Improved Gesture Trajectory Reconstruction from Acceleration Values for a Device Independent Gesture Recognition Approach

Mathias Wilhelm, Dirk Roscher, Sahin Albayrak
DAI-Labor/TU Berlin, Ernst-Reuter-Platz 7, 10587 Berlin, Germany
{First.Lastname}@dai-labor.de

Keywords: Device independence, device independent gesture recognition, gesture recognition, trajectory-based gesture recognition, shape analysis, multimodal interaction

1 Introduction

With the persistent spreading of computer technology in our everyday life and the resulting interaction between human and computer in many different situations, hand gesture interaction is one of the many interaction techniques becoming more and more important. Several different gesture devices like cameras, devices with integrated accelerometers or computer mice can be used for hand gesture interaction. Each device has different properties (like intrusiveness or handling) which influence its usability for different applications and situations. However, most applications providing gesture-based interaction support only one gesture device which restrict users especially if the application should be used in different situations. One reason is that developers have to pursue a higher implementation effort to support different gesture devices for the interaction process as each gesture device requires the utilization of a different gesture recognition algorithm. Furthermore, the effort of the user is also increased as every device and every recognition algorithm has to be trained individually. The most recent research solves the issue of device independence only for a limited group of gesture devices such as data gloves [3] or 3-axis accelerometers [4]. The approach presented in [5] goes one step further and accepts gesture input from cameras, accelerometers, and pen devices. It thus supports device independent hand gesture recognition for the most known gesture device types.

The approach extracts user’s gesture trajectory performed in space by applying different algorithms for different types of gesture data from different devices (i.e. images, acceleration or position data). The extracted trajectory is a device independent data representation of the performed gesture. The reconstructed trajectories from different gesture devices are very similar which allows the application of a single training set for different gesture devices. In the next step, the reconstructed trajectories are resampled to a certain number of points and transformed into the Kendall shape space [2], a scale- and position-free representation of the resampled trajectory in the complex space. Finally, the shape representation is compared to each shape template stored in a database by a rotation invariant distance measurement. Further details of the approach and results of extensive experiments are...
proposed in [5]. These experiments demonstrated besides the feasibility of the approach a lower recognition rate for the shapes generated from acceleration values. In this paper, the causes are analyzed and the extensions of the approach presented in [5] are explained in the following.

2 Analysis

The fundamental reason for the low recognition rates of gesture trajectories obtained from acceleration values is founded in the low trajectory reconstruction performance (Fig. 1). In further experiments, the trajectory reconstruction from acceleration values were studied and revealed two main influences: (1) the error resulting from the performed double integration of the accelerometer values and (2) the influence of the gravity.

The integration method used in the previous work [5] applies the Euler backward rule which simply sums up all values:

\[ y_t = y_{t-1} + dt \dot{y}_t, \]

where \( dt \) represent the sample period. This kind of integration depends very much on the sampling period and provides a large approximation error for the acceleration values. To reduce the drift resulting from the two integration steps, a constant is applied which causes a reduction of the approximation error at each integration step. This effect is reached by gradually leaking a small amount of the integrator output and causes a smoother, offset-free integration result. This type of integration is also called leaky integration and is formally defined as:

\[ y_t = \alpha y_{t-1} + dt \dot{y}_t, \]

where \( \alpha \) is the leaky constant ranging between 0 and 1. Another advantage of the leaky integration is the increased robustness in case of noisy signals. Consequently, the Kalman filter as proposed in [5] is no longer required.

The second component that significantly affects the result of the trajectory reconstruction from acceleration data is the gravity. In the previous work [5], a high pass filter is applied to remove the gravity. However, the study showed that the gravitational effect is usually not constant in the domain of dynamic hand gesture recognition because users tend to tilt and rotate the gesture device during the gesture execution. Consequently, the gravitational influence is a dynamic and low frequency influence that depends on the grade of change in tilt and rotation of the accelerometer during the gesture execution. It is difficult to select an optimal cutoff frequency for the high pass filter. When the cutoff frequency is too high, then parts of the gesture signal will be removed too. This leads to a disturbed signal and thus to an inadequate gesture trajectory. On the other hand, a too low cutoff frequency does not completely remove the gravitational effect. Different methods were evaluated for removing the gravitation such as an analysis in the frequency domain, an inertial navigation system (INS) based approach [1], and a zero baseline approach. With the complexity and the grade of compensation of the gravitational influence in mind, the zero baseline approach showed the best results. This approach is based on computing the mean of the acceleration signal of the segmented gesture and subtracting this mean from each acceleration value. This procedure causes an acceleration signal baseline at zero.
3 Experiments

To evaluate the influence of the improved trajectory reconstruction from acceleration values (Fig. 2) on the recognition results, two experiments are performed with the same gestures and the same OpenInterface-based implementation as used in [5]. In the first experiment, the same experiment setup including the same training files as used in [5] are used to evaluate the grade of improvement with a Nintendo Wii controller. The recognizer was trained with a computer mouse. For each gesture 10 examples were recorded and resampled with 7 points. The leaky constant $\alpha$ of the acceleration integrator component was set to 0.99. The user performed 100 test examples for each gesture with the Nintendo Wii controller and with the training device. The experiment states out a significant recognition improvement of 11% in comparison to the previous work (Table 1). However, some gesture shapes obtained from acceleration data offered a low recognition performance in despite of accurate reconstructed trajectories. For instance, correct reconstructed trajectories of gesture ‘Eight’ (Fig. 2) were often recognized as gesture ‘Left’ or ‘Circle’ (Fig. 3). This issue rises from the fact, that the lower part of the eight from acceleration data is usually much smaller than the eight obtained from the computer mouse. The same issue occurs with gesture ‘Triangle’ which was correctly reconstructed but was often classified as gesture ‘Cross’ or ‘Circle’ (Fig. 4).

<table>
<thead>
<tr>
<th>Device</th>
<th>Right</th>
<th>Left</th>
<th>Z</th>
<th>Shake</th>
<th>Triangle</th>
<th>E</th>
<th>Loop</th>
<th>Cross</th>
<th>Circle</th>
<th>Eight</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mouse</td>
<td>1.00</td>
<td>1.00</td>
<td>0.99</td>
<td>0.98</td>
<td>1.00</td>
<td>0.99</td>
<td>1.00</td>
<td>0.96</td>
<td>0.98</td>
<td>0.96</td>
<td>0.98</td>
</tr>
<tr>
<td>Wii</td>
<td>0.94</td>
<td>0.93</td>
<td>0.90</td>
<td>0.69</td>
<td>0.40</td>
<td>0.96</td>
<td>0.80</td>
<td>0.65</td>
<td>0.90</td>
<td>0.34</td>
<td>0.75</td>
</tr>
<tr>
<td>Wii/5f</td>
<td>0.70</td>
<td>0.65</td>
<td>0.93</td>
<td>0.60</td>
<td>0.43</td>
<td>0.82</td>
<td>0.76</td>
<td>0.60</td>
<td>0.93</td>
<td>0.20</td>
<td>0.66</td>
</tr>
</tbody>
</table>

In the second experiment, the system was trained with a Nintendo Wii controller and 10 examples for each gesture were recorded. The point number of the resampled trajectories was 10 and the leaky constant $\alpha$ was 0.99. The recognizer was tested with the Nintendo Wii controller and a computer mouse and the user who trained the recognizer performed 100 test examples with each device for each gesture. The results

---

1 Project site of the OpenInterface platform: http://www.openinterface.org/platform/
of this experiment show a recognition improvement of 10% in comparison with the previous work (Table 2) but these rates are below the rates from experiment one.

<table>
<thead>
<tr>
<th>Device</th>
<th>Right</th>
<th>Left</th>
<th>Z</th>
<th>Shake</th>
<th>Triangle</th>
<th>E</th>
<th>Loop</th>
<th>Cross</th>
<th>Circle</th>
<th>Eight</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mouse</td>
<td>0.98</td>
<td>0.97</td>
<td>1.00</td>
<td>0.10</td>
<td>0.76</td>
<td>1.00</td>
<td>0.69</td>
<td>0.76</td>
<td>0.65</td>
<td>0.21</td>
<td>0.72</td>
</tr>
<tr>
<td>Wii</td>
<td>0.99</td>
<td>0.90</td>
<td>0.89</td>
<td>0.40</td>
<td>0.80</td>
<td>0.98</td>
<td>0.92</td>
<td>0.81</td>
<td>0.79</td>
<td>0.51</td>
<td>0.80</td>
</tr>
<tr>
<td>Mouse [5]</td>
<td>0.99</td>
<td>0.63</td>
<td>0.95</td>
<td>0.93</td>
<td>0.00</td>
<td>0.38</td>
<td>0.89</td>
<td>0.40</td>
<td>0.94</td>
<td>0.15</td>
<td>0.62</td>
</tr>
</tbody>
</table>

4 Conclusion and Further Work

In this paper, improvements for the reconstruction of the gesture trajectory from acceleration values were presented which significantly boost the overall recognition result in comparison to the results presented in [5]. However, the experiments showed that a better resampling or shape point selection algorithm and a control of the rotation invariant measurement are needed. Furthermore, they also showed that the overall recognition rate depends on the applied gesture device. To overcome this device dependence of the training data, the application of an artificial generated training set should be studied in further work.

Fig. 3. Illustration of the similarity of scaled and rotated gestures ‘Eight’ and ‘Left’.  
Fig. 4. Illustration of the similarity of scaled and rotated gestures ‘Triangle’ and ‘Cross’.

References