PDA BASED HUMAN MOTION RECOGNITION SYSTEM

K. Leman

Institute for Infocomm Research, 21 Heng Mui Keng Terrace, Singapore 119613
karianto@i2r.a-star.edu.sg

G. Ankit[†]

National University of Singapore, 10 Kent Ridge Crescent, Singapore 119260
eng01278@nus.edu.sg

T. Tan

Curtin University of Technology, GPO Box U1987, Perth, Western Australia 6845
teletan@cs.curtin.edu.au

This paper describes the design and implementation of autonomous real-time motion recognition on a Personal Digital Assistant. All previous such applications have been non real-time and required user interaction. The motivation to use a PDA is to test the viability of performing complex video processing on an embedded platform. The application was constructed using a representation and recognition technique for identifying patterns using Hu Moments. The approach is based upon temporal templates (Motion Energy and History Images) and their matching in time. The implementation was done using Intel Integrated Performance Primitives functions in order to reduce the complexity of the application. Tests were conducted using 5 different motion actions like arm waving, walking from left and right of the camera, head tilting and bending forward. Suggestions were also made on how to improve the performance of the system and possible applications.

Keywords: Hu Moments, motion recognition, personal digital assistants, temporal templates.

1. INTRODUCTION

Today's palm-sized Personal Digital Assistant (PDA) is increasingly packed with computing power, memory and accessories. PDA can now perform relatively computation intensive task such as hand-writing and voice recognitions. We can also plug in a camera card (some have built-in camera), extra memory storage, modem, etc. These have opened up the possibility to explore innovative applications such as human gesture and activity recognition. Such applications when viable on a PDA platform can be ported into embedded systems transforming a normal camera into a Smart Camera. Simple tasks such as intruders (and their direction) detections and gesture recognitions can be widely explored in real-live deployment as there are no hindrances of expensive

[†] Garg Ankit graduated in July 2004 from the Electrical and Computer Engineering Department of National University of Singapore. He is now working with the Standard Chartered Bank, Singapore

setup and video transmission. Currently the most powerful CPU for the Palm OS based system is the 200 MHz ARM Processor while the most powerful CPU for Windows® CE / Pocket PC based system is the 400 MHz Intel® Processor. For a variety of reasons, all PDAs are configured with limited memory. A typical Palm OS based system is with 8-16 MB memory and a Windows CE / Pocket PC based system is with 16-64 MB memory. This space is for storage and application programs. The memory size is crucial for a computer vision application because some algorithms require a huge space.

This paper discusses the feasibility of development of Computer Vision algorithms like motion based recognition on PDAs. Research was conducted on past computer vision applications produced on Personal Digital Assistants and two such applications were found, a PDA based Face Recognition System [4] and a Translation of Foreign Road Signs Application [6]. Both these systems were built on non real-time data and involved user interaction. Therefore, it was considered an appropriate challenge to develop a real-time, autonomous computer vision application like motion recognition on a PDA.

2. Algorithm Surveys

Tracking/Recognition techniques are based on matching tokens from the image. They are extracted along the sequence and are used as observations for the tracking algorithm. There are several trends and tracking strategies. Most of these tracking methods can be divided into four groups:

(i) Three dimensional-based methods. They use precise geometrical representations of known objects. These methods present considerable computational loads that hinder a real-time system such as that needed in a traffic monitoring system [2].

(ii) Deformable model-based methods. These methods fit models to the contours of the moving objects in the scene. They exhibit initialization problems. When moving objects are partially occluded in the scene, initialization would fail since models can not be adapted to the real objects.

(iii) Feature-based methods. These track individual tokens such as points, lines or curves. These methods present two main disadvantages: they do not provide explicit grouping of tokens moving with coherent motion and are quite sensitive to occlusion. The most popular feature tracking algorithm is the Lucas-Kanade Feature Tracking algorithm [10].

(iv) Region-based methods. These methods define groups of connected pixels that are detected as belonging to a single object that is moving with a different motion from its neighbouring regions. Region tracking is less sensitive to occlusion due to the extensive information that regions supply. Characteristics such as size, shape or intensity can be directly obtained from them.

As seen from the characteristics of the various recognition methods described, the three-dimensional based and deformable model based methods are excluded from the selection process, due to intensive computational requirements and initialization problems respectively. This leaves us with two main methods of recognition i.e. region-based and feature-based.

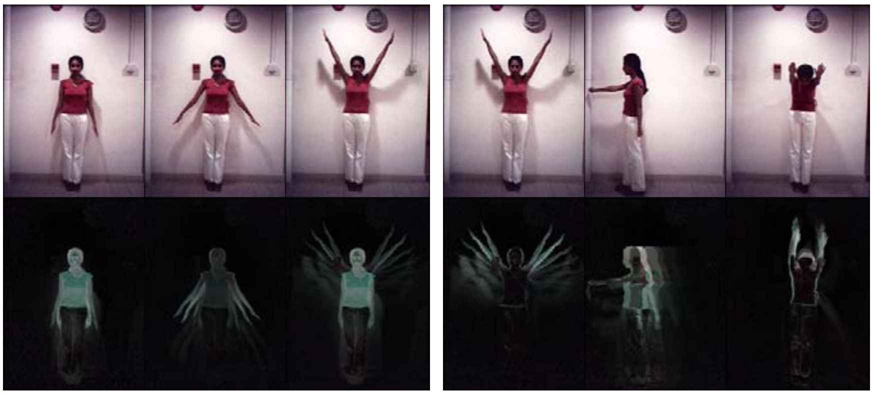
Amongst these two methods, region-based algorithm seems to be more efficient with fewer disadvantages; this is because of the extensive information that is provided from the region (or the image). This is particularly useful for PDA applications, as the video resolution is relatively small and the quality of video feed is poor. The feature tracking algorithm based on point features may not be as robust to implement on a PDA as the point features might get lost due to poor quality of video and lower resolution. Therefore, a region based recognition algorithm was found to be the most appropriate for the development of a motion recognition application on a PDA.

3. Algorithm Development

3.1. *Temporal Templates*

The goal is to construct a view-specific representation of an action, where action is defined as motion over time. It is assumed that the background is static, a snapshot of the background is captured in a normal room lighting condition. It is used as a reference to segment out objects that come into the scene through simple subtraction. Method on the automatic generation of static background can be done using statistical approach such as that in [11]. The test data are also generated with actor's hands clearly visible. This is obtained with the PDA at 4 meters away when using a camera with 350 kilo pixel resolutions.

Motion Energy Images: Consider the example of someone waving his/ her arms, as shown in Figure 1(a). The top row contains key frames in the sequence. The bottom row displays cumulative RGB motion images – to be described momentarily – computed from the start frame to the corresponding frame above. As expected, the sequence sweeps out a particular region of the image. It can be claimed that the shape of that region (where there is motion) can be used to suggest both the action occurring and the viewing condition (angle). In order for simplicity, it is assumed that the viewing angle is 90 degrees to the motion, i.e. right in front of the camera.



(a) (b)

Fig. 1. (a) Actual images and corresponding Motion Energy Images, (b) Actual images and Motion History Images of different actions.

These RGB cumulative motion images can be referred as Motion-Energy Images (MEI). Let I(*x, y, t*) be an image sequence, and let D(*x, y, t*) be a RGB image sequence indicating regions of motion; this can be obtained by performing image differencing (i.e. subtraction of one image from the next). Then the Motion-Energy Image E (x, y, t) is defined as:

$$E(x,y,t)=\sum D(x,y,t) \tag{1}$$

Motion History Images: To represent how the motion image is moving, we form a motion-history image (MHI). In an MHI H, pixel intensity is a function of the temporal history of motion at that point. [3] The MHI will be able to indicate only the motion parts of the object but not the object as a whole. The result is a scalar-valued image where more recently moving pixels are brighter. Examples of MHIs are presented in Figure 1(b).

3.2. *Action Discrimination*

Training examples of different actions were collected from a 90 degree viewing angle. Given a set of an MEI and MHI for each action, statistical descriptions of these images using moment-based features were computed. The choices are the 7 Hu Moments [5] which are known to yield reasonable shape discrimination in a translation- and scale-invariant manner. For each action, a statistical model of the moments was generated for both the MEI and MHI. A range (mostly distinct) of these moments were computed for each model action using a sample size and to recognize an input action, the moment values for the MEI and MHI of input action was compared with moment ranges of stored actions. If the moment (of the input action) falls in a particular range of the model action moments, then the input action is recognized as the model action. In case of overlapping model action ranges, the distance between the moments of the input action and the means of the two overlapping ranges are computed and the input action is recognized as the model action whose distance is the smallest from the input action.

The approach outlined has the advantage of not being computationally taxing making real-time implementation feasible; one disadvantage is that the Hu moments are difficult to reason about intuitively. Also, it is noted that the matching methods for MEI and MHI need not be the same; in fact, given the distinction made between where there is motion from how the motion is moving, one might expect different matching criteria.

4. Implementation and Experimental Results

The implementation of the motion recognition application was done on IPAQ H3970 Pocket PC (400 MHz PXA250 XScale^{TM} Microprocessor [9], 64 MB) using Microsoft Embedded Visual C++ 3.0 platform. The camera used for the application was LifeView FlyCAM-CF (350 kilo pixels) using the CF-I slot [8]. The motion recognition application was constructed with the Intel Integrated Performance Primitives [7] for Personal Client Architecture functions and libraries on Image Processing and normal C libraries and math functions.

Different model actions were taken into account in order to collect a sample database. The model actions were Arms Wave, Head Tilt, Bend Forward, Walk from Left and Walk from Right of camera. All the data was collected from an angle of 90 degrees from the motion and it was assumed that no occlusion of motion will occur. In all, there were 10 samples collected for each model action in order to compute the moment ranges and its mean. The moment ranges were taken as 3 standard deviations apart from the mean in order to recognize 89% of the observations (Chebyshev's Inequality). For each model action, from the data collected from the Hu moments, 1 or 2 distinct Hu moments were manually chosen for the recognition of input actions. These Hu moments were mainly ϕ_1 and ϕ_2 as the values of the remaining moments were extremely high and highly dispersed, making them less robust for recognition of input actions. The Hu Moments were chosen in such manner that the ranges should overlap as little as possible. The mean values of the Moments for MEI and MHI from the sample data collected for different model motion actions are given in Table 1. (Standard Deviation values are given in the parenthesis) The values of the ranges specified for the moments of the model motion actions are also specified in Table 1. A measure of efficiency of the motion recognition system using 20 test samples for each model action is also specified in Table 1.

A test was also done for the motion recognition of a different person, for the Walking motion from the right of the camera motion action. These tests were conducted 5

Table 1. Moment Values and Ranges for different Model Actions

		ϕ_1	ϕ_2	ϕ_3	Moment Ranges	Eff.
Arms Wave	MEI	***1142*** ***(54)***	-3105 (207)	-76165 (5965)	ϕ_1 (MEI) $980 \leq x \leq 1306$	70%
	MHI	1535 (36)	*2863* *(373)*	200202 (16951)	ϕ_2 (MHI) $1742 \leq y \leq 3982$	
Head Tilt	MEI	1115 (70)	***-2983*** ***(215)***	-20634 (6639)	ϕ_2 (MEI) $-3628 \leq x \leq 2338$	60%
	MHI	1474 (33)	3739 (179)	14691 (6252)		
Bend Forward	MEI	1214 (80)	-3863 (261)	-17361 (4371)	ϕ_1 (MHI) $1542 \leq x \leq 1788$	60%
	MHI	*1665* *(41)*	*-4246* *(68)*	26885 (5243)	ϕ_2 (MHI) $-4450 \leq y \leq -042$	
Walk (Right)	MEI	*1469* *(27)*	*-213* *(318)*	53961 (16572)	ϕ_1 (MEI) $1388 \leq y \leq 1550$	75%
	MHI	1232 (38)	-2488 (303)	-65859 (14084)	ϕ_2 (MEI) $-850 \leq x \leq 423$	
Walk (Left)	MEI	1472 (67)	-3463 (319)	-47512 (14639)	ϕ_1 (MHI) $1042 \leq x \leq 1402$	80%
	MHI	*1222* *(60)*	*-4035* *(287)*	-8030 (8201)	ϕ_2 (MHI) $-896 \leq y \leq 3174$	

times, and the system was able to detect the motion 4 times, with an efficiency of 80%. Thus, the test results obtained from the Motion Recognition application were fairly successful with the efficiency of recognition of all motions over 60%. Also, small tests indicated that the motion recognition application was not dependent on one person, but could also detect other persons.

5. Summary

In this paper, a real-time autonomous motion recognition application was constructed using a representation and recognition technique for identifying patterns using Hu Moments. The implementation of the application was made in real-time scenario, with utilization of Intel IPP functions in order to reduce the computational complexity of the application. Tests were conducted using 5 different motion actions like Arms Wave, Walking from left and right of the camera, Head Tilt and Bending Forward. The application was efficient more than 60 % of the times for these actions.

A variety of applications can be constructed using the Motion Recognition Application developed on the PDA. System that recognizes sign languages can be constructed for the benefit of the hearing impaired. In the area of Homeland Security, surveillance cameras could be made smarter with the similar embedded module of motion analysis. Only upon detection of specific motions or actions from the human object that the system would send the video streams for human verifications. In fact, with such embedded intelligences in the camera, many of today’s surveillance and monitoring applications can achieve much wider and complete coverage while maintaining similar requirement of the space of monitoring room and human resources.

6. References

1. Cedras, C., and Shah. M. Motion based recognition: A survey, image and vision computing, 13(2):129-155, March 1995.
2. J. Badenas, J.M. Sanchiz and F. Pla. Using temporal integration for tracking regions in traffic monitoring sequences, IEEE International Conference on Pattern Recognition, 2000.
3. James Davis, Gary Bradski. Real-time motion template gradients using Intel CVLib
4. Jie Yang, Xilin Chen, William Kunz. A PDA-based face recognition system, Proceedings of WACV, 2002.
5. M. Hu. Visual pattern recognition by moment invariants. IRE Trans. Information Theory, IT-8(2), 1962.
6. Vinu Pattery. Translation of foreign road signs using a personal digital assistant
7. Reference Manual. Intel Integrated Performance Primitives for PCA processors
8. API Manual. LifeView FlyCAM-CF camera.
9. White Paper. The Intel PXA250 application processor.
10. B.D. Lucas, T. Kanade. An iterative image registration technique with an application to stereo vision. Proceeding of the 7th International Joint Conference of Artificial Intelligence 1981.
11. Stauffer, C, Grimson, W. Adaptive background mixture models for real-time tracking. In Computer vision and Pattern Recognition, 1999.