A Dynamic Model for Real-Time Tracking of Hands in Bimanual Movements

Atid Shamaie and Alistair Sutherland

Centre for Digital Video Processing, School of Computer Applications,
Dublin City University, Dublin 9, Ireland
{ashamaie, alistair}@computing.dcu.ie

Abstract. The problem of hand tracking in the presence of occlusion is addressed. In bimanual movements the hands tend to be synchronised effortlessly. Different aspects of this synchronisation are the basis of our research to track the hands. We propose a Kalman filtering-based dynamic model to catch the movement of the hands. The spatial synchronisation in bimanual movements is modelled by the position and the temporal synchronisation by the velocity and acceleration of each hand. Based on this model, we introduce algorithms for occlusion detection and hand tracking during occlusion.

1 Introduction

Many tasks are done everyday by our two hands together. In order to achieve our goal, our hands' movement have to be finely coordinated. Activities where bimanual coordination is important include clapping, opening a wine bottle, typing on a keyboard, eating with knife and fork, drumming, guiding a pilot driving an aircraft to the parking, showing the size of something (e.g. a fish), etc. Tracking the hands in a bimanual movement is important in order to recognise the meaning of the movement.

Hand tracking and gesture recognition have been widely addressed in the literature [1]. Spatio-temporal hand gesture recognition [2], hidden Markov models for gesture recognition [3], parametric hidden Markov models [4], HMM-based threshold models for gesture recognition [5], position based gesture recognition [6], tracking interacting hands using Bayesian networks [7], tracking of articulated structures in disparity maps derived from stereo image sequences [8], tracking of multiple articulated objects in the presence of occlusion in moderately complex scenes [9], representing and recognising human body motion [10], hand tracking for behavior understanding [11] and many other techniques have been used to deal with the problems of tracking and recognition of hand and body gestures.

In this paper we introduce a new tracking algorithm based on a dynamic model and Kalman filtering for bimanual movements. In the next section the main problem is addressed. In Section 3, we will discuss bimanual movements in more detail. In Section 4, the process of segmentation and extraction of hands from the background is given briefly. Section 5 is dedicated to the dynamic model and the algorithms for occlusion detection and tracking. Some experimental results are presented in Section 6. Comparisons to the other works and conclusion are presented at the end of paper.

2 Tracking Hands in Bimanual Movements

In a bimanual movement, when one hand, completely or partially, covers the other hand resuming tracking accurately at the end of occlusion is crucial. Since we don't know what happens during occlusion, after the end of occlusion, we have to know which hand in the image is the right hand and which one is the left. The hands may pass each other, movement types *a*, *c*, *d*, and *h* shown in Fig. 1(a), 1(c), 1(d), and 1(h), or may collide and return in opposite direction, types *b* and *g* presented in Fig. 1(b) and 1(g). In some cases they may not to collide but return in opposite direction, types *e* and *f* shown in Fig. 1(e) and 1(f).

3 Bimanual Coordination

In bimanual movements, naturally, there is some sort of synchronisation between the hands [12]. This synchronisation appears in temporal and spatial forms [13]. Temporally, when the two hands reach for different goals, they start and end their movements simultaneously [13]. Spatially, we are almost not able to draw a circle and a rectangle by the two hands at the same time [13]. Researchers have presented different anatomical and perceptual explanations for this phenomenon [14][15]. The temporal synchronisation in bimanual movements enables us to detect the concurrent hand pauses. These pauses help us to track the hands during occlusion. In order to detect the hand pauses we monitor the hand velocities. A very well known experiment called Circle Drawing shows that the two hand velocities are highly synchronised with no phase difference in bimanual movements [16]. This synchronisation is the basis of the algorithm proposed here.

4 Hand Extraction and Pre-processing

In order to extract the hand from the background we use colour segmentation and an algorithm called Grassfire [17]. The Grassfire algorithm scans the image from left to right, top to bottom to find connected areas with the colour belonging to the hands. The first connected area is labeled number 1, the second is labeled number 2, and so on. However, because of the search manner of Grassfire, in two consequent image frames the hands may be labelled differently. The other problem is occlusion. As soon as the hands reach each other the Grassfire algorithm finds a big blob and labels it as one object. Detecting occlusion and tracking the hands are the problems to be addressed in the next section.

5 A Dynamic Model for Occlusion Detection and Tracking

In this section, first we deal with the problem of detecting occlusion and then tracking the hands.

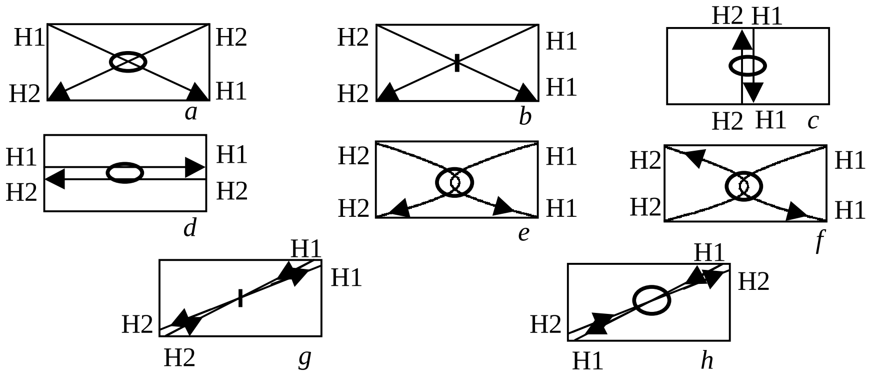

Fig. 1. The path of the hands in the 8 types of the bimanual movements. The thick ellipses represent the occlusion areas (*a, c, d, e, f, and h*), and the small lines represent collision (*b and g*)

5.1 Occlusion Detection

In order to detect occlusion, for every hand in an image a rectangle is constructed around it. The sides of each rectangle represent the left most, right most, top and bottom of the hand. By tracking the rectangles if there is any intersection between them it can be recognised as an alarm for occlusion. However, it is possible that without any prior intersection occlusion happens, Fig 2, and the Grassfire algorithm detects only one area, which is the same as the case that one hand hides behind a part of body. We use a model to predict future movement of each hand and catch occlusion.

5.2 A Dynamic Model

We propose a Kalman filtering-based algorithm to track the movement and predict the future position of the rectangles a few steps in advance. Every side of a rectangle is modelled by a dynamic model. This gives us model independence from the hand shapes. The position, velocity and acceleration of each side are considered in this model. These parameters are related together [18] based on the Equation 1 for *i*=1, 2 for the two hands and *j*=1, 2 for the left and right or top and bottom sides.

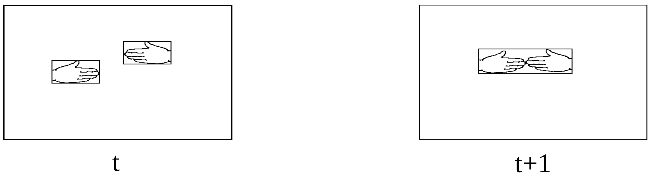

Fig. 2. A rectangle is formed around the hands or the big blob in occlusion

$$\begin{cases} x^i_{j,k+1} = x^i_{j,k} + h\dot{x}^i_{j,k} + \frac{1}{2}h^2\ddot{x}^i_{j,k} \\ \dot{x}^i_{j,k+1} = \dot{x}^i_{j,k} + h\ddot{x}^i_{j,k} \end{cases} . \tag{1}$$

where *h>0* is the sampling time [18], *k* is the time index, x is the position, $\dot{x}$ the velocity and $\ddot{x}$ the acceleration [18] of every side of a rectangle. In the normal Kalman filtering loop [19] the estimation of a state vector representing a rectangle side is updated with the current measurement of the position of the side and its prior estimation calculated in the last iteration of the loop. We use this prior estimation to predict the next position of each side of a rectangle one step in advance. If, by prediction, there is any intersection between the rectangles of the two hands, an occlusion alarm is set. Therefore, we are able to forecast a hand-occlusion before it actually happen.

5.3 Hand Tracking

By defining the centre of each hand and comparing them in consecutive images the problem of mislabelling in the consecutive frames can be resolved. By using this technique we are able to track the hands correctly even when something else like a part of body occludes them. It can be done by keeping records of the last position of the hand before occlusion and the position of the other hand. This is expected because when a hand moves behind something like the body or moves out of the image frame it most probably appears in an area close to the last position before the occlusion.

In order to track the hands when they occlude each other we introduce a technique based on the Kalman filtering model of section 5.2. As in Section 5.1 a rectangle around each hand is constructed. As soon as the occlusion is detected by our system a big rectangle around the big blob is formed. We use the Kalman filtering model of the last section to model the big rectangle. During occlusion, if the vertical sides of the rectangle stop, the velocities of these sides reach zero (or a small number with respect to their normal velocity). This is recognised as a horizontal pause of the hands. Therefore, they would go back horizontally. However, if the hands pass each other we observe a sign change in the velocities of the vertical sides without detecting any pause. These are similarly observable in the vertical or diagonal movements of the hands. By catching the hands pause during occlusion and comparing the current position of the hands at the end of occlusion, with respect to each other, with their position prior to occlusion we can decide the correct position of each hand. We label the sides of the big rectangle during occlusion with *a* and *b* for the horizontal and *c* and *d* for the vertical sides. The following measurements are defined,

$$\begin{cases} V_v = \sqrt{v_a^2 + v_b^2} \\ V_h = \sqrt{v_c^2 + v_d^2} \end{cases} . \tag{2}$$

where v_a and v_b stand for the velocity of horizontal sides *a* and *b*, v_c and v_d stand for the velocity of vertical sides *c* and *d*. Therefore, if any of these measurements get close to zero, say less than a small value of ε , a vertical or horizontal pause is detected. For example, if the hands have a movement of type *d*, Fig. 1(d), in which the hands just pass each other with no pause and no collision we detect no horizontal pause and conclude that the position of the hands is the opposite of their position prior to occlusion. In Figure 1(g), however, the hands collide and the model detects it. Therefore, the correct positions of the hands at the end of occlusion are the same as before the occlusion. In some of the movements like types *c* and *d*, Fig. 1(c) and 1(d), a horizontal or vertical pause may be detected during occlusion. This may cause a wrong decision by the algorithm. In order to deal with this we take the relative velocity of the rectangle sides into account. The standard deviation of the velocity differences of the horizontal sides, S_v, and vertical sides, S_h, are used. This labelling is used because the horizontal sides represent the vertical movement of the hands and vice versa. If a small standard deviation in one of these measurements is observed we do tracking by detecting the pauses in the movement of the other sides of the rectangle. The following is an abstract of the decision-making algorithm,

```
1.  If $S_v$ is less than a particular value
    1.A. If $V_h < \varepsilon$ then: the hands are horizontally back to their original position
    1.B. Else: the hands horizontally passed each other
2.  Else: if $S_h$ is less than a particular value
    2.A. If $V_v < \varepsilon$ then: the hands are vertically back to their original position
    2.B. Else: the hands vertically passed each other
3.  Else: if $V_h < \varepsilon$ then: the hands are horizontally back to their original position
4.  Else: if $V_v < \varepsilon$ then: the hands are vertically back to their original position
5.  Else: the hands passed each other
```

6 Experimental Results

For the sake of brevity, we just look at the results of the dynamic model in tracking and predicting one of the sides of a rectangle. The results for the others are similar. Fig. 3 shows a part of an experiment. The predicted next position, the triangles, at each time is pretty close to the actual position of the side of the rectangle at the next step, the solid small circles.

The velocities of the rectangle sides during occlusion show a kind of uniformity in some of the movements like types *a*, *b*, *d*, and *e*, Fig. 1(a), 1(b), 1(d), and 1(e). The result of an experiment is shown in Fig. 4(a). In this experiment the difference of the velocities at each time slice is less than a small value represented by the vertical lines in the graph. This uniformity causes small values of standard deviations of the relative velocities. This is true also for the vertical sides in movements of type *c*, Fig. 1(c).

However, the velocities of the sides of the rectangle in cases where the hands pass each other in opposite directions (e.g. movement type *h*) show opposite movements, Fig. 4(b). Large standard deviation in this case enables us to detect the non-uniform movements of the parallel sides of the rectangle. A change in the sign of the velocities in this figure is observable. When one hand passes the other they push the sides in the opposite directions. A few images of a movement of type *h* are shown in Fig. 5. For a movement of type *g*, in which the hand collision is detected by the model , the graphs

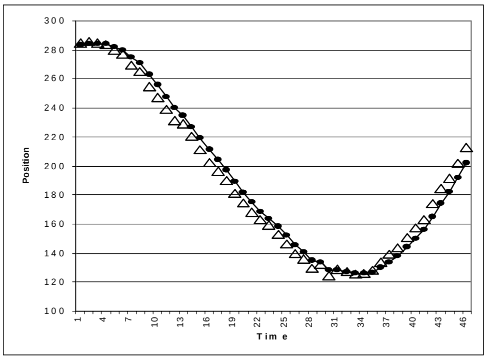

Fig. 3. The result of predicting by the dynamic model in a hand movement. The solid small circles represent the actual position and the triangles are the predictions one step in advance

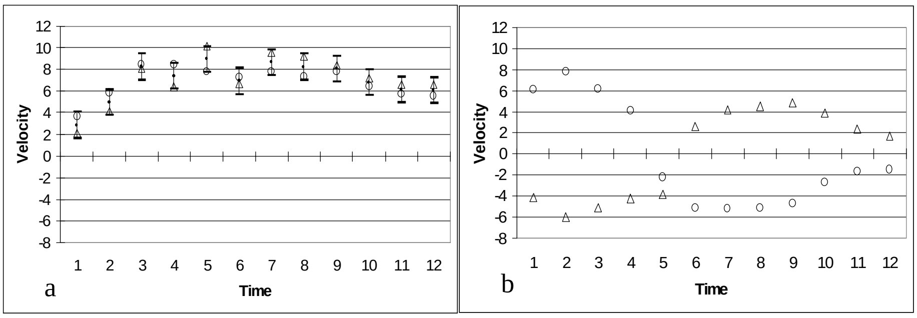

Fig. 4. Relative velocities of the horizontal sides of rectangles during occlusion in different movements. The small circles represent the top side and the triangles represent the bottom side

of the rectangle sides velocities during occlusion are presented in Fig. 6. The hand collision is detected in the 9th to 11th time slices in which all the velocities get very close to zero.

In order to measure the performance of the algorithm 3500 experiments were done with many different hand shapes. The results of the experiments are presented in Table 1. The other important parameter is the processing speed. With a fast camera working in 120 frames per second on a PII-1 GHz the algorithm is able to process 37.5 frames per second in average.

Although some other groups have done some work on hand tracking, none of them has considered the problem of tracking the hands during occlusion so widely. The CONDENSATION algorithm [20] is implemented by Gong et al. [7] works at a low speed able to process an image in 24 seconds on a PII-330 MHz. Gong et al. have also a Bayesian Network based technique for modelling the semantics of interactive behaviors able to process 5 frames per second on a PII-330 MHz [7]. Their result of 13% error are based on the number of images in the database and not on the number of events. In our experiments of Table 1 the tracking results are event-based in which each event (movement) may consist dozens of images.

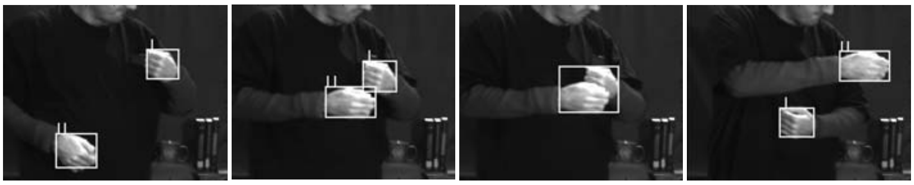

Fig. 5. A hand movement of type *h*. The hands are tracked correctly after occlusion

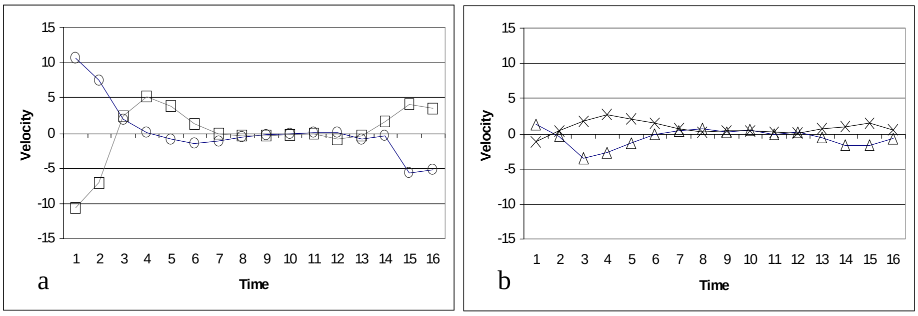

Fig. 6. Graphs of the velocities of the rectangle sides during occlusion. The velocities of the (*a*) vertical sides, (*b*) horizontal sides have drop near zero from 9th to 11th time slice

Table 1. Performance of the tracking algorithm using the dynamic model

Movement Type	# Events (movements)	# Errors	Error Rate (%)
a	400	25	6.25
b	600	106	17.67
c	400	65	16.25
d	500	29	5.8
e	400	41	10.25
f	400	44	11
g	400	17	4.25
h	400	24	6
Total:	3500	Weighted Average:	10.03

McAllister et al. [11] have, recently, introduced a model for hand tracking in smart desks and driving. In this model the camera should be in a top-view position in which the both hands and forearms are visible. Using optimization they fit a circle to the palm and a line to the forearm. For some of the hand shapes the algorithm cannot fit the circle.

However, the model presented in this paper doesn't have those restrictions and is independent of hand shape and camera position. Therefore, for different angle of views and the changing hand shapes during movements it works effectively.

References

1. Cipolla, R., Pentland, A.: Computer Vision for Human-Machine Interaction. Cambridge University Press (1998)
2. Lin, D.: Spatio-Temporal Hand Gesture Recognition Using Neural Networks. Proc. IEEE World Congress on Computational Intelligence (1998)
3. Starner, T., Pentland, A.: Visual Recognition of American Sign Language Using Hidden Markov Models. Proc. Int'l Workshop on Automatic Face and Gesture Recognition, Zurich, Switzerland (1995)
4. Wilson, A.D., Bobick, A.: Parametric Hidden Markov Models for Gesture Recognition. IEEE Trans. Patt. Anal. Mach. Intell., Vol. 21, No. 9 (1999)
5. Lee, H., Kim, J.: An HMM-Based Threshold Model Approach for Gesture Recognition. IEEE Trans. Patt. Anal. Mach. Intell., Vol. 21, No. 10 (1999)
6. Ng, C.W., Ranganath, S.: Gesture Recognition via Pose Classification. Proc. Int'l Conf. Pattern Recognition ICPR'00, Barcelona, Spain (2000)
7. Gong, S., Ng, J., Sherrah, J.: On the Semantics of Visual Behaviour, Structured Events and Trajectories of Human Action. Image and Vision Computing, Vol. 20 (2002)
8. Jojic, N., Turk, M., Huang, T.: Tracking Self-occluding Articulated Objects in Dense Disparity Maps. Proc. IEEE International Conf. Computer Vision ICCV'99, Kerkyra Greece (1999)
9. Dockstader, S., Tekalp, A.M.: Tracking Multiple Objects in the Presence of Articulated and Occluded Motion. Workshop on Human Motion HUMO'00, Austin Texas (2000)
10. Campbell, L., Bobick, A.: Recognition of Human Body Motion Using Phase Space Constraints. 5th International Conf. Computer Vision, Cambridge Massachusetts (1995)
11. McAllister, G., McKenna, S. J., Ricketts, I. W.: Hand Tracking for Behaviour Understanding. Image and Vision Computing, Vol. 20 (2002)
12. Jackson, G.M., Jackson, S.R., Husain, M., Harvey, M., Kramer, T., Dow, L.: The Coordination of Bimanual Prehension Movements in a Centrally Deafferented Patient. Brain, Vol. 123 (2000)
13. Diedrichsen, J., Hazeltine, E., Kennerley, S., Ivry, R.B.: Moving to Directly Cued Locations Abolishes Spatial Interference During Bimanual Actions. Psychological Science, Vol. 12, No. 6 (2001)
14. Mechsner, F.: Why Are We Particularly Good at Performing Symmetrical Movements. Max-Planck Research (2002)
15. Mechsner, F., Kerzel, D., Knoblich, G., Prinz, W.: Perceptual Basis of Bimanual Coordination. Nature, Vol. 414 (2001)
16. Kennerley, S., Diedrichsen, J., Hazeltine, E., Semjen, A., Ivry, R.B.: Callosotomy Patients Exhibit Temporal Uncoupling During Continuous Bimanual Movements. Nature Neuroscience. Online Publication (2002)
17. Pitas, I.: Digital Image Processing Algorithms, Prentice Hall (1993)
18. Chui, C.K., Chen, G.: Kalman Filtering With Real Time Applications. Springer-Verlag (1999)
19. Brown, R.G., Hwang, P.Y.C.: Introduction to Random Signals and Applied Kalman Filtering. John Wiley and Sons (1997)
20. Isard, M., Blake, A.: CONDENSATION - Conditional Density Propagation for Visual Tracking. Intl. J. Computer Vision, Vol. 29 (1998)