A 120 fps 1080p Resolution Block-based Feature Extraction Architecture Implementation for Real-time Action Recognition

Chun-Ting Yen, Wan-Yu Chen, and Liang-Gee Chen
Graduate Institute of Electronics Engineering
National Taiwan University
Taipei, Taiwan
Email: {akii960309, warmsnow}@video.ee.ntu.edu.tw, lgchen@ntu.edu.tw

***Abstract*—This paper introduces an efficient hardware accelerated feature extraction architecture with a high spec of 1920×1080 image resolution at 120 fps. We choose MoFREAK feature [1] to implement in our real-time action recognition system. MoFREAK is a local spatio-temporal feature, which combines the appearance and motion descriptor independently. We design a two phase architecture to balance the throughput difference between feature detection and feature description. Binary-mask image is adopted to detect feature point location efficiently. For feature description, to reduce high bandwidth requirement for spatial-temporal MoFREAK features, block-based keypoint technique is proposed to reduce bandwidth for grouped features. The synthesis result of our proposed architecture in TSMC 40nm technology works at 200 MHz with 1039K gate counts provides 1.2K block-based features per frame at 120 fps and 0.5K block-based features at 240 fps. With binary-mask image, we reduce about 88% cycles and bandwidth of scanning image. With block-based keypoint, we reduce about 81% of the salinet points and keypoints. The combination of binary-mask image and block-based keypoint reduces about 81% of the feature extraction system bandwidth.**

***Keywords*—Action Recognition, Feature Extraction, Block-based Keypoint, Architecture Design**

I. Introduction

One of the ultimate goal of computer vision is to design a intelligent robot, the first step is to understand the semantic meaning behind videos. Therefore, action recognition will be the key of the intelligent robot. A general action recognition framework can be divided into four steps: image pre-processing, keypoint detection, descriptor generation, and classification. To extract useful information from the enormous amount of video data, feature detection and descriptor generation are the most important parts to reduce the recognition data space and determine the recognition accuracy. The variations of videos increase the difficulty of extracting precise feature points, leading many researchers to develop better algorithms aiming at raising the recognition accuracy. However, the computation complexity of feature extraction in videos is too complicated to be real-time in past researches, such as dense trajectory [2] and fast HOG3D [3]. In this paper, we proposed an efficient action recognition architecture to extract the spatio-temporal features with ASIC implementation to accelerate the feature extraction for real-time action recognition at 120 fps 1080p (1920×1080) resolution.

There are many types of human activities recognition. Aggarwal et al. [4] organized this topic and divided into four different levels: gesture, action, interactions, and group activities. They also divided different approaches into a approach-based tree-structured taxonomy. In our action recognition system, we choose the local spatio-temporal feature called MoFREAK [1] to provide a efficient 3D feature for reliable action recognition accuracy. MoFREAK is a binary feature combines appearance descriptor to capture static information and motion descriptor to describe dynamic information in the video, with high recognition accuracy in datasets.

II. Irregular Feature Extraction

MoFREAK feature extracts appearance feature by FREAK [5] descriptor and motion feature by MIP [6] descriptor. The overview of our proposed architecture is illustrated in Fig. 1. The proposed feature extraction architecture for action recognition system is divided into four major parts: Keypoint Detector, Image Preload, Appearance Descriptor, and Motion Descriptor. As mentioned in MoFREAK, the input frames are processed into the absolute difference gray-level frames o permit implicit motion encoding. However, the number and distribution of features are content dependent. This characteristic results in a waste of feature extraction system bandwidth on fetching image patches without data reuse. Besides, we investigate the scenario that features has a high probability to appear together. Therefore, we present Binary-mask Image for feature description and Block-based Keypoint to reduce system bandwidth by the novel on-the-fly block based data-reuse scheme.

A. Binary-mask Image

As shown in Fig. 1, since the bus bitwidth is 128 bits, we need 129,600 cycles to scan the entire 1080p gray-level frame and waste enormous amount bandwidth. If we convert gray-level image into the binary image as an image mask, we can rapidly screen the entire frame in 16,200 cycles, which reduces 87.5% cycles and bandwidth of scanning image. Moreover, since we use a binary-mask image, we can also rapidly screen the salient points (non-zero pixels) by logic gate.

B. Block-based Keypoint

MoFREAK use multi-scale FAST corner detector [7] introduced in BRISK [8]. The detected corner keypoints are

978-1-4673-6853-7/17/$31.00 ©2017 IEEE

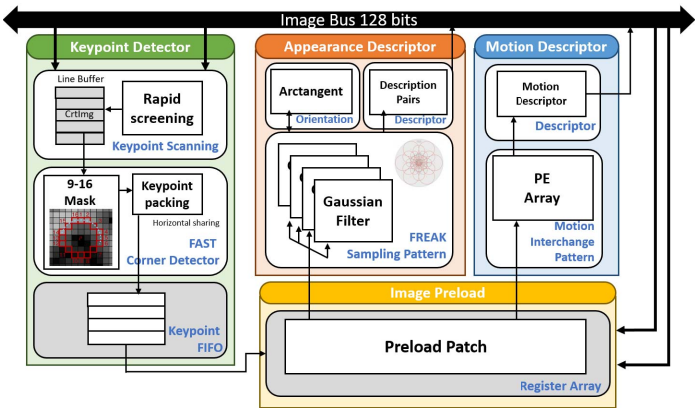

Fig. 1. The illustration of proposed architecture block diagram.

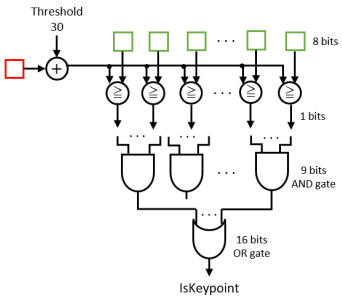

Fig. 2. The illustration of FAST 9-16 mask for keypoint detection.

usually adjacent to each other since the pixels around corner mostly meet the requirment of corner detector. Therefore, we group 10 adjacent pixels in horizontal direction as a block-based keypoint. With block-based keypoint, we can reuse the fetched image patch by sliding the description pattern to save the reload bandwidth and make keypoints regular for feature description. We use UT-interaction dataset [9] to measure the number of salient points and keypoints per frame, since it is the action dataset with the highest resolution. We resize the frame to 1080p resolution and apply block-based keypoint technique. The number of salient points and keypoints without grouping are about 12,500 and 3,900. After grouping pixels as a block-based keypoint, the number of salient points and keypoints are about 2,400 and 750, which reduce 81% number of salient points and keypoints and save 80.5% of the feature extraction system bandwidth.

III. Hardware Architecture

A. Keypoint Detector

We use the binary-mask image rapidly screen the whole frame and examine the salient points in the unit of a block-based keypoint. Value 1 represents salient point (candidate keypoint) and value 0 represents the background pixel. If the salient point is detected, the detector will load the required patch to examine. The keypoint detector applies circular mask with 16 pixels on the salient point to compare intensity between salient point and pixels on the circular mask. If there are more than 9 contiguous pixels on the circular mask brighter or darker than salient point, it will be detected as a corner, which is keypoint. We use logic gates to quickly examine whether salient point meet the condition as shown in Fig. 2. Then, we put block-based keypoint into the keypoint FIFO as the 2-stage pipeline to separate feature detector and feature descriptor to balance the throughput.

B. Image Preload

Once the keypoint FIFO output a block-based keypoint, image preload block will start to fetch the require patches. Motion descriptor requires both current and previous frame in patch size 32×19 to describe motion changes of the keypoint. Appearance descriptor requires current frame in patch size 64×51. Therefore, we load one 64×51 current frame patch to share with both appearance and motion descriptor and one 32×19 previous frame patch for motion descriptor. Since we apply block-based keypoint technique, we can share on set of current and previous frame patch by sliding the description pattern to describe the grouped keypoint in a block-based keypoint.

C. Appearance Descriptor

The FREAK sampling pattern consists of 43 sampling circles, which has 7 concentric circular layers with each 6 sampling circles and 1 sampling circle in the center. The sampling circles on the same layer use the same size of Gaussian filter. We exploit the symmetric and separable property to convert 2D Gaussian filter into the combination of horizontal and vertical 1D Gaussian convolutions. To accelerate the appearance description, the 43 sampling circles are fully parallel operations. After having all the values of sampling circles, FREAK descriptor uses arctangent to calculate the orientation of the keypoint and rotates the sampling pattern to align the orientation. Then, FREAK compares the intensity between the rotated sampling circle values and obtain the 16 bytes appearance feature. FREAK divided 2π into 256 equal parts.

To compute the orientation of the keypoint, we first calculate the value of the vertical and horizontal direction components by the orientation pairs presented in FREAK. Since the arctangent value can be obtained by two arguments $atan2(y, x) = arctan(\frac{y}{x})$. We use cross multiplication to find which orientation part that keypoint belongs to by comparing the value of the vertical and horizontal components to the normalized ratio. The detail of orientation block is shown in Fig. 3.

D. Motion Descriptor

As described in MoFREAK, MIP uses SSD to compare the intensity changes between the surrounding pixels of keypoint. We replace SSD by SAD to make the computation more hardware friendly with the loss of 3% accuracy which is acceptable. Because the computation of the MIP descriptor is much easier than FREAK descriptor, we exploit its regularity and apply the PE array to gain parallel of the computation. We use the 8 times parallel PE (processing elements) array to compute the SAD value of 8 possible motion directions in previous frame. As shown in Fig. 4(a), the i-axis represents eight possible motion directions and the j-axis represents the 9 pixels in a 3×3 patch for SAD computation, which the Pt

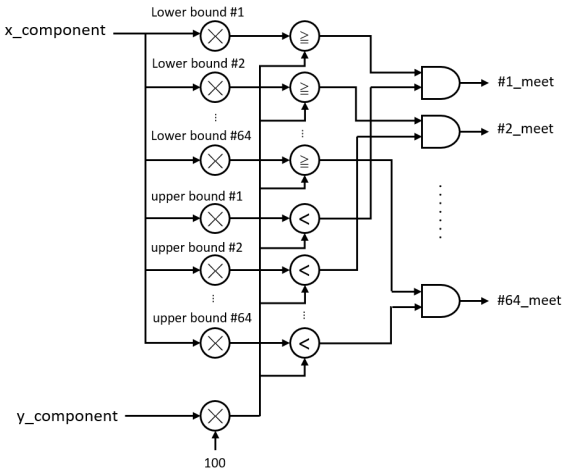

Fig. 3. The illustration of the orientation block using the cross-multiplication to calculate the ratio between horizontal component x and vertical component y.

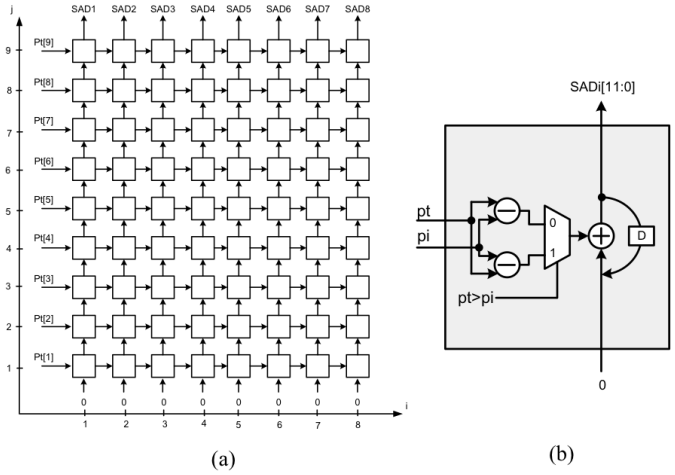

Fig. 4. (a) The illustration of the PE array with 8-times parallel, (b) The illustration of PE details.

signals are broadcast signals. The detail of each PE is shown in Fig. 4(b). The selection signal chooses the positive subtraction result and adds it to the previous subtraction result to calculate SAD value. We obtain 16 bytes motion feature by comparing the SAD values in eight possible motion directions.

IV. Implementation Result

The synthesis result of our proposed feature extraction architecture in TSMC 40nm technology at 120 fps 1080p resolution operates at 125 MHz with about 6100K gate counts and 3.9 Kbytes memory usage. Though the result already met our real-time spec, we still want to increase the clock rate and reduce the design area. We have three optimization strategies, which will be explained below.

A. Critical Path Reduction

The critical path occurs at the arctangent block of FREAK descriptor, which starts from calculating the orientation pairs to obtain x, y components and ends at the selector of the orientation result. The reason is that because of the multiplications of finding the upper bound and the lower bound of each divided orientation in cross-multiplication method. To prevent the bitwidth overflow, we use a long bitwidth to represent the x, y component. This causes22 the critical path happened when two argument with long bitwidth perform multiplication. We replace cross-multiplication by long division to obtain $\frac{y}{x}$ in $arctan(\frac{y}{x})$. Since the long division only use one subtractor to perform the division. The synthesis result of long division, the optimized architecture works at 200 MHz that increases 60% system bandwidth and reduces 97% arctangent block gate counts.

B. Area Cost Down

The parallel operation of the 43 sampling circles in FREAK pattern reduces the cycles of Gaussian filters computation. However, we have to pay a great price on the design area. Therefore, we would like to share the filters instead of the fully parallel operation. Since the sampling circles on the same layer use the same Gaussian filter size, we share the Gaussian filter in the same layer by reducing 43 Gaussian filters to 8 Gaussian filters for 8 layers. Moreover, we further reuse the input selector of the Gaussian filters. Because the sampling pattern rotates 256 different angles depending on keypoint orientation, the input Gaussian filter selectors of each sampling circle have 256 input options. Since the sampling circles rotate on the same concentric circular layer will still lie on the same concentric circle, we go further to reuse the Gaussian input selector on the same layers. The result shows that, with the resource sharing of Gaussian filters and input selectors, we can reduce 92.7% gate counts in Gaussian filter block.

C. Utilization Improvement

We use the ping-pong mode buffer to improve the utilization of image preload block. The ping-pong mode buffer uses two selection signals to switch between two buffers. When feature descriptor is using the data in buffer 1, the selectino signal will switch the incoming block-based keypoint to buffer 2 to continue loading the next required patches.

D. Sythesis Result

The synthesis result of optimized design in TSMC 40nm technology is shown in Tabel I. Because we apply the ping-pong mode buffer in image preload block, the memory usage increases to 7.9 Kbytes. With the optimization strategies, we reduce about 82% gate counts compared to the design without optimization. The optimized design achieve real-time spec at 120 fps 1080p resolution and operate at 200MHz in 1039K gate counts since we replace the cross-multiplication by long division to reduce the critical path. The detail of the bandwidth of each block is shown in Tabel II, which #SP denoted as the number of salient points and #KP denoted as the number of keypoint. The total bandwidth per frame without any optimization is 17.79 Mbytes/frame, after applying the binary-mask image and block-based keypoint, the bandwidth per frame can be reduced to 3.48Mbytes/frame and saves about 81%. The total bandwidth of the feature extraction system is 417.6 Mbytes/sec at 120 fps. We can further increase the frame rate to 240 fps, and the feature extractino system bandwidth will raise to 835.2 Mbytes/sec.

TABLE I. COMPARISON OF SYNTHESIS RESULTS BEWTEEN ORIGINAL ARCHITECTURE AND OPTIMIZED ARCHITECTURE.

Item	TSMC 40 nm Technology	
	Original Architecture	After Optimization
Operating Frequency	125 MHz	200 MHz
Rsolution	Full HD 1920×1080	
Gate Count	6017 K	1039 K
Memory	3.9 Kbytes	7.9 Kbytes
Bandwidth	3.48 Mbytes/frame → 417.6 Mbytes/sec @ 120fps → 835.2 Mbytes/sec @ 240fps	

TABLE II. THE BANDWIDTH PER FRAME OF EACH BLOCK. (THE UNIT OF BANDWIDTH IS MBYTES/FRAME)

FAST Detector - Rapid Screening	Cycles	#SP	Bandwidth
Without Optimization	129600	N/A	2.0736
With Binary Mask Image	16200	N/A	0.2592
FAST Detector - Line Buffer	**Cycles**	**#SP**	**Bandwidth**
Without Optimization	4	12437	0.7960
With Block-based Keypoint	7	2392	0.2680
Image Preload - Current Frame	**Cycles**	**#KP**	**Bandwidth**
Without Optimization	204	3844	12.5468
With Block-based Keypoint	204	732	2.3893
Image Preload - Previous Frame	**Cycles**	**#KP**	**Bandwidth**
Without Optimization	38	3844	2.3372
With Block-based Keypoint	38	732	0.4451
MoFREAK Features	**Cycles**	**#KP**	**Bandwidth**
Without Optimization	1	3844	0.0615
With Block-based Keypoint	1	732	0.0117

E. System Comparison

Due to the lack of previous works of feature extraction architecture, which focus on action recognition with ASIC technology, we compared our design to [10], [11], [12] these three local feature extraction works. [10] proposed a layer parallel SIFT with integral image, and it is a parallel hardware design with an on-the-fly feature extraction. The implementation used 580K gate counts with 90 nm CMOS technology, and offered 6K features points/frame for VGA image (640×480) at 30 fps and about 2K feature points/frame for 1080p image at 30 fsp at the clock rate of 100 MHz. [11] presented an optimized SURF hardware implementation using about 600K gate counts with TSMC 65 nm technology, operate at 200MHz and capable for 1080p image at 57 fps but it did not mention how many feature points it can offer. [12] introduced the FAST detector with BRIFE descriptor architecture implementation using 78.3K gate counts by Samsung 0.13μm technology, which achieve 94.3 fps in 1080p resolution at 200MHz operating frequency for about 0.5K feature points per frame.

It is hard to fairly compare among these works since our design extracts 3D feature for real-time action recognition application. The proposed design is capable of 1080p resolution with 1.2K block-based keypoints at 120 fps and 0.5K block-based keypoints at 240 fps, which are measured in the worst case of block-based keypoint that all of the grouped points are detected as keypoints. The comparison is shown in Table III. If you convert the worst-case block-based keypoints into the number of individual feature points are 12K feature points at 120 fps with 417.6 Mbytes/sec bandwidth and 5K feature points at 240 fps with 835.2 Mbytes/sec bandwidth. Compared to other works, we provide a much higher frame rate with much more feature points than others and achieve real-time spec for action recognition.

TABLE III. COMPARISON OF RELATIVE WORKS WITH ASIC IMPLEMENTATION. (*BLOCK-BASED KEYPOINTS)

	[10]	[11]	[12]	Proposed
Descriptor/ Descriptor	SIFT	SURF	FAST/ BRIFE	MoFREAK
Technology	90 nm	TSMC 60 nm	Samsung 130 nm	TSMC 40 nm
Operating Frequency	100 MHz	200 MHz	200 MHz	200 MHz
Resolution	VGA 1080p	1080p	1080p	1080p
Keypoints/Fsp	6K(VGA) 30fps 2K(1080p) 30fps	N/A 57fps	0.5K 94.3fps	1.2K* 120fps 0.5K* 240fps
Gate Count	580 K	600 K	78.3 K	1039 K
Memory	66.625 Kbytes	400 Kbytes	128 Kbytes	7.9 Kbytes

V. CONCLUSION

We provide a efficient hardware accelerated feature extraction architecture for real-time action recognition at 120 fps 1080p resolution. Since the maximum number of features per frame in our proposed design is much more than other works, in the future we can use sliding window to find the ROI of several persons in a complex scenario and recognize the interaction of each person.

REFERENCES

[1] C. Whiten, R. Laganiere, and G.-A. Bilodeau, "Efficient action recognition with mofreak," in *International conference on Computer and Robot Vision (CRV)*. IEEE, pp. 319–325, 2013.

[2] H. Wang, A. Kläser, C. Schmid, and C. Liu, "Dense trajectories and motion boundary descriptors for action recognition," in *International journal of Computer Vision*, vol. 103, no. 1, pp. 60–79, 2013.

[3] N. Li, X. Cheng, S. Zhang, and Z. Wu, "Realistic human action recognition by Fast HOG3D and self-organization feature map," in *Machine Vision and Applications*, vol. 25, no. 7, pp. 1793–1812, 2014.

[4] J. K. Aggarwal and M. S. Ryoo, "Human activity analysis: A review," *ACM Computing Surveys (CSUR)*, vol. 43, no. 3, p. 16, 2011.

[5] A. Alahi, R. Ortiz, and P. Vandergheynst, "Freak: Fast retina keypoint," in *Computer vision and pattern recognition (CVPR)*. IEEE, pp. 510–517, 2012.

[6] O. Kliper-Gross, Y. Gurovich, T. Hassner, and L. Wolf, "Motion interchange patterns for action recognition in unconstrained videos," in *European Conference on Computer Vision*. Springer, pp. 256–269, 2012.

[7] E. Rosten and T. Drummond, "Machine learning for high-speed corner detection," in *European conference on computer vision*. Springer, pp. 430–443, 2006.

[8] S. Leutenegger, M. Chli, and R. Y. Siegwart, "Brisk: Binary robust invariant scalable keypoints," in *International conference on computer vision*. IEEE, pp. 2548–2555, 2011.

[9] M. Ryoo, C.-C. Chen, J. Aggarwal, and A. Roy-Chowdhury, "An overview of contest on semantic description of human activities (sdha) 2010," in *Recognizing Patterns in Signals, Speech, Images and Videos*. Springer, pp. 270–285, 2010.

[10] L.-C. Chiu, T.-S. Chang, J.-Y. Chen, and N. Y.-C. Chang, "Fast sift design for real-time visual feature extraction," in *IEEE Transactions on Image Processing*. vol. 22, no. 8, pp. 3158–3167, 2013.

[11] L. Liu, W. Zhang, C. Deng, S. Yin, S. Cai, and S. Wei, "Surfex: A 57fps 1080p resolution 220mw silicon implementation for simplified speeded-up robust feature with 65nm process," in *Proceedings of the IEEE 2013 Custom Integrated Circuits Conference*, IEEE, pp. 1–4, 2013.

[12] J.-S. Park, H.-E. Kim, and L.-S. Kim, "A 182 mw 94.3 f/s in full hd pattern-matching based image recognition accelerator for an embedded vision system in 0.13-cmos technology," in *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 23, no. 5, pp. 832–845, 2013.