Real-time hand gesture recognition system based on Q6455 DSP board

Qiong Fei, Xiaoqiong Li, Tao Wang, Xiongkui Zhang, Guoman Liu
Dept. of Electronic Engineering of Beijing Institute of Technology, Beijing 100081,P.R.China

Abstract

This paper presents detailed description of a real-time hand gesture recognition system using embedded DSP board and image processing approaches. Such a system which can identify hand postures and dynamic gestures has manifold potential applications range from sign languages to human computer interaction. We use Q6455 DSP board based on 4 TI-TMS320C6455 DSPs as the computational unit. This versatile signal processing module has powerful computing and interconnection capability. To recognize hand postures, algorithm based on skin color segmentation and geometric invariant feature has been used. During identify dynamic gestures, optical flow tracking approach and direction encoding are adopted. The implementation of high reliable algorithm on DSP board keeps the system robust and efficient. Experimental results show that the proposed system performs well in recognizing hand postures and dynamic gestures real timely. The accuracy and scalability of this system are also soundly proved.

Key words: *DSP, hand gesture recognition, invariant moments, optical flow*

1. Introduction

Human-Computer Interaction is assuming as one of the most important techniques in the Artificial Intelligence nowadays. However, traditional HCI is based on keyboards and mice which inherently limit the speed and naturalness of human's interaction with computers. One long-term attempt in HCI has been to migrate the "natural" means that humans employ to communicate with each other into HCI[1][6]. Speech recognition and interface are successful accomplishments in this direction, at the same time, adding non-verbal interactions into HCI became increasingly significant. Hand gesture recognition can be considered as a promising approach to realize this thought.

Generally, hand gestures can be classified into hand postures and dynamic gestures [2]. The first one focuses on hand shape and position, while the second one intends to convey the meaning of hand movements of people. In the present day, there are different tools for gesture recognition, based on the approaches ranging from statistical modeling, computer vision and pattern recognition, image processing, etc. [3] Statistical modeling, such as discrete hidden Markov models (HMMs) has been used by Yamato et al [4] for recognizing image sequences of six different tennis strokes from different subjects. Mesh features were first vector-quantized and then used as input for the HMMs. However, the cost of extracting all mesh features would be too high. Image-processing techniques [5] such as analysis and detection of shape, texture, color, motion, optical flow, image enhancement, segmentation, have been widely used in gesture recognition. However, the computational intensive image processing approach is difficult to work efficiently only on software platform. The recognition system will be more applicable and flexible if it responses to the video input instantaneously, in other words, real timely. Such system can be realized on embedded platform, especially on DSP board with high computing capability.

This paper presents a hand gesture recognition system based on Q6455 DSP board which consists of 4 TMS320C6455 DSPs as the main computing processors. The image processing is implemented on Q6455 and makes full uses of the resources of the DSP board. Adopting the monocular computer vision, the system includes two modes: static hand posture recognition and identification of dynamic gestures. The image processing algorithm for static mode is based on geometric invariant feature while the dynamic mode employs optical flow tracking and direction encoding to ensure the tracking process to be accurate and stable. This system is highly robust and has strong capability of error correcting. Furthermore, the recognition rules works closely with the actual writing norms which enable the end users to master easily.

The reminder of this paper is organized as follows: Section 2 introduces the image processing algorithm implemented in the embedded platform;

978-0-7695-3571-5/09 $25.00 © 2009 IEEE
DOI 10.1109/GCIS.2009.97

Section 3 describes the hardware platform of hand gesture recognition system; Section 4 presents the embedded DSP software structure; Experiment results are discussed in Section 5 and finally conclusions are made in Section 6.

2. Image Processing Algorithm

We employ skin color segmentation and invariant feature extraction which have been widely used in gesture recognition. Consider dynamic gesture recognition, we choose optical flow method and direction encoding. This approach is proved to be effective and reliable in motion estimation but computing intensive.

2.1. Static hand posture recognition

Static hand posture recognition mainly includes three components: Skin color segmentation, geometric invariant feature extraction and standard feature matching.

2.1.1. Skin color segmentation. The first step is to segment the hand gesture from the background based on pixel color. The distribution of skin colors clusters in a small region of the chromatic color space. Several color spaces have been proposed in the literature for skin detection applications. YCbCr has been widely used since the skin pixels form a compact cluster in the Cb-Cr plane.[7] A pixel will be labeled as skin-like if its chrominance vector falls into the region of Cb[i]<125&& Cr[i]>130. And its luminance value is within the interval 235>Y[i]>45. These values were chosen empirically to reduce the miss detection rate since it is impossible to reduce the false alarm rate produced by skin-like colored background objects.[19]

2.1.2. Geometric invariant feature extraction. Invariant geometric features describe properties being invariant under image translation, rotation, and scaling.[9] We apply two-dimensional moment invariants for gesture classification. The recognition schemes based on these invariants could be truly position, size and orientation independent, and also flexible enough to learn almost any set of patterns.[10] For digital image f(x,y), the moments of an area in the image is calculated from the points within this area. They are robust to noise. The p+q moment for f(x,y) is mathematically defined as: [10]

$$m_{pq} = \sum_x \sum_y x^p y^q f(x,y) \qquad (1)$$

The p+q central moment for f(x,y) is mathematically defined as:

$$M_{pq} = \sum\sum (x-\bar{x})^p (y-\bar{y})^q f(x,y) \qquad (2)$$

where $\bar{x}$ and $\bar{y}$ are the core coordinate of the binary image. The normalized central moment of f(x,y) is:

$$N_{pq} = \frac{M_{pq}}{M_{00}^r} \qquad (3)$$

where $r = {}^{(p+q+2)}\!/_{2}$, $p+q = 2, 3, 4.....$

After training, based on the classification of the on-going three patterns, “Rock, Scissors, Paper”, we select the following features as the components for the feature vector.

$$\begin{aligned}
\phi_1 &= N_{20} + N_{02} \\
\phi_2 &= (N_{20} - N_{02})^2 + 4N_{11}^2 \\
\phi_3 &= (N_{30} - 3N_{12})^2 + (3N_{21} - N_{03})^2 \\
\phi_4 &= (N_{30} + N_{12})^2 + (N_{21} + N_{03})^2
\end{aligned} \qquad (4)$$

Thus, the feature vector is:

$$L = \{\phi_1, \phi_2, \phi_3, \phi_4\} \qquad (5)$$

2.1.3. Standard feature matching. We compare the invariant feature vector L_{unknown} obtained from the image that is to be recognized with the standard pattern L_{standard}. The least square error method is used to determine which pattern is the best matching.

2.2. Dynamic hand gesture recognition

Dynamic hand gesture recognition contains two components: (i) optical flow based object tracking (ii) direction encoding.

In order to track and segment the target object, we configure the tracking start-point and end-point for the target object. The tracking is triggered and ended based on a set of procedures. Optical flow based object tracking is reliable but computation-consuming [11][12][15][16][18]. The computational complexity can be significantly reduced by configuring a fixed trigger point of tracking. This also reduces the interference of moving objects (not the target object) in the background.

Specifically, we use Lucas-Kanade’s method[13][14][17] for tracking since it has been proved as a very accurate and reliable optical flow estimator algorithm.[13]-[15] Assume I(x,y) is the luminance of point (x,y) at time t, and u(x,y), v(x,y) are x and y components of the optical flow, respectively. Following Lucas-Kanade’s method, we assume that the motion vectors in a small region Ω are constant and

then estimate the optical flow by using weighted least-squares.

In a small spatial region Ω, the estimation error of optical flow is defined as:

$$\sum_{(x,y)\in\Omega} W^2(\mathbf{x})(I_x u + I_y v + I_t)^2 \qquad (6)$$

where W(x) denotes a window function (it gives more influence to constraints at the centre of the window).

Assuming $\mathbf{v} = (u, v)^T$, $\nabla\mathbf{I}(\mathbf{x}) = (I_x, I_y)^T$, then (6) is:

$$\mathbf{A}^T\mathbf{W}^2\mathbf{A}\mathbf{v} = \mathbf{A}^T\mathbf{W}^2\mathbf{b} \qquad (7)$$

where the n points at time t are in region Ω, that is $\mathbf{x}_i \in \Omega$,

$$\mathbf{A} = [\nabla\mathbf{I}(\mathbf{x}_1), \ldots, \nabla\mathbf{I}(\mathbf{x}_n)]^T,$$
$$\mathbf{W} = diag[W(\mathbf{x}_1), \ldots, W(\mathbf{x}_n)], \qquad (8)$$
$$\mathbf{b} = -(I_t(\mathbf{x}_1), \ldots, I_t(\mathbf{x}_n))^T.$$

Solution of (7) is $\mathbf{v} = [\mathbf{A}^T\mathbf{W}^2\mathbf{A}]^{-1}\mathbf{A}^T\mathbf{W}^2\mathbf{b}$. When $\mathbf{A}^T\mathbf{W}^2\mathbf{A}$ is a nonsingular matrix, we can analytically obtain solution $\mathbf{v}$. In the 2×2 matrix, all four summations can be obtained from the points in region Ω.

$$\mathbf{A}^T\mathbf{W}^2\mathbf{A} = \begin{bmatrix} \sum W^2(\mathbf{x})I_x^2(\mathbf{x}) & \sum W^2(\mathbf{x})I_x(\mathbf{x})I_y(\mathbf{x}) \\ \sum W^2(\mathbf{x})I_y(\mathbf{x})I_x(\mathbf{x}) & \sum W^2(\mathbf{x})I_y^2(\mathbf{x}) \end{bmatrix} \qquad (9)$$

Expressions (6) and (7) can also be considered as the weighted least-squares estimates of $\mathbf{v}$ from the normal velocities $\mathbf{v}_n = s\mathbf{n}$. That is, (6) is equal to

$$\sum_{\mathbf{x}\in\Omega} W^2(\mathbf{x})w^2(\mathbf{x})[\mathbf{v}\cdot\mathbf{n}(\mathbf{x}) - s(\mathbf{x})]^2 \qquad (10)$$

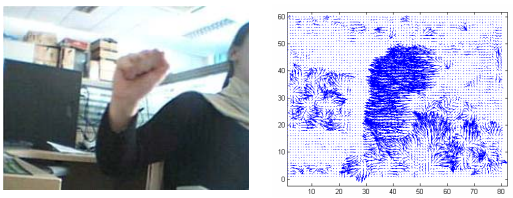

Figure 1 Optical flow tracking

Figure 1 shows a video frame in the simulation. In the optical flow field, constant optical flows can be observed for the moving hand.

3. Hardware Platform

Concerning the intensive video data and complex image processing tasks, the processor of the system should perform well in computation power in order to recognize the gestures real timely and correctly. With this consideration we choose DSP as the computational unit and distribute the image data to 4 DSPs using pipelining in sequential images processing.

The whole system consists of 3 components: video capture device, embedded Q6455 board and host PC. Q6455 acquire video image from the camera through Ethernet and carries the core processing of detecting, calculating, comparing and judging. Finally, the recognition results are transmitted to host PC. The host PC provides the graphic user interface which help users to set DSP parameters and observe the video processing results. The hardware platform block diagram is shown as Figure 2:

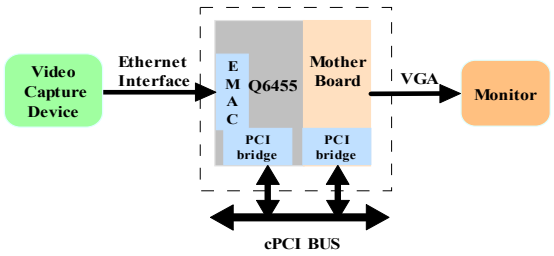

Figure 2 Hardware platform block diagram

3.1. Video Capture Device

The video Capture Device is made up of an analog camera and a video capture card which is equipped with an input interface of analog video image and an output interface of 1 gigabit Ethernet MAC. It converts the analog video input to YCbCr digital images. Each capture image is digitized to a matrix of 720 pixels×576 pixels with 24-bit color. The Q6455 DSP module can acquire 25 frames of image per second.

3.2. Q6455 signal processing board [8]

Q6455 signal processing board contains four TMS320C6455 DSPs.TMS320C6455 is the state-of-the-art DSP from Texas Instruments which provides 9600MIPS with a average power consuming of 2.3W. Concerning the interface flexibility and bandwidth, it integrates one 4-lane Serial RapidIO interface (SRIO) run at 3.125bps/line, one gigabit EMAC (Ethernet Media Access Controller) for long distance communication. The scheme of Q6455 signal-processing model is show as Figure 3.

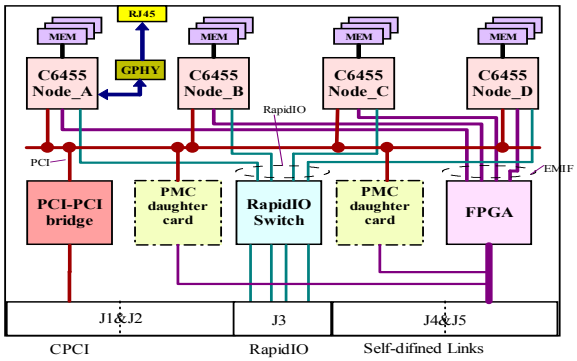

Figure 3 Scheme of Q6455 signal processing module

Up to four communication networks are used in Q6455 signal processing module. The gigabit Ethernet NIC is for Ethernet installing, using existing, well-debugged TCP/IP communication protocols. This low cost network, although only supporting low speed communication, is reliable and useful for remote system administration and maintenance. The cPCI bus Links four DSPs and two PMC daughter cards by PCI-PCI bridges at 64bitss@66MHz is narrow in bandwidth but convenient for intra-chassis system control and management. The SRIO network as a high bandwidth network is the backbone of our high performance scalable computing system, where most communication takes place. This network supports Qos and traffic-managing with an extremely low latency on the scale of several micro seconds. The core component of SRIO network is RapidIO Switch which makes it possible that any two DSPs in the SRIO network can operate as a 4x serial bi-directional link at 25Gbits/sec. Every C6455 node consists of one TMS320C6455, 512MB DDRII-500 SDRAM, 16M SBSRAM and 4MB Flash memory.

4. Embedded DSP Software Structure

The designed pattern of the DSP software comprises two parts: dynamic mode and static mode. The flow chart of the algorithm appears in Figure 4:

In order to fulfill the requirement of real time recognition, the processing work is distributed to four DSPs on Q6455 board: DSP_ A works as the data pre-processor and other three DSPs work as core processor. DSP_ A first receives image data from video capture device through Ethernet interface and then all the image frames are stored in L2 cache. After the pre-processing in DSP_A has been done, the results of image format conversion and skin color segmentation are sent to DSP_B, C and D respectively using SRIO.

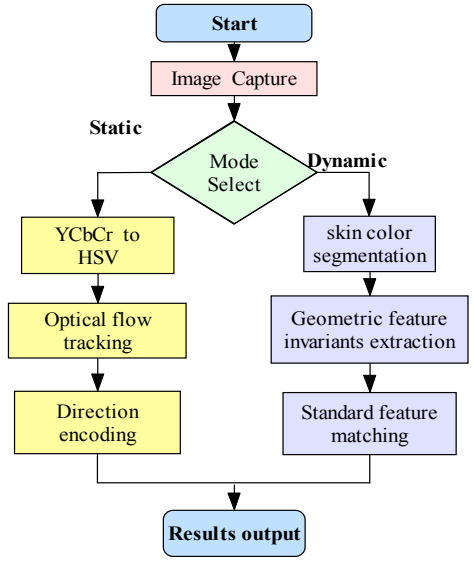

Figure4 Algorithm flow chart

The distribution of image frames from DSP_A to other 3 DSPs is demonstrated as Figure 5:

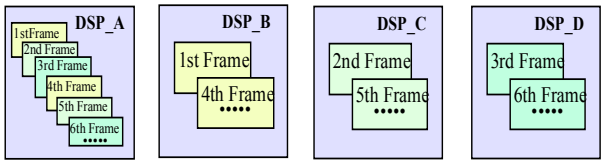

Figure 5 Data Distribution

DSP_B, C and D make the core processing on received data, such as feature extraction in static mode and optical flow tracking in dynamic mode. After the feature identification, recognition results can be shown on host PC.

4.1. Static mode

The static mode is based on supervised learning in which the learning rule is provided with a set of standard feature invariants. The gesture recognition process is commenced with an image pre-processing technique in DSP_A which involves transition of gray values, binaryzation and skin color segmentation which employs the YCbCr color space. A pixel is labeled as a skin pixel if its color values conform to the following constraints: [12]

$$235>Y[i]>45;\ Cb[i]<125\ \&\&\ Cr[i]>130.$$

The binarilized image data are then sent to other DSPs through SRIO and geometrical feature invariant are extracted. After comparing with the standard feature invariant, the recognition results are transmitted to host PC through PCI bus and display to end users.

4.1. Dynamic mode

The dynamic mode is also based on training process in which the training rule is a set of hand motion codes. In DSP_A, YCbCr color format is transferred into HSV color format with the consideration of efficient tracking. After the pre-processing, image data and the old feature point of last frame are sent to other DSPs. In order to reduce the computational complexity, the old feature point is used to determine the region using Lucas-Kanade optical flow to get the next new feature point. These points connect to paint the trace of the gesture on PC which will help the user finish the hand motion, simultaneously, DSP_A acquire next image frame and repeat the pre-processing, optical flow tracking and direction encoding. Comparing with the training collection, the recognition results are sent to the host PC.

5. Experiment results

The tests on system function and hardware performance have been carried out. The proposed system was used to recognize human's hand postures and dynamic gestures real-timely by surveying the acquired video image sequences. It was proved to be reliable and stable in the recognizing process.

5.1. System function index

Through testing, this system can process video images with the size of 720 pixels×576 pixels in 24-bit color, 25 frames of image sequence per second. Under the static mode, it can give the recognition results in 40ms. Under dynamic mode, the system follows the hand movements and then gives the meaning of the dynamic gestures in 40ms.

- The time spends on acquiring an image frame with size of 720 pixels×576 pixels and displaying it on host PC is less than 25ms. The whole computing time of image processing and judging costs less than 15ms.

- The pattern space of hand gestures is scalable in either static or dynamic mode. In the present, static mode can identify three hand postures — "Stone", "Scissor" and "Paper" while the dynamic mode can recognize "NO.0~9".

- During the tracking process of dynamic mode, host PC can show the trace of hand movement synchronously which assists the end user to finish the gesture.

5.2. Hardware performance index [8]

This system sets high requirements toward the computing speed and the storage of DSP board. Q6455 board's computation, memory and interconnect capabilities are mainly summarized in Table1: [8]

Table 1 Q6455Globle resources summary

Node (4)	DSP	9600MIPS@1.2GHz
	Memory	512MB DDRII-500; 16MB SBSRAM; 4MB Flash; 2MB L2 cache on chip;
Global	cPCI BUS	PICMG 2.3 Compatible; 64bit、66MHz; Transparent/Non-trans;
	Custom links	4, 1600MB/s each
	SRIO links	4, 1720MB/s each
	PMC	2, 1600MB/s each
	FPGA	25600MMACS
	NIC	120MB/s in GMII mode

5.3. Functional test

Hundreds of experiments have been performed under both complex and simple backgrounds, static and dynamic recognition mode respectively. The functional test results are given in Table 2 as follows:

Table 2 Functional test results

Video Data Acquiring Speed		237.3Mbps(25fps)	
Static Recognition Probability	Simple Background	97.6%	92.7%
	Complex Background	87.8%	
Dynamic Recognition Probability	Simple Background	95.4%	90.6%
	Complex Background	85.8%	

6. Conclusion

This paper presents a method which uses embedded DSP board and image processing techniques to recognize human's hand postures and dynamic gestures real timely. The adoption of direction encoding based on optical flow tracking in dynamic gesture recognition greatly increases the system's anti-jamming ability and the recognition space. At the same

time, this system shows high computational power benefited from the using of SRIO network and gigabit Ethernet NIC. This high-performance processing capability ensures a scalable system. Future work will focus on system optimization, with the emphasis on increasing the pattern space and algorithm complexity.

7. References

[1] V.I. Pavlovic, R. Sharma and T.S. Huang, "Visual Interpretation of Hand Gestures for Human-Computer Interaction: A Review", *Transactions on pattern analysis and machine intelligence,* IEEE, July1997, vol. 19, No.7, pp.677-695

[2] C. Chang, "Feature-Preserving Algorithm for Gesture Recognition", *Optical Engineering,* March 2007, vol.46(3),pp. 037201.1-037201.8

[3] S. Mitra and T. Acharya, "Gesture Recognition: A Survey", *Transactions on systems, man, and cybernetics-part C: Applications and reviews,* IEEE, May 2007, vol.37, No.3, pp.311-324

[4] J. Yamato, J. Ohya, and K. Ishii, "Recognizing human action in time-sequential images using hidden Markov model", *Proceedings of IEEE Computer Society Conference on Computer Vision and Recognition*, IEEE, Champaign, IL, June 1992, pp. 379–385.

[5] R.C. Gonzalez and R.E. Woods, *Digital Image Processing.* Prentice Hall, 1977.

[6] Y. Wu and T. S. Huang, "View-independent recognition of hand postures," *Proceedings. IEEE Conference on Computer Vision and Pattern Recognition,* Hilton Head Island, SC, June 2000, Vol. 2, pp. 88–94.

[7] A. Albiolt, L. Torres and E.J. Delp, "Optimum Color Spaces for skin detection", *Proceedings of the IEEE International Conference on Image Processing,* IEEE, Thessaloniki October 2001, vol.1, pp. 7-10.

[8] X.K. Zhang, G.M. Liu and M.G. Gao, "A High-Performance Scalable Computing System for Real-Time Signal Processing Applications", *Congress on Image and Signal Processing,* IEEE, 2008, pp.556-560

[9] C.H. Teh and R.T. Chin, "On Image Analysis by the Methods of Moments", *Transactions on pattern analysis and machine intelligence ,* IEEE, July1988, vol. 10,No. 4, pp. 496-513

[10]M.K. Hu, "Visual Pattern Recognition by Moment Invariants", *IRE Transactions on information theory,* February1962, pp. 179-187

[11] J.L. Barron, D.J. Fleet, and S.S. Beauchemin, "Performance of Optical Flow Techniques," *International Journal of Computer Vision,* February 1994, vol. 12(1), pp. 43-77.

[12] B. K. P. Horn, B. G. Schunck, "Determining Optical Flow", *Artificial Intelligence Laboratory*, MIT, Cambridge, MA, 1981, pp. 185-203.

[13] B. D. Lucas, T. Kanade, "An Iterative Image Registration Technique with an Application to Stereo Vision", *Proceedings of Imaging Understanding Workshop,* April 1981, pp. 121-130.

[14]S.H. Lim, A.E. Gamal, "Optical flow estimation using high frame rate sequences", *Proceedings of the IEEE International Conference on Image processing,* IEEE, Thessaloniki, October 2001, vol.2, pp. 925-928.

[15] P. Anandan, "A computational framework and an algorithm for the measurement of visual motion", *International Journal of Computer Vision,* January 1989, vol. 2, pp. 283-310.

[16] A. Verri, F. Girosi, and V. Torre, "Differential techniques for optical flow", *Journal of the Optical Society of America: Optics, Image Science and Vision,* May 1990. vol. 7, pp. 912-922,

[17] W.S.P. Fernando, Lanka Udawatta and Pubudu Pathirana, "Identification of moving obstacles with Pyramidal Lucas Kanade optical flow and k means clustering", *Third International Conference on Information and Automation for sustainability,* Melbourne, VIC, December 2007, pp. 111-117.

[18] **"**Analysis of Differential and Matching Methods for Optical Flow", *Proceedings of Visual Motion Workshop,* Irvine, CA, March 1989, pp. 173-180.

[19] A. Albiol, L. Torres, C.A. Bouman, and E. J. Delp, "A simple and efficient face detection algorithm for video database applications," *Proceedings of the IEEE International Conference on Image Processing,* IEEE, Vancouver, Canada, September 2000, vol. 2, pp. 239-242.