t) = q(0, i_{\text{world}})q^{-1} \quad (i \in \mathbb{R}^3)\]

project \(z_{\text{world}}\) on the devices screen surface:

\[z_{\text{proj}, t} = z_{\text{world}} - \langle z_{\text{world}} | z_{\text{device}, t} \rangle \cdot z_{\text{device}, t}\]
Proposed Method

Apples proprietary algorithm gives:

\[(0, i_{\text{device}}, t) = q(0, i_{\text{world}})q^{-1} \quad (i \in \mathbb{R}^3)\]

project \(z_{\text{world}}\) on the device's screen surface:

\[z_{\text{proj}}, t = z_{\text{world}} - \langle z_{\text{world}}|z_{\text{device}}, t \rangle \cdot z_{\text{device}}, t\]

compute the yaw-angle:

\[\cos(\varphi_{\text{yaw}}) = \pm \langle y_{\text{device}}, t|z_{\text{proj}}, t/||z_{\text{proj}}, t||\rangle\]

having the yaw-angle, we rotate the camera image to produce a virtual upright image
Proposed Method

Camera coordinate system

project $z_{\text{world}}$ on the device screen surface

compute the yaw-angle
Standard Approach

image comming from mobile device

Viola + Jones face detection
Proposed Method

- Image coming from mobile device
- IMU

**Compensate ego-motion**

- Compute yaw-angle

- Generate virtual upright image

- Viola + Jones face detection
Proposed Method
Proposed Method
Proposed Method
Proposed Method
### Experimental Results

<table>
<thead>
<tr>
<th>Camera</th>
<th>Face</th>
<th>Symbolic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sequence 1</td>
<td>straight</td>
<td>straight</td>
</tr>
<tr>
<td>Sequence 2</td>
<td>straight</td>
<td>rotated</td>
</tr>
<tr>
<td>Sequence 3</td>
<td>rotated</td>
<td>straight</td>
</tr>
</tbody>
</table>

Table 0: Evaluation of test cases
### Experimental Results

#### Table: Evaluation of test cases

<table>
<thead>
<tr>
<th>Sequence</th>
<th>Camera</th>
<th>Face</th>
<th>Symbolic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sequence 1</td>
<td>straight</td>
<td>straight</td>
<td>![Icons]</td>
</tr>
<tr>
<td>Sequence 2</td>
<td>straight</td>
<td>rotated</td>
<td>![Icons]</td>
</tr>
<tr>
<td>Sequence 3</td>
<td>rotated</td>
<td>straight</td>
<td>![Icons]</td>
</tr>
</tbody>
</table>

- **Sequence 1**: The camera and the face are almost perfectly aligned upright. Both algorithms detect the face in every frame.
- **Sequence 2**: The camera is almost perfectly aligned upright and the face rotates to the left and to the right. Both algorithms perform comparably.
- **Sequence 3**: An upright face in front of the camera and the camera being rotated. The graph shows that both algorithms achieve a similar performance in this case.

---

**Graph**

- The graph in Figure shows that both algorithms achieve a similar performance in this case.
- The 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 4, 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 0 frames, a camera yaw angle between ±9° and ±11°, and a face rotation around z_world between ±30° and ±45°.

---

**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 6. For the evaluation, we recorded a sequence with 0 frames.

---

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

<table>
<thead>
<tr>
<th>Camera</th>
<th>Face</th>
<th>Symbolic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sequence 1</td>
<td>straight</td>
<td>straight</td>
</tr>
<tr>
<td>Sequence 2</td>
<td>straight</td>
<td>rotated</td>
</tr>
<tr>
<td>Sequence 3</td>
<td>rotated</td>
<td>straight</td>
</tr>
</tbody>
</table>

Figure 1: Example frames of all sequences: fia1 Seq6 0: upright camera and upright face. Both algorithms detect the face; fib1 Seq6 .: upright camera and rotated face. Both algorithms performed comparably; fic3d1 Seq6 :: rotated camera and upright face. The proposed fusion outperformed the standard algorithm.

Table 1: Evaluation of test cases

Sequence 04 as shown in Figure a 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 0-; frames and yaw angles between $\approx -9^\circ$ and $\approx 11^\circ$.

Figure (b) shows that the extension causes no loss of 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 .6. The face, however, turns from the ideal position to a rotated one. As expected and shown in figure b, 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 0-; frames, a camera yaw angle between $\approx -8^\circ$ and $\approx 8^\circ$, and a face rotation around $z$ world between $\approx -30^\circ$ and $\approx 45^\circ$.

The graph in Figure c 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 c. For the evaluation we recorded a sequence with 0-; frames.
Experimental Results

<table>
<thead>
<tr>
<th>Camera</th>
<th>Face</th>
<th>Symbolic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sequence 1</td>
<td>straight</td>
<td>straight</td>
</tr>
<tr>
<td>Sequence 2</td>
<td>straight</td>
<td>rotated</td>
</tr>
<tr>
<td>Sequence 3</td>
<td>rotated</td>
<td>straight</td>
</tr>
</tbody>
</table>

Figure a: Example frames of all sequences: fia1 Seq6: upright camera and upright face. Both algorithms detect the face; fib1 Seq6: upright camera and rotated face. Both algorithms performed comparably; fic3d1 Seq6: rotated camera and upright face. The proposed fusion outperformed the standard algorithm.

Table 0: Evaluation of test cases

- Sequence 4 as shown in Figure a 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 0-10 frames and yaw angles between $\approx -9^\circ$ and $\approx 11^\circ$.

- Figure b: Detection rate on video sequence 4. The camera and the face are almost perfectly aligned upright. Both algorithms detect the face in every frame. The y-axis states whether a face is detected (1) or not (0). The x-axis depicts the yaw angle of the camera ego-motion.

- Similar to the first sequence, we did not change the direction of the camera orientation in sequence 6. The face, however, turns from the ideal position to a rotated one. As expected and shown in Figure a, 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 10-20 frames, a camera yaw angle between $\approx -8^\circ$ and $\approx 8^\circ$, and a face rotation around the z-world-axis between $\approx -30^\circ$ and $\approx 45^\circ$.

- The graph in Figure c 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 a. For the evaluation, we recorded a sequence with 10-20 frames (8...).
• proposed an efficient fusion of IMU and boosting based face detection by Viola and Jones
Conclusion

• proposed an efficient fusion of IMU and boosting based face detection by Viola and Jones

• compared to other (rotation invariant) boosting based method our method is fast enough for a mobile device
Conclusion

• proposed an efficient fusion of IMU and boosting based face detection by Viola and Jones

• compared to other (rotation invariant) boosting based method our method is fast enough for a mobile device

• video sequences + motion data will be available
Conclusion

• proposed an efficient fusion of IMU and boosting based face detection by Viola and Jones

• compared to other (rotation invariant) boosting based method our method is fast enough for a mobile device

• video sequences + motion data will be available

• online demo is available (just ask)
• proposed an efficient fusion of IMU and boosting based face detection by Viola and Jones

• compared to other (rotation invariant) boosting based method our method is fast enough for a mobile device

• video sequences + motion data will be available

• online demo is available (just ask)

• compensate face motion is an ongoing project
• proposed an efficient fusion of IMU and boosting based face detection by Viola and Jones

• compared to other (rotation invariant) boosting based method our method is fast enough for a mobile device

• video sequences + motion data will be available

• online demo is available (just ask)

• compensate face motion is an ongoing project

Thanks for your attention!!!