Temporal Coherence Based Mismatch Removal for High Frame Rate and Ultra-Low Delay System

Zhe Wang, Songlin Du, and Takeshi Ikenaga
Graduate School of Information, Production and Systems, Waseda University, Kitakyushu, Japan

Abstract—**High frame rate and ultra-low delay matching system plays an increasingly important role in human-machine interactive application. Because these applications are used in real-life complex situation which have requirement for matching accuracy, mismatch removal is needed. Although many mismatch removal algorithms have been proposed, few of them are hardware friendly and excuted in real-time application with lower delay due to complicated arithmetic operations and using iterative methods. This paper proposes a hardware-oriented temporal coherence based mismatch removal method for high frame rate and ultra-low delay matching system. In the proposed method, three adjacent frames are used with three times matching among each other, if the keypoint finds itself after three times matching, it is directly treated as true, which means the keypoint satisfies the temporal constraints. For the keypoints which do not find themselves, it makes statistics of matching frequency for different parts in the image to get current matching area, and uses the matching area to distinguish. Software experiment result shows that proposed mismatch removal method increases by 4.05% than basic ORB match, and by 0.33% than conventional work RANSAC. Hardware experiment result indicates that the designed image processing core successfully achieves real-time processing of 784fps VGA (640×480) video on field programmable gate array (FPGA), with a delay of 0.828 ms/frame.**

I. Introduction

During these years, high frame rate and ultra-low delay image processing system is becoming more and more important in human-machine interactive applications such as gesture recognition [1], object tracking, projection mapping [2] and so on. Hu [3] proposed a local feature based hardware-oriented high frame rate and ultra-low delay ORB matching system, which is robust to translation and in-plane rotation. Xu [4] solves the deformation problem based on Hu's work. In order to improve the matching performance and make matching system more robust, a hardware-friendly mismatch removal method needs to be designed with lower complexity calculation and parallel processing to achieve the ultra-high speed on FPGA.

For the purpose of mismatch removal, Random Sample Consensus(RANSAC) proposed by Fischler and Bolles [5] is an effective and widely used method, which gets consensus set of true matches by iterative random sampling of keypoints pairs between two matching images to calculate homography matrix, the method finishes after it finding the optimal fitting consensus true matches set. The advantage of RANSAC is able to do robust calculation of consensus, even a significant number of mismatches are existing, the results after RANSAC still have higher accuracy. Another efficient mismatch removal method is Hierarchical Progressive Trust Model (HPT) [6] proposed by Du, which uses three-layers model to expand true matches set one by one. HPT uses thin-plate splines transformation and expectation maximization (EM) algorithm to construct model and estimate optimal parameters, and the EM algorithm finishes when it converges. HPT has better performance than RANSAC and is robust for most of situation. Both of RANSAC and HPT actually try to get globally geometrical relationship between two images, which means they start calculation after whole images are input. But waiting for input of images increases extra delay for hardware system. Both of two methods are iterative without upper bound on the time they takes, there are also data dependency between each time iteration, which is against design principles of certain resource cost and parallel processing for hardware programming, it also causes the higher delay of whole system until they get optimal results. Both of two methods use operations such as division, square root, logarithm and so on, which are complicated, time consuming and unfriendly for high frame rate and ultra-low delay system.

Aiming at high frame rate and ultra-low delay matching system, this paper proposes a temporal coherence and blocking weighting based hardware friendly mismatch removal method. Each mismatch removal operation needs to use three adjacent frames and do three times matching among each other, and the target is to remove mismatches between the second frame and the third frame. If the keypoint finds itself after three times matching, it is directly treated as true, which means the keypoint satisfies the temporal constraints. For the keypoints which don't find themselves, it makes statistics of matching frequency for different parts in the image to get current matching area, and uses the matching area to distinguish. Both of these two proposals are based on strong temporal coherence of images from high frame rate system, both of them do mismatch removal on keypoint level to avoid higher delay of global processing. These two proposals are designed at the point of lower complexity, parallel processing, and not using complicated functions such as division, sin, tan and so on. With simple operations, no iteration, and avoiding global processing, it achieves the goal of ultra-low delay.

II. Proposed mismatch removal method

This section shows the detailed steps of proposed mismatch removal method. The framework of proposed method is shown in Fig. 1. Our method is based on our previous work [7] ORB matching system which just implemented basic matching and did't deal with mismatches. The proposed method uses keypoints descriptors from each three adjacent frames as input, and the output is all the true matches between the second frame and the third frame after mismatch removal. Firstly, three adjacent frames are used to do brute force match among each other. After three times matching, there is a matching route. For each keypoint in the second frame, through this matching route, it finally goes back to the second frame and find itself. The purpose of this step is to firstly find the keypoints with strong temporal coherence which is powerful evidence to prove these matches are true. Secondly, although some keypoints don't have strong temporal coherence, their matching are objectively true. To avoid removing these true matches, using true matches from three times matching to make statistics of matching frequency for each part in image to get matching area which means currently keypoints distribute frequently in the area, then it checks whether these matches belong to the area and distinguish true or false.

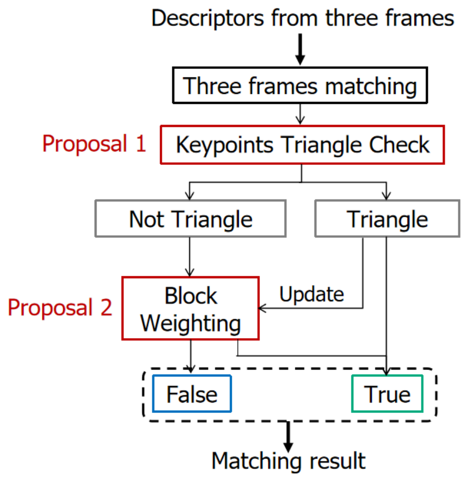
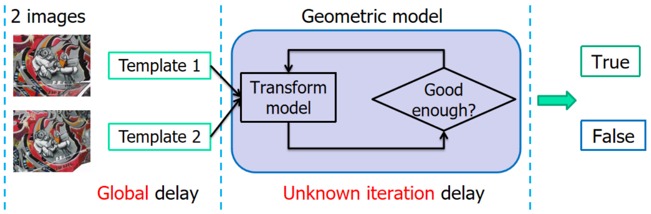


Fig. 1: Framework of the proposed mismatch removal method.

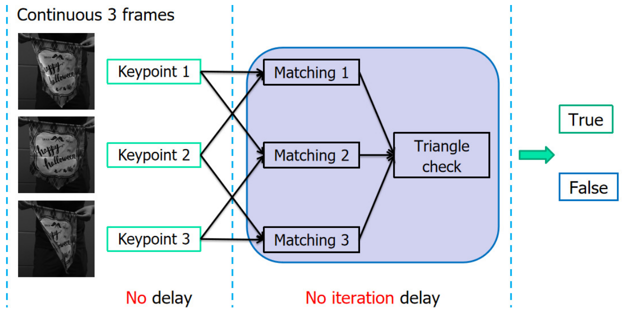
Subsection A is about using temporal coherence to find some initial true matches. In subsection B, block weighting is used to get the matching area with the true matches and distinguish the left matches. Subsection C shows the hardware structure and implementation of proposed method on FPGA.

A. Temporal Constraint Based Keypoints Triangle Check

Fig. 2 shows conceptual differences between conventional mismatch removal methods and the first proposal. Conventional methods use complex arithmetic operations and iteration algorithms to calculate geometrical relationship between two images. But for different images, there are different iteration times, which means unknown resource cost and higher delay for hardware system. For high frame rate video, there is extremely strong relationship between adjacent frames. When matching with images from high frame rate system, images provide rich information and potential correspondence in time dimension so that matching is a continuous process, this is a key difference compared to general situation. With information in time dimension, mismatch removal method does not need to construct model to get geometrical constraints between two images, so that complex mathematical operation and iteration are not required. Only focusing on keypoint level, global processing delay is avoided, and parallel operation is also used to each match pair. So that's why this proposal is hardware friendly with ultra-low delay.



(a) Conventional methods



(b) Proposed methods

Fig. 2: Conceptual differences of proposal 1

If time interval of input images is extremely small, it means slight differences in adjacent frames, and same keypoints constantly appear at least three frames until they are disappeared due to object transformation. So the method uses each three frames as input in one time, and matching starts at the second frame. After three times matching for each keypoint, there is a matching route. By checking this matching route, all matches are divided into two parts: triangle part and not triangle part. The detailed illustration is shown in Fig. 3 and Fig. 4.

For triangle part in Fig. 3, from keypoint A through the matching route, it finally finds itself, which means all keypoints in this route are the same keypoints and there is no doubt that match between two same keypoints is true. So matches from this part are directly treated as true. The triangle is named for easily understanding due to the shape of matching route.

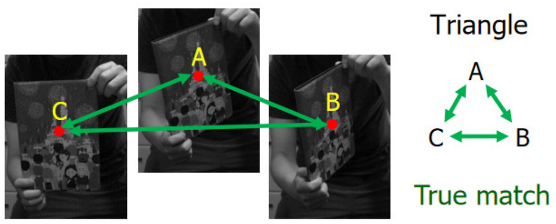
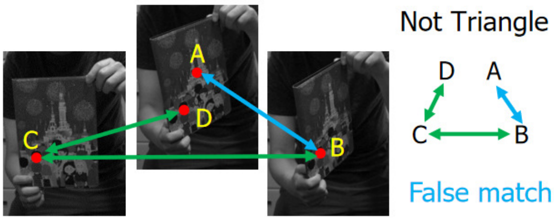
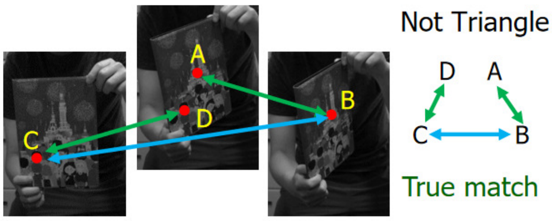


Fig. 3: Triangle match

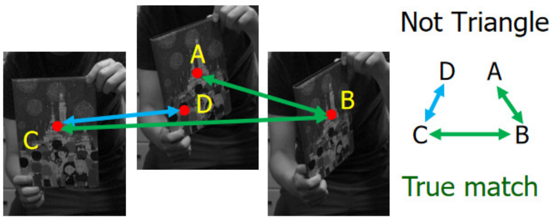
For not triangle part in Fig. 4, from keypoint A through the matching route, it finds other keypoint, which means at least one time mismatch happens. Firstly, if mismatch happens between the second and the third frame, because the target is to remove mismatch between the second and the third frame, these matches are false and should be removed. But if mismatch happens between the third frame and the first frame or the first frame and the second frame, for example a new keypoint just appear in the second and the third frame which matches are true match. Under these situations, obviously the matching routes are not triangle, and only this single proposal can't distinguish where mismatch happens. And if all this part is removed, it may happen that although after mismatch removal all matches are true but removed true matches are more than false matches, which doesn't mean the good performance of mismatch removal method. So for not triangle part, block weighting is proposed to solve this problem.



(a) Mismatch happens between frame 2 to frame 3



(b) Mismatch happens between frame 3 to frame 1



(c) Mismatch happens between frame 1 to frame 2

Fig. 4: Situations of triangle check.

B. *Block Weighting Judgement of Matching Area Statistics*

Fig. 5 shows the concept of blocking weighting judgement. For images from high frame rate system, object transformation is very slow. So moving range of each keypoint is very limited. For matching of transformation process, it must be in the certain range of areas in image, which means keypoints keep appearing and matching in this area. These areas can be obtained by making statistics of matching frequency for each part in image and described as weighting for each part. Through updating the weighting of each part with only true matches, it shows the truely matching areas in image. And for matches from not triangle part, if they are in these areas, they are true, otherwise they are false. Here are detailed steps of proposal block weighting judgement.

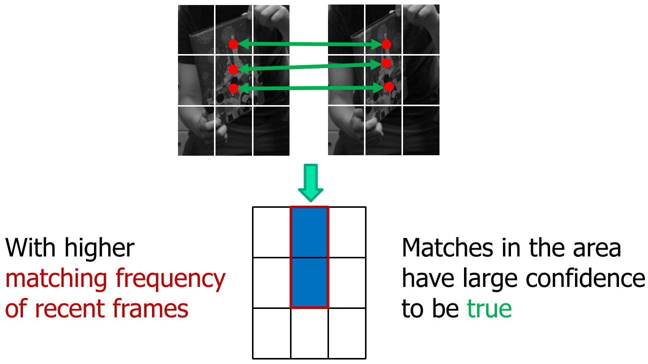


Fig. 5: Concept of proposal 2

Firstly, each image are divided into blocks. The weighting of each block means matching frequency in this block, and higher weighting means currently keypoints are distributed and matched in this block. After first proposal of triangle check, some true matches with strong temporal coherence have already found, these true matches are used as update factor for weighting of each block. The update rule for block w_{ij} is:

$$\begin{cases} w_{ij} + xn, & \text{if block with number } n \text{ true matches} \\ w_{ij} - y, & \text{if block with zero matches} \\ w_{ij} \geq 0, & \text{always greater than or equal to zero} \end{cases} \quad (1)$$

The parameter x is postive factor for true match. The parameter y is parameter for block with no matches. The reason of set negative factor is for example in Fig. 6. When the object moves down, if there is no negative factor to decrease the weighting of block without any matches, the weighting won't reflect the truely matching area.

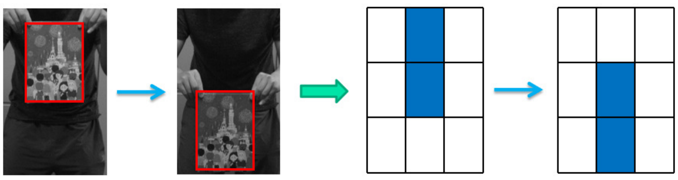


Fig. 6: Example of block weighting update

Because matching is transformation process, the initial weighting of each block needs to accumulate after several frames. A warm-up stage is needed by first proposal of keypoints triangle check to generate initial weighting for each block. So for frames of the warm-up stage, only keypoints triangle check is excuted to generate initial weighting. After warm-up, both two proposals are excuted to do mismatch removal.

TABLE I: Matching performance.

Sequence	ORB match	RANSAC	HPT	Time coherence	Our method
3D rotation	86.54	91.01	94.16	75.11	92.11
Rotation & Deformation	91.96	97.26	97.64	90.62	96.97
2D rotation	89.76	94.33	94.77	79.10	94.51
Illumination	95.25	97.35	97.80	93.79	97.01
Translation	84.53	87.42	91.03	75.50	88.68
Scale change	90.00	93.18	94.42	79.94	93.46
Camera moving	91.74	95.30	96.51	88.47	95.41
Average	89.97	93.69	95.19	83.22	94.02

C. Hardware Structure of Matching System

For both two proposals, there are only matching, which actually are xor operation of hamming distance calculation, addition and substraction. No complex arithmetic operations exist. And the method is doing processing for each keypoint without waiting for global image, there are also no iterations. For three times matching, actually there are two times parallel matching on hardware, because one time matching result has already obtained in last time of mismatch removal. For block update, all blocks are parallelly updated after the last pixel of the image input. So the whole method obeys the design principle of hardware system with high parallelism. Because all operations are easy, iteration and global processing are avoided for hardware, it achieves the target of ultra-low delay. Fig .7 shows the total hardware structure. PC, high frame rate camera and FPGA together compose the whole system.

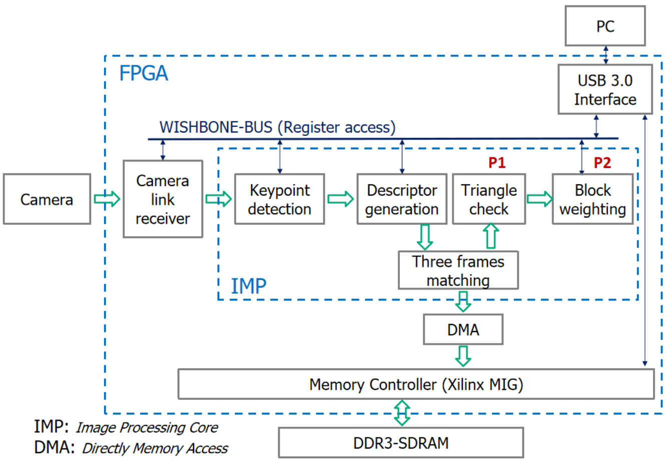


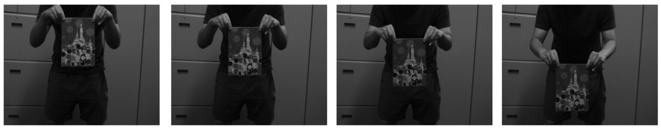
Fig. 7: Hardware structure

III. Evaluation Results

A. Test Dataset

We use high-speed camera BASLER acA2000-340, which captures 640×480 video with a frame rate of 784 fps. The captured images are sent to PC for further matching processing.

We test 7 types of sequences, and all sequences include certain kinds of transformation factors. All 7 types includes: 3D rotation, rotation & deformation, 2D rotation, illumination, translation, scale change, and camera moving. Fig. 8 shows some examples of test sequences.



(a) Translation

(b) 2D rotation

Fig. 8: Examples of test sequences

B. Matching Result and Analysis

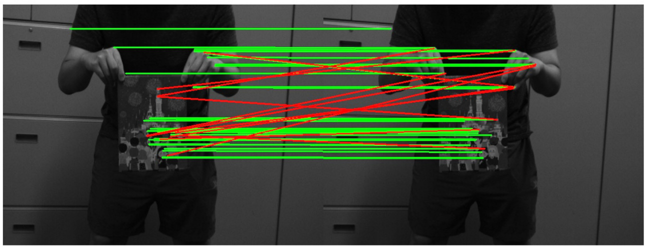
The matching performance is evaluated by the widely used metrics precision, recall and f-score [8], which are defined as follows:

$$\text{precision} = \frac{\#\ \text{correct matches}}{\#\ \text{correct matches} + \#\ \text{false matches}} \tag{2}$$

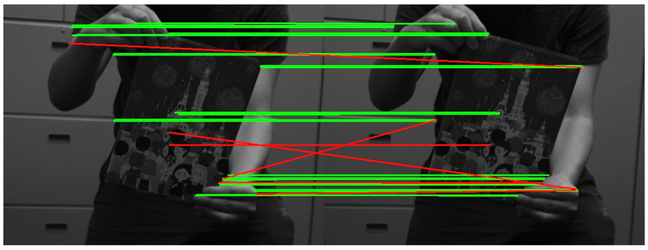
$$\text{recall} = \frac{\#\ \text{correct matches}}{\#\ \text{total matches in ground truth}} \tag{3}$$

The f-score is defined as the function of precision and recall, i.e. and So higher f-score reflects better matching performance.

We also compare performance with ORB, RANSAC, HPT, and just temporal coherence based keypoints traingle check. The experimental results on f-score are shown in Table. I. And larger f-score means better matching performance. As can be seen from Table. I, our method is obviously higher than original ORB, little higher than RANSAC, and still lower than HPT. Fig. 9 shows some examples of matching result, the green line means true match and red line means removed mismatch. As can be seen, green lines distribute in smooth space, and red lines distribute in random space, and our method do remove some mismatches.



(a) Translation



(b) 2D rotation

Fig. 9: Examples of matching result

C. Hardware Evaluation

We use Xilinx Kintex-7 XC7K325T FPGA board, BASLER acA2000-340 Camera and Intel Core i7-4790 CPU 3.6GHz PC to implement the high frame rate and ultra-low delay matching system based on the proposed mismatch removal method. The logic synthesis and implementation on FPGA are performed by Vivado V2017. 2(64-bit). Fig. 10 shows hardware demostration of our proposed method.

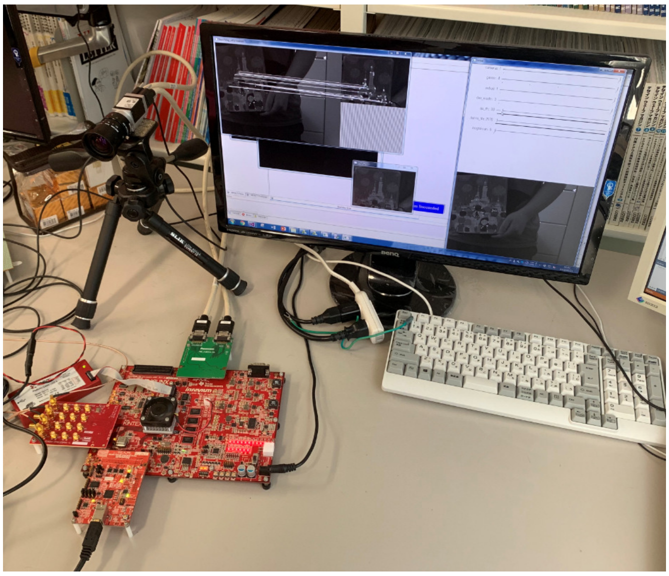


Fig. 10: Hardware demostration

Table. II shows the hardware performance of the system. The processing delay for a frame is less than 1ms, which means the capability of supporting the real-time processing of the system under the current input frequency and with VGA (640×480) video.

TABLE II: Hardware resource cost

Item		Hardware performance
Resource utilization	# LUT	148355 (72.79%)
	# LUTRAM	29081 (45.44%)
	# FF	123469 (30.29%)
	# BRAM	27.50 (6.18%)
	# DSP	36 (4.29%)
	# IO	208 (41.60%)
	# MMCM	3 (30.00%)
	# PLL	1 (10.00%)
Performance index	Input frequency	100.00MHz
	Input frame rate	784fps
	Processing delay	0.828ms/frame

IV. Conclusion

In this paper, a hardware-oriented mismatch removal method for high frame rate and ultra-low delay matching system has been proposed. The matching result shows that the proposed method has much better performance than basic match. For conventional mismatch removal method, the performances of proposed method with much more complexity reduction and higher parallelism are even better than widely used method RANSAC and closer to HPT. For hardware evaluation, hardware experiment on FPGA indicates that proposed method is capable of supporting the real-time matching system in processing high frame rate videos (784fps, 640×480 VGA video) with an ultra-low delay speed (0.828 ms/frame).

For our future research, we will try more ways to improve the matching accuracy. Combining more powerful matching algorithm and mismatch removal is our next direction.

V. Acknowledgments

This work was supported by Waseda University Grant for Special Research Projects (2019C-581)

References

[1] Thomas. Schlmer, Benjamin. Poppinga, Niels. Henze, Susanne. Boll, "Gesture recognition with a Wii controller," *Proc. 2008 ACM 2nd Int. Conf. Tangible and embedded interaction*, pp. 11-14, 2008.

[2] Matthias. Heesen, Marc. Dziennus, Anna. Schieben, "Interaction design of automatic steering for collision avoidance: challenges and potentials of driver decoupling," *IET Intell. Transp. Syst,* Vol.9, No.1, pp. 95-104, Jan, 2014.

[3] Tingting. Hu, Takeshi. Ikenaga, "Pixel Selection and Intensity Directed Symmetry for High Frame Rate and Ultra-Low Delay Matching System," *IEICE Trans. Inf. & Syst,* Vol.E101-D, No.5, pp. 1260-1269, Feb, 2018.

[4] Yuhao. Xu, Tingting. Hu, Songlin. Du, Takeshi. Ikenaga, "Partial Descriptor Update and Isolated Point Avoidance Based Template Update for High Frame Rate and Ultra-Low Delay Deformation Matching," *Proc. IEEE Int. Conf. Pattern Recognition*, pp. 1827-1832, 2018.

[5] Martin A. Fischler, Robert C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM,* Vol.24, No.6, pp. 381-385, June, 1981.

[6] Songlin. Du, Takeshi. Ikenaga, "Hierarchical Progressive Trust Model for Mismatch Removal under both Rigid and Non-Rigid Transformations," *IEICE Trans. Fundamentals* Vol.E101-A, No.11, pp. 1786-1794, Nov, 2018.

[7] Tingting. Hu, Takeshi. Ikenaga, "FPGA Implementation of High Frame Rate and Ultra-Low Delay Vision System with Local and Global Parallel based Matching," *Proc. IAPR Int. Conf. Machine Vision Applications,* pp. 286-289, 2017.

[8] Krystian. Mikolajczyk, Cordelia. Schmid, "A performance evaluation of local descriptors," *IEEE Trans. Pattern Anal. Machine Intell,* Vol.27, No.10, pp. 1615-1630, Oct, 2005.