Motion Detection as Interaction Technique for Games & Applications on Mobile Devices

Stephan A. Drab
Upper Austria University of Applied Sciences Hagenberg, Mobile Computing
Interaction Techniques on Mobile Devices

- Common Interaction Techniques:
  - Keys
  - Stylus
  - Voice Recognition
### Concept of the Motion Detection Technique

**Classification in Terms of Usability:**

<table>
<thead>
<tr>
<th></th>
<th>Very Good</th>
<th>Good</th>
<th>Very Bad</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reaction time</td>
<td>very good</td>
<td>good</td>
<td>very bad</td>
</tr>
<tr>
<td>Input quantity</td>
<td>very good</td>
<td>medium</td>
<td>bad</td>
</tr>
<tr>
<td>Intuitivity</td>
<td>very bad</td>
<td>good</td>
<td>very good</td>
</tr>
</tbody>
</table>
Concept of the Motion Detection Technique

- Optical Markerless Inertial 2D-Tracking
- Idea: Device Motion => Motion in Camera Images
Concept of the Motion Detection Technique

- Optical Markerless Inertial 2D-Tracking
- Idea: Device Motion => Motion in Camera Images

- Analysis of Motion in Camera Images
- Very intuitive User interaction
**Concept of the Motion Detection Technique**

- **Classification in Terms of Usability:**

<table>
<thead>
<tr>
<th>Criteria</th>
<th>Very Good</th>
<th>Good</th>
<th>Very Bad</th>
<th>Good</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reaction time</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Input quantity</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Intuitivity</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

 Stephan A. Drab, Upper Austria University of Applied Sciences Hagenberg, Mobile Computing
Common Tracking Algorithms

- **Algorithms capable of analysing Scene Motion**
  - Edge Detection and Tracking
  - Analysis of Scene Components
  - Block Matching
  - Analysis of Optical Flow
Common Tracking Algorithms

- **Edge Detection and Tracking**
  - 3D Tracking of Edges and Vertices
Common Tracking Algorithms

- **Analysis of Scene Components**
  - Analysis of Movement of 2D Scene Components
Common Tracking Algorithms

- **Block Matching**
  - Method from Video Compression Sector

- Interpret Block References as Motion Vectors
- Motion Estimation = \((\text{sum of all } v_{\text{motion}}) \times \text{block\_size/\text{nr\_blocks}}\)
- In our example: \((8/9; 8/9) \approx (1; 1)\)
Common Tracking Algorithms

- Analysis of Optical Flow
Common Tracking Algorithms

- Summary:
  - Edge Detection and Tracking
    - Computing time dependent on Scene Complexity (AR, 3D)
  - Analysis of Scene Components
    - Computing time dependent on Scene Complexity (2D)
  - Block Matching
    - Image sizes very small (Mozzies)
  - Analysis of Optical Flow
    - Fairly interactive framerates (Sweep Technology)

=>

Development of an **Efficient Optical Markerless Inertial 2D-Tracking Algorithm**
Motion Detection Algorithm

- **Starting Basis:**
  - We have two Successive Images captured by the Camera:
Motion Detection Algorithm

- **Starting Basis:**
  - We have two successive images captured by the camera:

- **Wanted:**
  - The relative motion vector of the image contents
Motion Detection Algorithm

- **Idea**
  - Correlate the images using every possible 2D Shift
Motion Detection Algorithm

• **Idea**
  - Correlate the images using every possible 2D Shift
  - Algorithm too complex by Orders of Magnitude
    - Computing Time dependent on 4th power of image size
    - Reduce amount of Information to Correlate
Motion Detection Algorithm

- **Concept**
  - Project the Image onto its X- and Y-Axis
**Motion Detection Algorithm**

- **Concept**
  - Project the Image onto its X- and Y-Axis
  - Calculate Color MSE for Overlaps of every 1D-Shift

<table>
<thead>
<tr>
<th>Shift</th>
<th>MSE</th>
<th>Shifted Projection Buffers</th>
</tr>
</thead>
<tbody>
<tr>
<td>-4</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>-3</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>-2</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>-1</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>21</td>
<td></td>
</tr>
<tr>
<td>+1</td>
<td>23</td>
<td></td>
</tr>
<tr>
<td>+2</td>
<td>35</td>
<td></td>
</tr>
<tr>
<td>+3</td>
<td>63</td>
<td></td>
</tr>
<tr>
<td>+4</td>
<td>101</td>
<td></td>
</tr>
</tbody>
</table>
**Motion Detection Algorithm**

- **Concept**
  - Project the Image onto its X- and Y-Axis
  - Calculate Color MSE for Overlaps of every 1D-Shift
  - Best Matching Shift is the Actual Motion
Motion Detection Algorithm

• **Concept**
  - Project the Image onto its X- and Y-Axis
  - Calculate Color MSE for Overlaps of every 1D-Shift
  - Best Matching Shift is the Actual Motion
  - Computing time
    • $n \cdot \text{size}^4 \Rightarrow o \cdot \text{size}^2 + 2 \cdot p \cdot \text{size}^2$
Motion Detection Algorithm

- **Advantages**
  - Fast Motion Detection Algorithm
  - Capable of running on a Mobile Device
  - works with Greyscale Images (YUV)
  - Relatively Resistant to
    - Rotation
    - Scaling
Motion Detection Algorithm

- **Advantages**
  - Fast Motion Detection Algorithm
  - Capable of running on a Mobile Device
  - works with Greyscale Images (YUV)
  - Relatively Resistent to
    - Rotation
    - Scaling

- **Disadvantage**
  - The algorithm estimates the Motion inexact when pointing the camera at:
    - repetitive patterns
    - images with a low dynamic range, e.g.
      - dark spots
      - white walls
Demonstrators

- Demonstrators for the Concept of MotionDetection as Interaction Technique on Mobile Devices
  - CameraCursor
  - TestApplication
  - TheBiggerPicture
  - LabyrinthGame
  - MapNavigator
Demonstrators – CameraCursor

- Test of Cursor Concept
- Motion Information moves Cursor
- Lines can be drawn
Demonstrators – TestApplication

- Test & Debug of MotionDetection
- Motion Information is visualized by a red cross
- The cross always points at the same spot
Demonstrators – The Bigger Picture

- Multiple Camera Images combined to one
- Motion Information tells where to Accumulate the Captured Frame
- Works in Real-Time
Demonstrators – LabyrinthGame

- **PhysicsDemo of Labyrinth Game** (remake of the old style wooden version)
- **Pitch of Mobile Device** accelerates the sphere.
- **Visualization of Motion** Information acceleration, speed and position of sphere
- **Further graphic enhancement** needed, e.g.
Demonstrators – MapNavigator

- Map Navigation Tool
- Designed to view large maps
- Motion Information moves the viewing window
Further Development & Future Work

- User Tests regarding Usability & Acceptance
Further Development & Future Work

- User Tests regarding Usability & Acceptance

- **MotionDetection**
  - Further Optimization of the MotionDetection
  - Detection of Rotation and Scaling
Further Development & Future Work

- User Tests regarding Usability & Acceptance

- **MotionDetection**
  - Further Optimization of the MotionDetection
  - Detection of Rotation and Scaling

- **Integration of the Cursor Concept in the native GUI of WindowsCE**
## Further Development & Future Work

- **User Tests regarding Usability & Acceptance**

- **MotionDetection**
  - Further Optimization of the MotionDetection
  - Detection of Rotation and Scaling

- **Integration of the Cursor Concept in the native GUI of WindowsCE**

- **Motion Gestures**
Any Questions?

contact:
stephan.drab@fh-hagenberg.at
mc.fh-hagenberg.at