Cyclic Motion Detection

Ping-Sing Tsai and Mubarak Shah
Computer Science Department
University of Central Florida
Orlando, Florida 32816

Katharine Keiter and Takis Kasparis
Electrical and Computer Engineering
University of Central Florida
Orlando, Florida 32816

Abstract

The motion of a walking person is analyzed by examining cycles in the movement. Cycles are detected using autocorrelation and Fourier transform techniques of the smoothed spatio-temporal curvature function of trajectories created by specific points on the object as it performs cyclic motion. A large impulse in the Fourier magnitude plot indicates the frequency at which cycles are occurring. Both synthetically generated and real walking sequences are analyzed for cyclic motion. The real sequences are then used in a motion based recognition application in which one complete cycle is stored as a model, and a matching process is performed using one cycle of an input trajectory.

1 Introduction

A great deal of work has been done in the field of psychology to show that people can recognize objects from their trajectories [7, 11]. It has been theorized that humans can recognize an object based on the motion of several points on that object by inferring the three dimensional structure of the object from the transformations the two dimensional image undergoes. Cutting [3] gives examples of six different types of motion: *rolling wheels, walking people, swaying trees, aging faces, the rotating night sky* and *expanding flow fields.* Todd [11] is interested in distinguishing between rigid and several types of non-rigid motion such as bending, stretching, twisting and flowing. By displaying the trajectories of either rigid or non-rigid objects, Todd shows that human observers are able to distinguish between the two. Goddard [5] has proposed a computational model for visual motion recognition in the moving light displays. He believes that the visual system continuously computes invariants used to represent objects and movement. These invariants are used to index into 2D memory models. Having identified the most likely candidate, the viewpoint is computed and a verification stage operating in 3D confirms or denies the hypothesis. Another possible method would use motion information to reconstruct various static qualities, and use those static qualities to index into memory and recognize the object. However, Goddard has argued for a recognition process operating directly on motion information. Engel and Rubin [4] describe an implementation of an algorithm for detecting motion boundaries given discrete position input. Motion boundaries comprise starts, stops, pauses, and force impulses. Their algorithm represents image motion velocity in polar coordinates. Force impulses are asserted when the slope of zero-crossing of the second derivative of speed or direction exceeds a threshold.

In computer vision the work related to detection of motion before recognition has been reported. Allmen and Dyer [1] detect cyclic motion by tracking curvature extrema in spatio-temporal images. Repeating patterns are detected using a scale-space representation. In their approach, 3-D spatio-temporal volumes are formed by stacking a dense sequence of image frames, and when an edge operator is applied, this ST-volume contains surfaces and volumes which represent object motion swept out through time. ST-curves are detected on the ST-surfaces by connecting edge points into contours, and the curvature extrema are then found. The curvature extrema are used as tokens which are connected from one frame to the next, forming ST-curves. The ST-curves recover the cyclic behavior of the ST-surfaces. Repeating patterns in the ST-curves are then detected by matching the scale-space features of every curve. Both fine and coarse cyclic motion can be observed since curvature scale-space represents curvature over many scales.

In this paper, cyclic motion is detected in the motion paths of joint elements during human walking. Cyclic motion can be defined as the motion undertaken by an object that follows a repeating path over time. Examples include a person walking, running, skipping, riding a bike, a pendulum swinging, a ball bouncing, wings flapping, and a piston moving. An application of cyclic motion detection is the detection of gait problems in an injured person by comparing the path followed by specific points on the walking body of an injured person to the path created by the same points on a healthy person. Similarly, athletic performance can be improved when an expert examines the paths created by points on an athletes body during training. The detection of cycles is also useful in recognition problems, since specific types of motion may be recognized according to the cycles a moving object makes.

We use correlation and Fourier transform techniques to detect cycles in 2-D trajectories created by points on a moving object. We consider the trajectory as a spatio-temporal curve in (x,y,t) space. Cyclic motion is detected by finding cycles in the curvature of this spatio-temporal curve. This approach for detecting cycles is simpler than one that uses curvature scale space, because scale space approach essentially matches portions of scale space to find repeated patterns of curvature for periodicity, which is time consuming. Also, our approach can detect periodicity not evident in the spatial domain because of the presence of uncorrelated noise. It is also computationally efficient because the Fourier transform can be computed via the FFT (Fast Fourier Transform) algorithm. The detected cycles are then applied to a method proposed by Rangarajan et al [10] for matching pairs of single trajectories. Instead of storing all the trajectories with different cycles as models in order to find the correct match for an input trajectory, we store the trajectory with one complete cycle as our model, and do the matching with one cycle of the input trajectory.

Polana and Nelson [9] used similar techniques to judge the degree of periodicity; this differs with our main concern which is to extract one cycle from an input trajectory with unknown cycles, and subsequently uses it for matching. They considered an image sequence as a spatio-temporal solid with two spatial dimensions and one time dimension, and detected periodicity using the Fourier transform. They compute reference curve (which is essentially a trajectory) by tracking the centroid of moving region in several frames. They use reference curve to align the frames, and then compute gray level signals at every pixel in the image frame. The gray level signals are used to detect periodicity. The gray-level signals used by Polana and Nelson are different from the curvature signals, generated from the trajectories, used in our approach.

2 Detection of Cycles Using the Fourier Transform

A trajectory is defined as a sequence of points $((x_1, y_1), (x_2, y_2), (x_3, y_3), \ldots\ldots (x_t, y_t))$, ordered by an implicit time dimension. We can represent a 2-D trajectory as two 1-D trajectories, $x(t)$ and $y(t)$, or two 1-D time functions, namely speed and direction. Once coordinates of points that make up a trajectory are acquired, this 1-D information can be Fourier transformed to detect cycles. However, when a 2-D trajectory is represented by two 1-D signals, different frequencies from the two signals may be detected, and a problem is how to combine two different frequencies to get the correct frequency for the trajectory.

To avoid this problem we instead consider a trajectory as a spatio-temporal curve ($[x(1), y(1), 1]$, $[x(2), y(2), 2]$, $[x(3), y(3), 3]$, ... $[x(t), y(t), t]$). We compute the curvature of this curve which is a function of time by using a 1D version of the quadratic surface fitting procedure described by Besl and Jain [2]. The curvature, κ, is defined as follows:

$$\kappa = \frac{\sqrt{A^2 + B^2 + C^2}}{((x')^2 + (y')^2 + (t')^2)^{3/2}} \quad (1)$$

where

$$A = \begin{vmatrix} y' & t' \\ y'' & t'' \end{vmatrix}, \quad B = \begin{vmatrix} t' & x' \\ t'' & x'' \end{vmatrix}, \quad \text{and} \quad C = \begin{vmatrix} x' & y' \\ x'' & y'' \end{vmatrix}.$$

The notation $|\cdot|$ denotes the determinant. We use the discrete approximation to compute the derivatives, for example, $x'(t) = x(t) - x(t-1)$ and $x''(t) = x'(t) - x'(t-1)$. Since we assume Δt to be constant, t' will equal 1, and t'' will be 0.

A number of pre-processing steps can be used to improve the detection of cycles. The curvature function exhibits large and narrow impulses at points of sudden changes on the trajectory. These impulses contain large high frequency components that may interfere with the detection of cycles, and it would be beneficial to be suppressed. A median filter is particularly suitable filter for this task since it can suppress narrow impulses while preserving smoother regions of the curvature. In our work the first step is to suppress narrow impulses using a conditional median filter [8] which can better preserve the shape of the curvature function while suppressing the large and narrow impulses. This filter performs median filtering only on samples where the absolute value of the difference between the sample and the corresponding median exceeds a threshold. With this strategy smooth signal regions remain intact, but sufficiently narrow and large impulses are suppressed. The second pre-processing step is to remove the DC component of the curvature in order to avoid the zero frequency impulse. We subtract the average value of the curvature function from the original curvature function before we perform the Fourier transform. The third step is to compute the autocorrelation of the curvature. If the motion is cyclic there will be some self-similarity within the curvature function which becomes more evident in the autocorrelation function. Finally the Fourier transform of the autocorrelation is used to detect the presence of cycles and the period of the cyclic motion. A large impulse will occur on the frequency axis of the Fourier magnitude plot at the fundamental frequency of the cycles that are present. One cycle of the trajectory is extracted using the indicated location of the largest impulse. Smaller impulses may also be present (harmonics) at integer multiples of the fundamental.

3 Experiments

In our experiments we used the FFT algorithm to compute the Fourier transform. To achieve sufficient frequency resolution the data array to be transformed was padded with zeros to become of length 2048 samples and a 2048-point FFT was used. It should be noted that computation of the autocorrelation function prior to transformation is not necessary because by the Wiener-Khintchine theorem the Fourier transform of the autocorrelation of a signal is the same as it's energy spectral density (Fourier magnitude squared). However, in order to demonstrate the effect of each pre-processing step, the autocorrelation function was still computed.

3.1 Synthetic data

The first experiment was performed using synthetic data obtained from a program by Cutting [3], which generates files containing the coordinates of certain points on the body of a simulated walking person. Values are input to the program to determine factors such as hip swing and shoulder excursion, and the program uses laws of physics to determine the x and y coordinates of each point as the person walks. Feature points are at the following locations: ankle, wrist, elbow, knee (right and left), hip, shoulder (right), and head. For each cycle there are 40 instances at which coordinates are calculated, and the program outputs coordinates for twelve cycles, giving a total of 480 frames. Figure 1 shows the results for motion of the right ankle point. (We chose the trajectory of the right ankle point to demonstrate our method because it has obvious cycles.) The twelve cycles that were created by Cutting's program are shown in Figure 1.(a), which shows the trajectory created by the x and y coordinates of the right ankle point. The curvature function is shown in Figure 1.(b). The result of the autocorrelation is shown in Figure 1.(c), and the magnitude of the Fourier transform of the autocorrelation is shown in Figure 1.(d). We can clearly see that a large impulse occurs on the frequency axis of the Fourier magnitude plot.

Figure 2 demonstrates that the proposed method can detect and extract one cycle from a trajectory with unknown number of cycles. Figure 2.(a) and (c) are the curvature functions of Figure 1.(b) with different length (180 frames and 310 frames). The proposed method successfully detected and extracted the same cycle (as shown in Figure 2.(b) and (d)) for both cases.

3.2 Real data

The proposed method was also tested on a real walking sequence that was obtained from Goddard [6] at University of Rochester. A walking person sequence with 132 frames was tested. The trajectory of the shoulder point is shown in Figure 3.(a). The curvature function is shown in Figure 3.(b). The magnitude of the Fourier Transform of the autocorrelation function is shown in Figure 3.(c). A large impulse is clearly shown on the frequency axis of the Fourier magnitude plot. Figure 3.(d) shows the correspondent one cycle of the trajectory.

4 Application: Motion-Based Recognition

One important application for cyclic motion detection is in motion-based recognition. In many cases, where an object has a fixed and predefined motion, the trajectories of several points on the object may seem to uniquely identify the object. Therefore, it should be possible to recognize certain objects based on motion information obtained from the trajectories of representative points. Rangarajan *et al* [10] proposed a method for matching pairs of single trajectories utilizing a scale-space representation as the basis for matching. They represent a 2-D trajectory as two 1-D functions, namely speed and direction, and convolve the 1-D speed and direction signals with the second derivative of Gaussian over a range of σ values to produce the 2-D scale-space image. They then determine the strength and polarity by applying the first derivative of the Gaussian at each zero-crossing point in the scale-space image. The strength and polarity of each zero-crossing is referred to as the *zero-crossing potential.* Match scores between the two trajectories are determined by computing the difference between their smoothed zero-crossing potentials.

Rangarajan's method assumes that the correspondence between the model trajectory and the input trajectory is known. For an object with cyclic movement, they need to store all the trajectories with difference cycles as models in order to find the correct match for an input trajectory. Since we can detect cycles in an input trajectory (assuming the object has cyclic motion), we only need to store and do the matching with one complete cycle as our model. In order to minimize the computation and problems due to the noise sensitivity of the direction function, we use one cycle of the filtered curvature for the matching algorithm, instead of speed-direction as Rangarajan did. The modified matching algorithm is summarized as following:

1. Compute the curvature signal, $\kappa[t]$, from one complete cycle of the input trajectory(using equation (1)), and filter it through a conditional median filter.

2. Generate the curvature scale-space image by convolving the filtered curvature signal with the second derivative of the Gaussian over a range of σ values, and locate the curvature zero-crossings by scanning the scale-space image and testing the values in a neighborhood around each point.

3. Determine the strength and polarity by applying the first derivative of the Gaussian to the curvature $\kappa[t]$ (i.e. $\kappa[t] * \frac{-t}{\pi\sigma^2}e^{\frac{-t^2}{2\pi\sigma^2}}$) at each zero-crossing point in the scale-space image. The strength $(|\kappa[t] * \frac{-t}{\pi\sigma^2}e^{\frac{-t^2}{2\pi\sigma^2}}|)$ and polarity (sign) of each zero-crossing is referred to as the *zero-crossing potential.* This step produces a *2-D* array, $\beta_\kappa[t, \sigma]$, containing the zero-crossing potentials at each point. In this array, points which are not zero-crossings will hold a zero value.

4. Diffuse the zero-crossing potentials β_κ using a *2-D* Gaussian mask with sigma equal to one, and store the result in array γ_κ.

5. Scale the value in γ_κ by the scaling factors $\frac{\sum\sum \alpha_\kappa[t,\sigma]}{\sum\sum \gamma_\kappa[t,\sigma]}$, where $\alpha_\kappa[t,\sigma]$ is the diffused zero-crossing potentials for one complete cycle of the model trajectory.

6. Perform an element by element subtraction of the α and γ arrays, and store the result in array ϵ_κ.

7. Compute the match score as $1 - \frac{\sum\sum |\epsilon_\kappa(t,\sigma)|}{2 * |\sum\sum \alpha_\kappa(t,\sigma)|}$.

A perfect match between trajectories will produce a match score of 1.

To demonstrate this application, the walking sequences of persons K and W were used. We videotaped a person, K, walking at two different times, and generated two distinct image sequences, $K1$ and $K2$. We also videotaped another person, W, and generated a single image sequence, $W1$. There are 32 frames in each sequence. Figure 4 (a) through (f) show frames 1, 6, 12, 18, 24, and 30 of sequence $K1$. Figure 4.(g) shows the stick figure of the 9 body points which are generated manual. Figure 4.(h) is the trajectory of the *left heel* point, which is used as the model.

In order to get longer sequences with difference number of cycles, we generated the sequence K^2 with 128 frames by repeating the original $K2$ sequence four times, and the sequence W^1 with 256 frames by repeating the original $W1$ sequence eight times with random noise. We also added some random noise after the first cycle of each sequence to avoid the precise replication. Figure 5 (a) and (b) show the trajectory of the *left heel* point of sequences K^2 and W^1. The curvature of the trajectories are shown in Figure 5 (c) and (d). The proposed method extracted 32 frames as one cycle for both sequences.

The matching results for sequences $K1$ and $K2$ using one complete cycle are shown in Figure 6. Figure 6 (a) and (b) are zero-crossing potentials of the curvature scale-space of trajectories $K1$ and $K2$, and Figure 6 (c) and (d) are the diffused version of the zero-crossing potential. The difference picture between Figure 6 (c) and (d) is shown in Figure 6.(e). The match score between $K1$ and $K2$ is 0.836. (For a perfect matching the match score should be 1.)

The matching results for sequences $K1$ and $W1$ using one complete cycle are shown in Figure 7. Figure 7 (a) and (b) are zero-crossing potentials of the curvature scale-space of trajectories $K1$ and $W1$, and Figure 7 (c) and (d) are the diffused version of the zero-crossing potential. The difference picture between Figure 7 (c) and (d) is shown in Figure 7.(e). The match score between $K1$ and $W1$ is 0.137, which is low enough to declare a mismatch. It is clear that the cyclic motion detection is helpful in reducing the overhead of the motion-based recognition.

5 Conclusions

In this paper, we presented a method for cyclic motion detection using autocorrelation and Fourier Transform techniques. We represent a 2-D trajectory as a 1-D signal: curvature, which is a function of time. Cycles are detected successfully in the frequency domain by using the Fourier Transform of the pre-processed curvature signal of the trajectory. The proposed method was tested on synthetic data and real data of a walking person. We also demonstrated an application, motion-based recognition, for the cycle detection method. Although we use only the magnitude information of the Fourier transform for cyclic motion detection, we believe that the phase information is also valuable, especially for solving the correspondence problem (the phase shift problem) in the matching process.

References

[1] Allmen, M and Dyer, C. Cyclic motion detection using spatiotemporal surfaces and curves. In *Proc. 10th Int Conf on Pattern Recognition*, pages 365–370. IEEE Computer Society, June 1990.

[2] P.J. Besl and R.C. Jain. Invariant surface characteristics for 3d object recognition in range images. *CVGIP*, 33:33–80, 1986.

[3] Cutting, J.E. A program to generate synthetic walkers as dynamic point-light displays. *Behaviour Research Methods and Instrumentation*, 10(1):91–94, 1977.

[4] S.A. Engel and J.M. Rubin. Detecting visual motion boundaries. In *Workshop on Motion: Representation and Analysis (Charleston, SC, May 7–9, 1986)*, IEEE Publ. 86CH2322-6, pages 107–111, 1986.

[5] N.H. Goddard. The interpretation of visual motion: Recognizing moving light displays. In *Proceedings, Workshop on Visual Motion, (Irvine, CA, March 20–22)*, pages 212–220, 1989.

[6] Goddard, N. *The perception of articulated motion: Recognizing Moving Light Displays*. PhD thesis, University of Rochester, 1992.

[7] Johansson, G. Visual perception of biological motion. *Scientific American*, 232:76–89, June, 1975.

[8] T. Kasparis, N. Tzannes, and Q. Chen. Detail preserving adaptive median filters. *Journal of Electronic Imaging,*, 1(4):358–364, 1992.

[9] R. Polana and R. Nelson. Detecting activities. In *Proc. of Computer Vision and Pattern Recognition*, pages 2–7, June 1993.

[10] Rangarajan, K., Allen, Bill, and Shah, M. . Matching motion trajectories. *Pattern Recognition*, 26:595–610, July, 1993.

[11] Todd, J.T. Visual information about rigid and nonrigid motion: A geometric analysis. *J. Experimental Psychology: Human Perception and Performance*, 8:238–252, 1982.

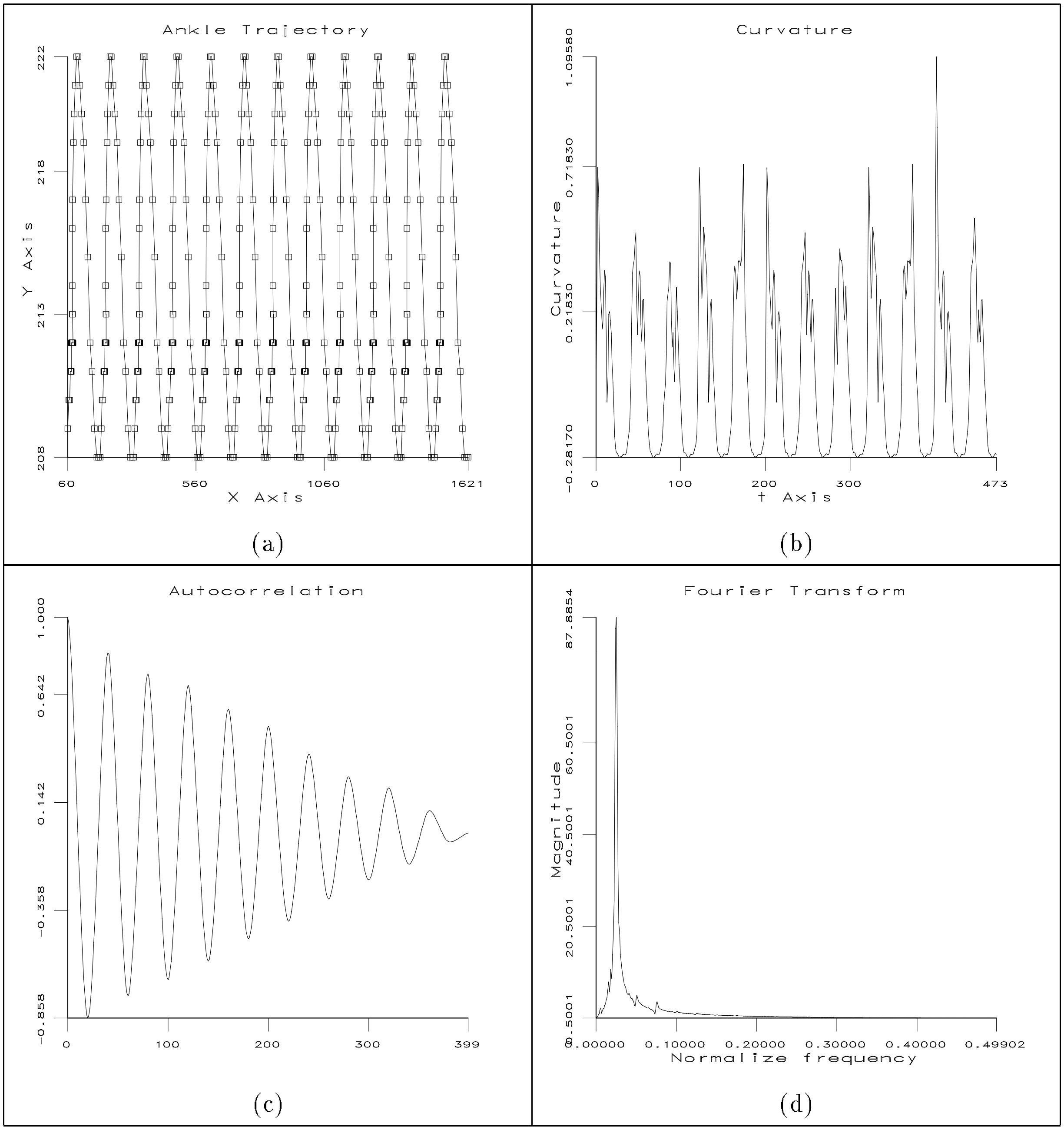


Figure 1: Results for the synthetic walking person trajectory (obtained from a program by Cutting). (a). The trajectory of the right ankle point. (480 frames). (b). The curvature function of (a). (c). The autocorrelation function of (b). (d). The magnitude of the Fourier Transform of the autocorrelation function.

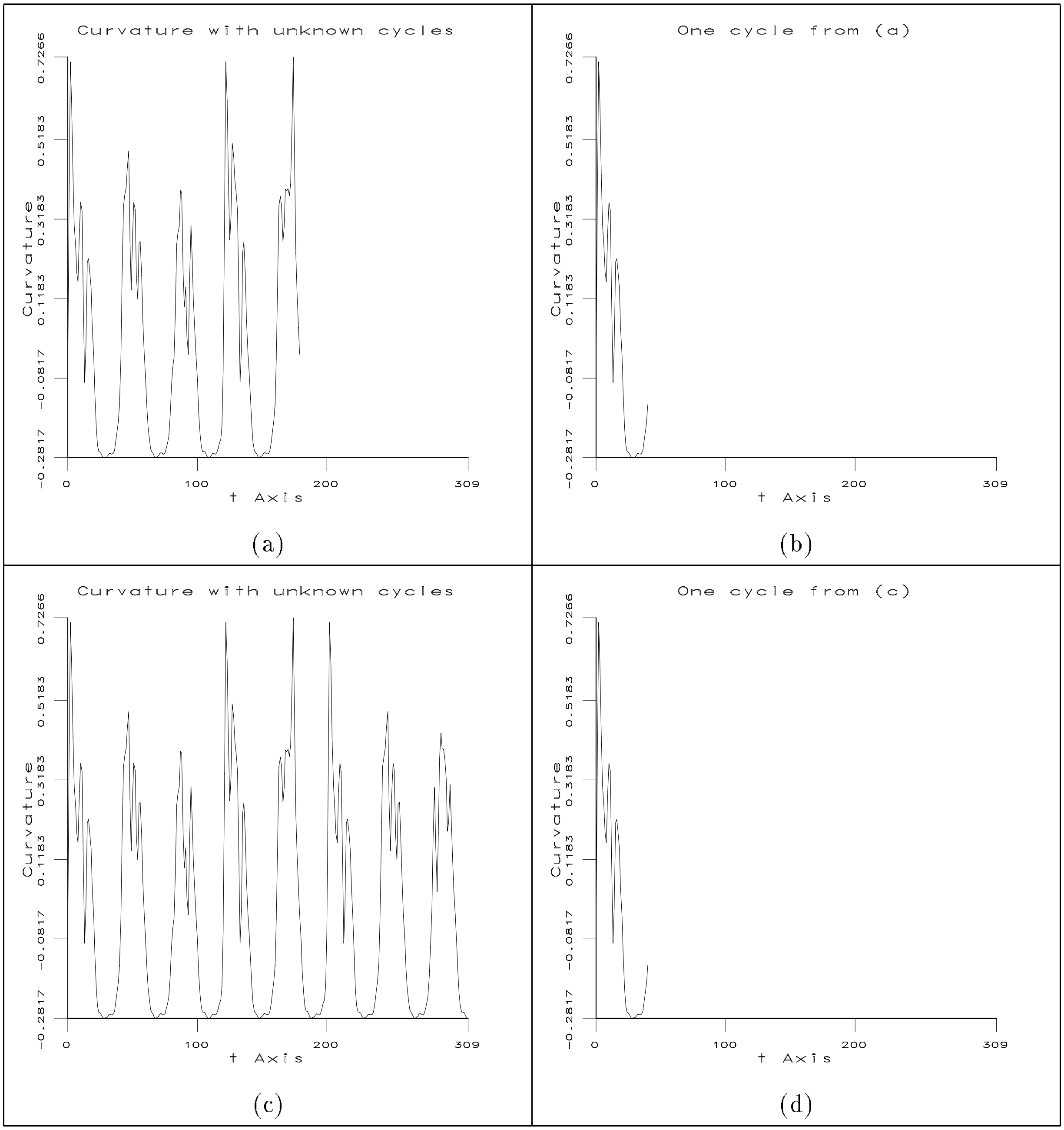


Figure 2: Results for cycle detection and extraction. (a). The curvature function of Figure 1.(b) with unknown cycles. (180 frames) (b). One cycle extracted from (a) using the proposed method. (c). The curvature function of Figure 1.(b) with unknown cycles. (310 frames) (d). One cycle extracted from (c) using the proposed method.

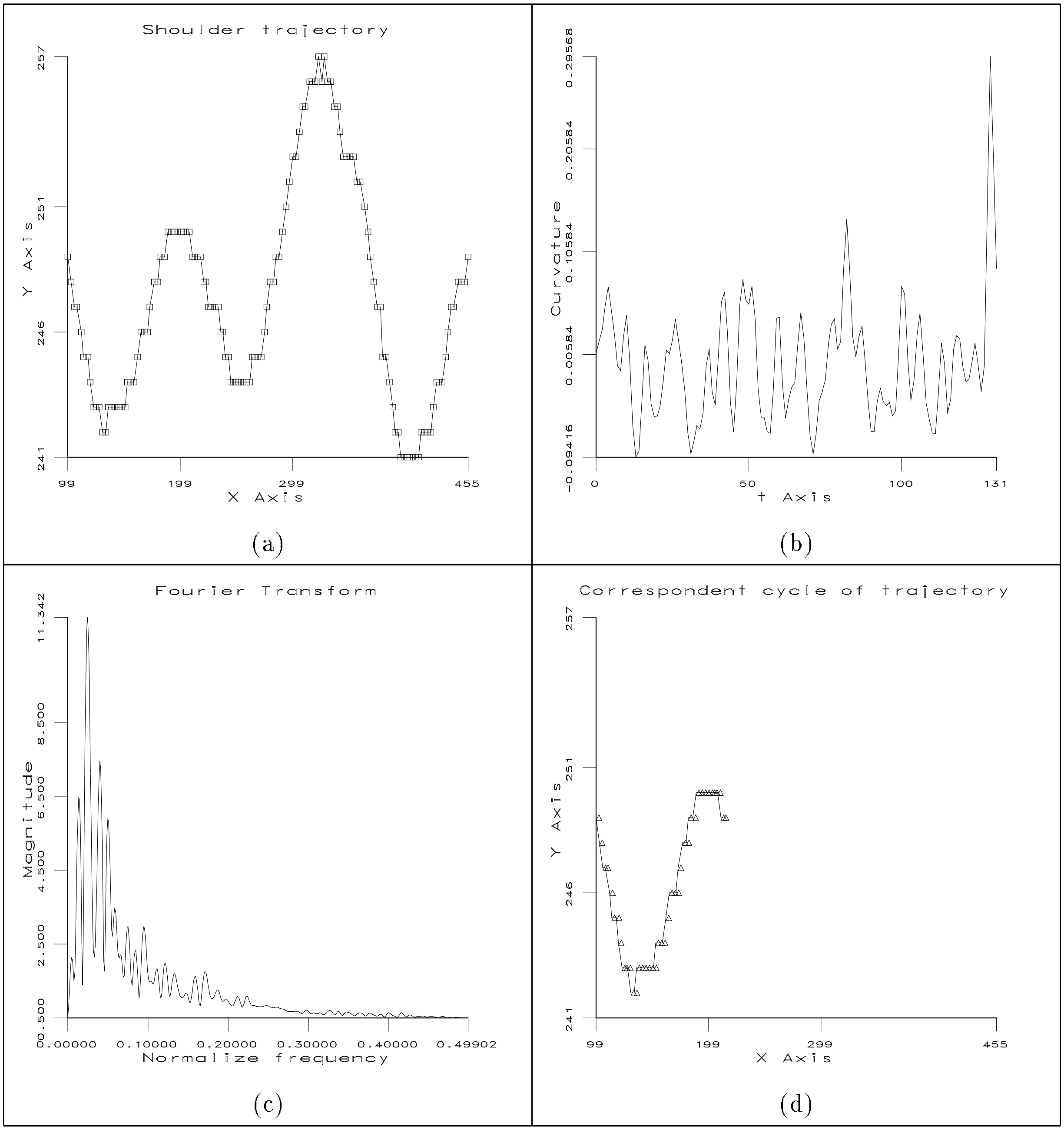


Figure 3: Results for the real walking person. (a). The trajectory of the shoulder point (132 frames). (b). The curvature function of (a). (c). The magnitude of the Fourier Transform of the autocorrelation function. (d). The correspondent one cycle of the trajectory.

(a)

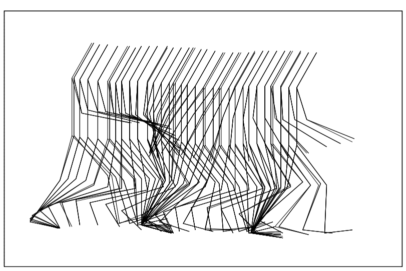
(b)

(c)

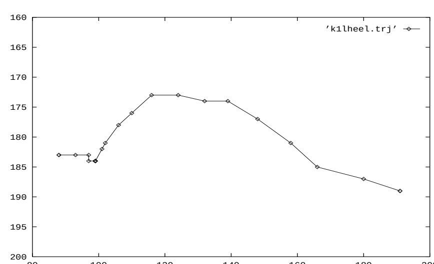
(d)

(e)

(f)



(g)



(h)

Figure 4: Image Sequence $K1$ in which person K is walking. There are 32 frames in this sequence. (a)-(f) Frames 1, 6, 12, 18, 24 and 30 of the sequence K^1, (g) The stick figure drawings of 9 body points. (h) Trajectory of K^1_{lheel}.

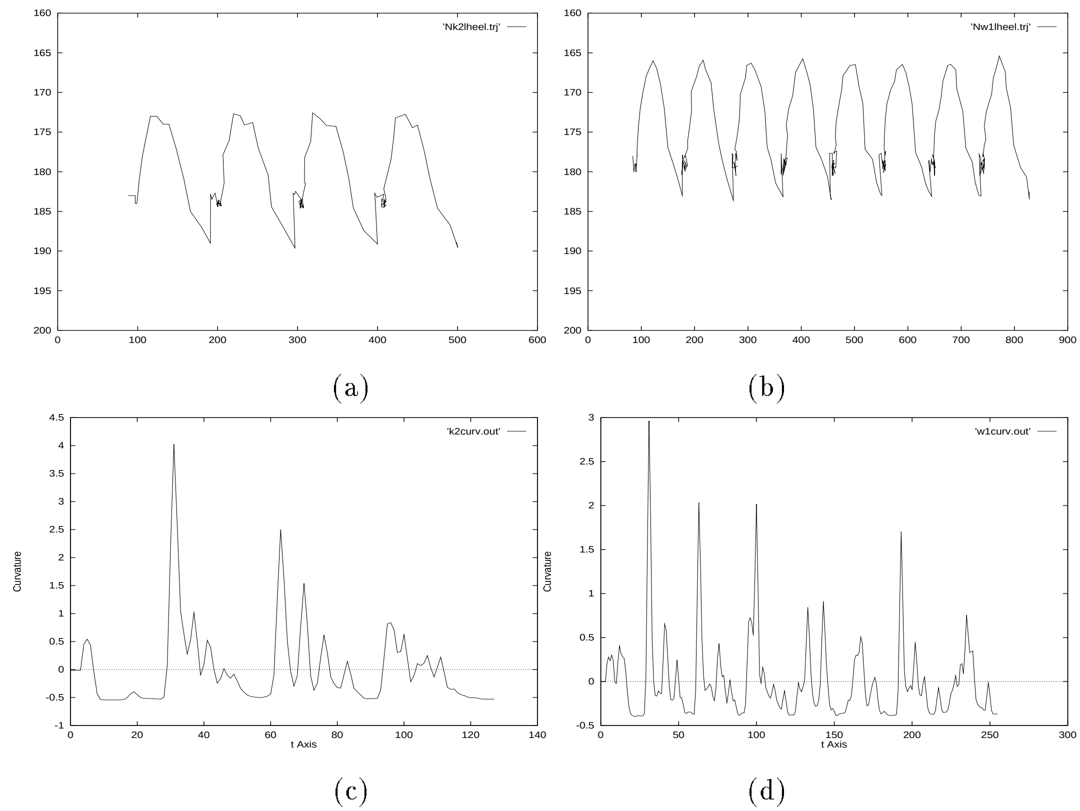
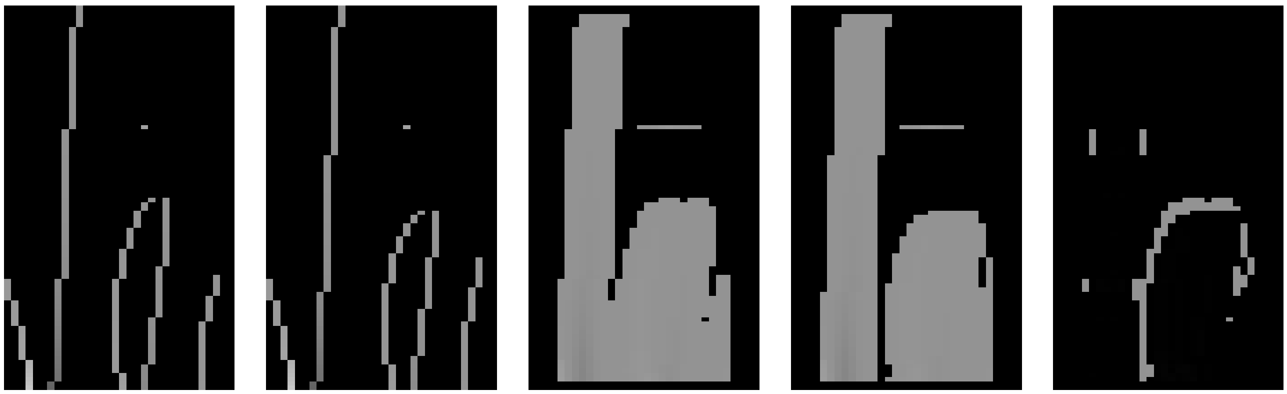
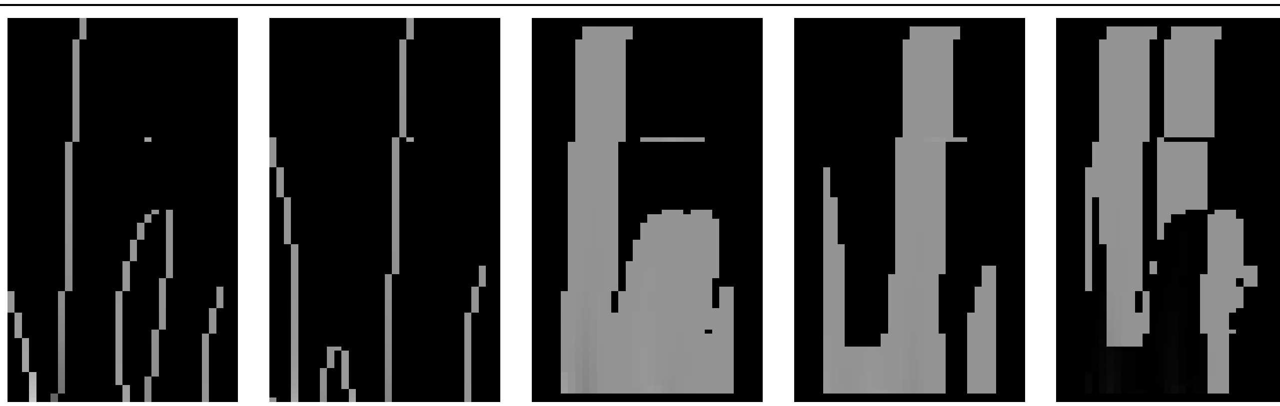


Figure 5: (a) Trajectory of K^2_{lheel}. There are 128 frames which are obtained by repeating the original $K2$ sequence four times. (b) Trajectory of W^1_{lheel}. There are 256 frames which are obtained by repeating the original $W1$ sequence eight times. (c) The curvature function of the sequence K^2. (d) The curvature function of the sequence W^1. The proposed method extracted 32 frames as one cycle for both sequences.



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

Figure 6: The matching results for K1 and K2. (a) The zero-crossing potential of the curvature scale-space of trajectory K1. (b) The zero-crossing potential of the curvature scale-space of trajectory K2. (c) The diffused version of (a). (d) The diffused version of (b). (e) The difference picture between (c) and (d). The match score between K1 and K2 is 0.836.



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

Figure 7: The matching results for K1 and W1. (a) The zero-crossing potential of the curvature scale-space of trajectory K1. (b) The zero-crossing potential of the curvature scale-space of trajectory W1. (c) The diffused version of (a). (d) The diffused version of (b). (e) The difference picture between (c) and (d). The match score between K1 and W1 is 0.137.