SUMMARY High frame rate and ultra-low delay matching system plays an increasingly important role in human-machine interactions, because it guarantees high-quality experiences for users. Existing image matching algorithms always generate mismatches which heavily weaken the performance the human-machine-interactive systems. Although many mismatch removal algorithms have been proposed, few of them achieve real-time speed with high frame rate and low delay, because of complicated arithmetic operations and iterations. This paper proposes a temporal constraints and block weighting judgement based high frame rate and ultra-low delay mismatch removal system. The proposed method is based on two temporal constraints (proposal #1 and proposal #2) to firstly find some true matches, and uses these true matches to generate block weighting (proposal #3). Proposal #1 finds out some correct matches through checking a triangle route formed by three adjacent frames. Proposal #2 further reduces mismatch risk by adding one more time of matching with opposite matching direction. Finally, proposal #3 distinguishes the unverified matches to be correct or incorrect through weighting of each block. Software experiments show that the proposed mismatch removal system achieves state-of-the-art accuracy in mismatch removal. Hardware experiments indicate that the designed image processing core successfully achieves real-time processing of 784fps VGA (640 × 480 pixels/frame) video on field programmable gate array (FPGA), with a delay of 0.858 ms/frame.

key words: high frame rate, ultra-low delay, image matching, mismatch removal, temporal constraints, block weighting judgement

1. Introduction

In recent years, high frame rate and ultra-low delay image processing system is becoming more and more important in human-machine interactive applications, such as object tracking[1], projection mapping[2], simultaneous localization and map building (SLAM)[3], and autonomous driving[4]. Hu and Ikenaga[5], [6] proposed a local feature based hardware-oriented high frame rate and ultra-low delay matching system, which is robust to translation and in-plane rotation. Based on Hu’s work, Xu et al.[7], [8] solved the problem of non-rigidity deformation through template update. However, although existing works[5]–[8] are robust to challenges such as rotation and deformation, mismatches are still existing in the matching results. In order to improve the matching performance and make matching system to be more robust, a hardware-friendly mismatch removal method needs to be designed with lower complexity calculation and parallel processing to achieve high speed and ultra-low delay.

As for mismatch removal, random sample consensus (RANSAC)[9] 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 finding the optimal fitting consensus true matches set. The advantage of RANSAC is that it is able to do robust calculation of consensus, even a significant number of mismatches are existing, the results from RANSAC still have higher accuracy. To generalize the image transformation model used in RANSAC, Li and Hu[10] proposed to learn a correspondence function for mismatch removal, and named the approach as ICF (Identifying point correspondences by Correspondence Function). The correspondence function in ICF is also estimated iteratively by using diagnostic technique and support vector machine (SVM). Another effective mismatch removal method is hierarchical progressive trust model (HPT)[11] which designs a three-layered model to expand true matches. 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 shows better performance than RANSAC and is robust for various situations. Recently, Jiang et al. [12] proposed a progressive filtering strategy through iterative optimization, and named the approach as progressive filtering for feature matching (PFFM).

All of RANSAC, ICF and HPT actually try to get globally geometrical relationship between two images, which means that they start calculation after whole images are input. But waiting for input of images increases extra delay for hardware system. Both of the two methods are iterative with no fixed iteration times. All the existing approaches, including PFFM, depend on iteration. There exists data dependency caused by iterations, which is against the 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. In addition, they use complex operations, such as division, square root, logarithm and so on, which are complicated for hard-
Aiming at high frame rate and ultra-low delay, this paper proposes a temporal constraints and block weighting judgement based high frame rate and ultra-low delay mismatch removal system. Since high frame rate video has close relations between continuous frames, the strong temporal coherence is utilized to design constraints. In the proposed system, two temporal constraints (proposal #1 and proposal #2) are designed to firstly find some true matches, and uses these true matches to generate block weighting (proposal #3). Proposal #1 finds out some correct matches through checking a triangle route formed by three adjacent frames. Proposal #2 further reduces mismatch risk by adding one more time of matching with opposite matching direction. Finally, proposal #3 distinguishes the unverified matches to be correct or incorrect through weighting of each block. Compared with conventional works[9]–[12], the proposed method has two main merits. First, the proposed method does not estimate global transformations between two frames. So global processing can be avoided to reduce the delay. Second, the proposed method performs mismatch removal at keypoint level, also avoids the large delay of global processing. The system is designed from the point of lower complexity, parallel processing, and no 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. The work described in this paper is an extension of our previous conference paper[13], with significate new contents including both new proposals and new experimental results.

The rest of this paper is organized as follows. Section 2 presents the proposed high frame rate and ultra-low delay mismatch removal system which includes three proposals. Section 3 presents the hardware structure of the proposed high frame rate and ultra-low delay mismatch removal system. Experimental results on both of software and hardware are reported and analysed in Sect. 4, followed by conclusion and future works given in Sect. 5.

2. Proposed High Frame Rate and Ultra-Low Delay Mismatch Removal System

The framework of the proposed mismatch removal for high frame rate and ultra-low delay matching system is shown in Fig. 1. The proposed method uses high frame rate video as input. The basic matching module is based on the work of Hu and Ikenaga[5], [6]. Firstly, the system uses Harris corner detector[14] to detect keypoints in each frame. Secondly, for each keypoint, a binary descriptor is generated by binary robust independent elementary features (BRIEF)[15].

The proposed mismatch removal system contains three proposals. In particular, the proposed method uses feature descriptors from each three adjacent frames as input, and the output is the true matches. Because high frame rate video has strong temporal coherence, the proposed system is based on two temporal constraints (proposal #1 and proposal #2) to firstly find some true matches, and uses these true matches to generate block weighting. Through weighting of each block, it distinguishes all the matches as true or false (proposal #3). As for the first temporal constraint, three adjacent frames are used to do brute-force matching among each other, and the matching direction is as the following order: from the second frame to the third frame, then to the first frame, and finally back to the second frame. 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 finds 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. As for the second temporal constraint, there are already matching results from the second frame to the third frame, which means that, for each keypoint descriptor in the second frame, there is a matching keypoint descriptor in the third frame. By adding one more time of matching from the second frame to the first frame, each keypoint finds a matching keypoint descriptor in the first frame. By checking the similarity of descriptors in the first frame and in the third frame, it finds more true matches. In image matching, Hamming distance is used to verify the similarity between two descriptors, so this proposal calculates the Hamming distance of the descriptors in the first frame and in the third frame and compares with a threshold to find more true matches. With true matches verified by proposal #1 and proposal #2, proposal #3 divides images into blocks and make statistics of matching frequency for each block in image as its weight. All blocks have positive weights form current matching area. And for left matches which are not verified by proposal #1 and proposal #2, if it is in the area, it has large probability to be true. So block weighting is a judgement criterion for the left matches.

2.1 Proposal #1: Temporal Constraint Based Keypoints Triangle Check

Figure 2 shows conceptual difference between conventional
mismatch removal methods [9], [11] and the proposal #1. Conventional methods use complex arithmetic operations and iteration algorithms to calculate geometrical relationship between two images. But for different images, different iteration times are needed, which cause larger hardware resource cost and higher delay for hardware system. For high frame rate video, there is extremely strong relationship between adjacent frames. When matching frames of high frame rate video, the images provide rich temporal information and potential correspondence in time dimension since matching is a continuous process. With information of temporal coherence, the proposed mismatch removal method does not construct models to get geometrical constraints between two images, so that complex mathematical operation and iteration are avoided. Only focusing on keypoint level processing, global processing delay is avoided, and parallel operation is also possible to be performed for match pairs. Because of these reasons, this proposal is hardware friendly with ultra-low delay.

If time interval between input frames is extremely small, the differences between adjacent frames are also small, and same keypoints constantly appear in at least three frames until they are disappeared due to object transformation. So proposal #1 uses each three frames as input in each time. The matching direction is from the second frame to the third frame, then to the first frame, and finally back to the second frame. After three times of matching, a matching route is formed. By checking the shape of this matching route, all matches are divided into two parts: triangle part and not triangle part. A graphical illustration is shown in Fig. 3 and Fig. 4. As can be seen from Fig. 3, for the matches from triangle part, there are only three keypoints in the matching route, which means that the three keypoints in the matching route are the same keypoint. So there is no doubt that for the matches from triangle part are true.

Because the shape of the matching route shown in Fig. 3, triangle is named for the matches in this situation. For not triangle part shown in Fig. 4, after three times of matching, the matching route is not a triangle, which means
that a new keypoint appears and at least one time of mismatch happens. Since three times of matching have been performed and mismatch may happen in any time, there are three types of situations. Firstly, if a mismatch happens between the second frame and the third frame as shown in Fig. 4(a), because the goal is to remove mismatch between the second frame and the third frame, this match should be removed. Secondly, if a mismatch happens between the third frame and the first frame as Fig. 4(b) shows, because no matching keypoint in the first frame, the match between the third frame and the first frame must be false. Because the goal is to distinguish the matches between the second frame and the third frame, and the shape of the matching route is not a triangle for this situation. Although the ground-truth of matches from this type may be true, just using proposal #1 is not able to distinguish. Thirdly, if mismatch happens between the first frame and the second frame, the match route is not a triangle, but the match between the second frame and the third frame also has a probability to be true. The ground-truth of matches from this type may be true, but proposal #1 is not able to distinguish this situation as well. Considering all these three situations, the matching routes are obviously not triangle, and only proposal #1 is not able to distinguish where mismatch happens. And if all this part is removed, it may happen that, although all matches after mismatch removal are true, removed true matches are more than false matches, which doesn’t guarantee a good performance. So proposal #2 and proposal #3 are proposed to find true matches in not triangle part.

2.2 Proposal #2: Temporal Constraint Based Hamming Distance Verification

The concept of proposal #2 is shown in Fig. 5. The matching direction of proposal #1 is from the second frame to the third frame, then to the first frame, and finally back to the second frame. As the example shown in Fig. 5(a), if mismatch happens from B to C, proposal #1 actually uses wrong keypoint C to find D, which causes the shape of matching route is not a triangle. For keypoint D, it is a separate keypoint from A. So the matching from the third frame to the first frame may also bring mismatch risk to the entire mismatch removal process. So proposal #2 adds one more time of matching with opposite matching direction from the second frame to the first frame to remove the mismatch risk from the third frame to the first frame of proposal #1 and find more true matches.

Why does the matching from the third frame to the first frame brings mismatch risk? This is because of the brute-force matching. As Fig. 6 shows, from frame A to frame B for D1 (Descriptor #1) in frame A, it goes through all descriptors of frame B and matches D’1 with minimum Hamming distance. But for D2 in frame A, it doesn’t check D2, because it has been matched with D1.

In opposite matching direction from frame B to frame A, for D’1 it also go through all descriptors of frame A, but it matches D2 with minimum Hamming distance. And for D’2, it doesn’t check D2 since D2 has been matched with D’1. So as can be seen, with brute-force match but in different matching direction, the matching results are different. As Fig. 5(b) shows, because the matching results in the third frame have already gotten, proposal #2 firstly finds the descriptor in the first frame, and then calculates the Hamming distance between the descriptors in the first frame and the third frame. Checking the Hamming distance with the threshold as shown in Fig. 7, if it is larger than the threshold, that means they are different keypoints, and mismatch does happen, the match between the second frame and the third frame are false, as sown in Fig. 7(a). If it is smaller than the threshold, that means they are the same keypoint, and all the three keypoints are the same, their matches are also true, as shown in Fig. 7(b). So proposal #2 distinguishes two types of situation and finds more true matches.

2.3 Proposal #3: Block Weighting Judgement of Matching Area Statistics

Figure 8 shows the concept of proposal #3. Because high frame rate video has strong temporal coherence, movement of each keypoint between continuous frames is not large. In
other words, keypoints keep appearing and being matched in a certain area. This area is able to be obtained by making statistics of matching frequency for each part in image and be described as weights for each part. Through updating the weights of each part with only true matches, it shows the truly matching area in image. And for matches from not triangle part, if they are in the area, they have large probability to be true matches. The detailed steps of proposal #3 are introduced as follows.

Firstly, each image is divided into blocks. The weight of each block is represented by the corresponding matching frequency. Higher weight means that the current keypoints are distributed and matched in this block. The weights of the blocks are initialized as zeros. The updating cannot start from the first frame, because the weights represent the statistics of matching frequency for blocks. Only after a certain number of frames (e.g., 200 frames in our settings) have been matched, the stable statistics of matching frequency for each block can be obtained. The process before start updating likes a warming up. After using proposal #1 and proposal #2, some true matches with strong temporal coherence have already been found. These true matches are used as update factor for weights of each block. The updating rule for the weight \( w_{ij} \) of the block \( ij \) is

\[
\begin{align*}
    w_{ij} + x n, & \quad \text{if block with } n \text{ true matches} \\
    w_{ij} - y, & \quad \text{if block with zero matches} \\
    w_{ij} \geq 0, & \quad \text{always no less than zero}
\end{align*}
\]

where the parameter \( x \) is a positive factor for true matches. The parameter \( y \) is a negative parameter for blocks with no matches. The reason of setting negative factor is illustrated through an example in Fig. 9. When the object moves down, if there is no negative factor to decrease the weight of the block without any matches, the weighting won’t reflect the truly matching area. Both of the optimal values of \( x \) and \( y \) are obtained by adjusting through several times of experiments.

From accuracy point-of-view, either small or large blocks decrease robustness. Because large blocks usually contain both of object pixels and partial background pixels, which cannot distinguish the object boundary clearly. Meanwhile, very small blocks (e.g. 1 pixel or \( 2 \times 2 \) pixels) are similar to pixel-level statistics, which cannot fully take the information of neighboring pixels into calculation, also leading to low robustness. From processing speed point-of-view, the sequential processing speed on software is of course affected by the block size. More blocks would take longer time to be sequentially processed. However, the goal of this paper is to propose a highly-parallel algorithm for hardware implementation which should guarantee high frame rate and ultra-low delay. The blocks in this paper are processed in parallel, so the processing speed is not largely-affected by the number of blocks, as they are processed simultaneously. Although the processing time is not affected by the number of blocks, one should note that more blocks takes more hardware resources for parallel processing. In this paper, taking both of robustness and hardware resources into consideration, the block size is set as \( 8 \times 8 \) pixels.

With true matches from proposal #1 and proposal #2 and accumulating for a period of time, for blocks in current
for matches in these blocks, they are directly classified as true. So, for left matches which are not distinguished as true matches by proposal #1 and proposal #2, the proposal #3 checks the block weights of these matches whether they are larger than a threshold. If it is, the match is classified as true, otherwise the match is classified as false. The threshold is determined empirically. For example shown in Fig. 10, the judgement is

\[
\begin{cases} 
\text{True match,} & \text{if } w_{22} > \text{threshold} \\
\text{False match}, & \text{otherwise}
\end{cases}
\]

(2)

3. Hardware Structure of the Mismatch Removal System

Figure 11 shows the specific hardware structure of this system. The basic data transfer module is based on\cite{6}. The video sequence taken by the high frame rate camera is output in pixels and received by the camera link receiver of FPGA. The generated descriptors are used to do matching between three continuous frames. Using the matching results to firstly do the triangle check and then Hamming distance verification, and using the results of the first two steps to update the block weights. The output matches are labelled with a flag signal to represent true or false. Finally, the data of matching results are output from the image processing core. In the control of a memory controller, the output data are written into DDR3-SDRAM by FPGA and then read from DDR3-SDRAM by PC. Besides, PC is able to communicate with FPGA via WISHBONE-BUS, adjusting relevant parameters and threshold values in the image processing core, and transmitting the initial template descriptor to FPGA.

4. Experimental Results and Analysis

4.1 Evaluation Environment

In order to analyze the feasibility and reliability of the proposed system, evaluation is conducted in both software and hardware. The exact evaluation environments are introduced as follows. Evaluation of matching accuracy after mismatch removal is tested by software programming. The state-of-the-art local feature descriptor ORB \cite{16} is utilized to generate matches. As for matching performance, the baseline ORB matching \cite{16} and two conventional works, i.e. RANSAC \cite{9} and HPT \cite{11}, are also tested for comparative study. The software evaluation environment is Visual Studio 2013 with OpenCV 3.1.0 on a PC of Windows 10 Pro operating system. Evaluation on hardware performance is conducted to evaluate the processing delay and resource utilization of the high frame rate and ultra-low delay mismatch removal system. We use Xilinx Kintex-7 XC7K325T FPGA board and BASLER acAC2000-340 camera to implement the high frame rate and ultra-low delay mismatch removal system. The system demonstration is shown in Fig. 12. The logic synthesis and implementation on FPGA are performed by Vivado v2017.2 (64-bit).

We use a dataset of high frame rate videos with seven types of challenges for experiments. Each sequence has 2000 frames with each has 640×480 pixels, and the frame rate is 762 fps. Figure 13 shows representative frames of the seven types of sequences. The seven types of challenges include 3D rotation, rotation & deformation, 2D rotation, illumination change, translation, scale change and camera

![Fig. 11](image.png)

**Fig. 11** Hardware structure of the matching system with proposed mismatch removal method, where P1, P2 and P3 are short for proposal #1, proposal #2 and proposal #3, respectively.
moving. As can be seen, the sequences of 3D rotation, 2D rotation, translation, and scale change are taken from a rigid object, and the sequence of rotation & deformation is constructed with a non-rigid object which has complex deformations during object moving. All of the seven types of sequences are tested with basic ORB matching, two conventional works RANSAC and HPT, and several types of combination among the three proposals.

The ground truth of representative frames are manually labelled for comparison. The matching performance is evaluated by the widely used metrics, including precision, recall and $f$-score [17], which are defined as

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

and

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

As the name implies, precision reflects the correct rate of the matching result, while recall reflects the number of correct matches which should be found. And the $f$-score is defined as a combination of precision and recall, i.e.

$$f\text{-score} = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}}.$$  

Higher precision and higher recall together lead to a higher $f$-score, which means the mismatch removal method finds more correct matches and loses few miss matches. So higher $f$-score reflects better matching performance.

### 4.2 Software Evaluation Results and Analyses

For each type of video sequence, we test the baseline ORB matching [16], two conventional works RANSAC [9] and HPT [11], the proposed method with several combinations of three proposals. Like RANSAC [9], ICF [10] is also based on global transformation relationship, which is not as hardware-oriented as the proposed approach. We therefore compare with well-known RANSAC [9] for simplicity. As introduced before, proposal #1 is temporal constraint based keypoints triangle check, proposal #2 is temporal constraint based Hamming distance verification, and proposal #3 is block weighting judgement of matching area statistics. Totally, seven matching methods are organized as follows:

- ORB matching [16]: baseline ORB matching without any mismatch removal methods.
- RANSAC [9]: baseline ORB matching with random sample consensus algorithm.
- HPT [11]: baseline ORB matching with hierarchical progressive trust model.
- P1: baseline ORB matching with proposal #1 (temporal constraint based keypoints triangle check).
- P1 + P3: baseline ORB matching with proposal #1 & proposal #3 (temporal constraint based keypoints triangle check and block weighting judgement of matching area statistics).
- P1 + P2: baseline ORB matching with proposal #1 & proposal #2 (temporal constraint based keypoints triangle check and Hamming distance verification).
- P1 + P2 + p3: baseline ORB matching with proposal #1 & proposal #2 & proposal #3 (temporal constraint based keypoints triangle check and Hamming distance verification, and block weighting judgement of matching area statistics).
Table 1 compares the results of the seven methods measured by average f-score. As can be seen, both of two conventional works RANSAC and HPT have better performance than baseline ORB matching, which means both of two conventional works improve the performance of baseline ORB matching and do remove mismatches. Because HPT is a three-layer model, for the block level it uses transformation function model, and for picture level it uses EM algorithm, which are more complex than just calculating the homography matrix. So it is reasonable that the performance of the proposed algorithm (including P1, P2 and P3). This is the reason why our proposal cannot outperform HPT in accuracy. On the other hand, it is potential to use the proposed method to improve HPT for both higher accuracy and faster processing speed. Using high frame rate video as input, it is possible to replace the pixel-level mismatch removal in HPT with our proposal #1 and proposal #2, and replace the block-level mismatch removal in HPT with our proposal #3. Although our proposals can be utilized to speed up HPT, it is still impossible for HPT to reach high frame rate and ultra-low delay as our system does, because of the essential iterations and global processing in the HPT algorithm. That’s the reason why the proposed system is important for its unique merits on both accuracy and speed.

Figure 14 shows software demonstration of the proposed mismatch removal method. The green lines are true matches which are not verified by proposal #1 and proposal #2. Because HPT is one of the most accurate methods. The high accuracy is also close to state-of-the-art methods. From accuracy point-of-view, although does not outperform HPT, the proposed method achieves similar f-score with HPT, meaning that the proposed method archives acceptable accuracy, because HPT is one of the most accurate methods. The high accuracy of HPT is driven from complicated calculations and hardware-unfriendly operations, such as iteration and float-number-based probability calculation. In our scenario, we cannot use these complex and hardware-unfriendly operations, because it is impossible (at least very difficult) to implement them on hardware for parallel processing. Considering our goal, we use very simple operations, such as addition, subtraction and binary shifting to design the proposed algorithm (including P1, P2 and P3). This is the reason why our proposal cannot outperform HPT in accuracy. On the other hand, it is potential to use the proposed method to improve HPT for both higher accuracy and faster processing speed. Using high frame rate video as input, it is possible to replace the pixel-level mismatch removal in HPT with our proposal #1 and proposal #2, and replace the block-level mismatch removal in HPT with our proposal #3. Although our proposals can be utilized to speed up HPT, it is still impossible for HPT to reach high frame rate and ultra-low delay as our system does, because of the essential iterations and global processing in the HPT algorithm. That’s the reason why the proposed system is important for its unique merits on both accuracy and speed.

Table 1 Matching accuracy of seven methods (f-score %).

| Sequence | ORB matching | RANSAC | HPT | P1 | P1+P3 | P1+P2 | P1 + P2 + P3 |
|----------|--------------|--------|-----|----|------|-------|-------------|
| 3D       | 86.54        | 91.01  | 94.16 | 75.11 | 92.11 | 90.55 | 93.37       |
| R & D    | 91.96        | 97.26  | 97.64 | 90.62 | 96.97 | 91.81 | 97.06       |
| 2D       | 89.76        | 94.33  | 94.77 | 79.10 | 94.51 | 93.32 | 94.51       |
| Illumine | 95.25        | 97.35  | 97.80 | 93.79 | 97.01 | 92.14 | 97.46       |
| Trans    | 84.53        | 87.42  | 91.03 | 75.50 | 88.68 | 81.71 | 90.21       |
| Scale    | 90.00        | 93.18  | 94.42 | 79.94 | 93.46 | 88.07 | 93.66       |
| Camera   | 91.74        | 95.30  | 96.51 | 88.47 | 95.41 | 93.26 | 95.72       |
| Average  | 89.97        | 93.09  | 95.19 | 83.22 | 94.02 | 90.12 | 94.56       |
matches, and the red lines are false matches. As can be seen, the green lines distribute in smooth space, and the two keypoints matched by a green line are the same keypoint on the object. And the red lines distribute in random space, and the two keypoints matched by a red line are different, which are obviously false. And the proposed method does distinguish the false matches and remove them. In final output, all the false matches are removed, and only true matches are used as output.

In summary, the evaluation on matching accuracy indicates that the proposed temporal constraints and blocking weighting judgement based mismatch removal method outperforms the conventional work RANSAC, and is close to HPT. With satisfactory matching performance, there is no complex arithmetic operation. Only addition and subtraction are used. There is also no iteration process in whole method. Meanwhile, the proposed method is much easier and more friendly for hardware implementation than RANSAC and HPT.

4.3 Hardware Evaluation Results and Analyses

After successful verification of the reliability and effectiveness of proposed methods by software implementation, we implement all three proposals into the high frame rate and ultra-low delay system with FPGA, for the purpose of evaluating their performances, especially processing speed and delay, on hardware.

Table 2 reports the resource utilization and hardware performance of the proposed mismatch removal system. The utilization of every kind of logic units is about two-thirds. That means there is still much resources to implement more functions or modules into the system. Meanwhile, the processing delay for each input frame is 0.858 ms/frame (less than 1 ms/frame), meeting the requirement of real-time matching for the purpose of high frame rate and ultra-low delay. Therefore, the proposed hardware-oriented mismatch removal algorithm is proved to be reasonable and feasible, which are capable of supporting the real-time processing of matching system with mismatch removal under input frequency and high frame rate with the resolution of 640×480 pixels. The matching performance of the temporal constraints and block weighting judgement based mismatch removal for high frame rate and ultra-low delay matching system is tested in practice. Example matching results are shown in Fig. 15. As can be seen, the target object in the input frame is successfully matched and without mismatches.

To sum up, the evaluation on hardware performance indicates that the proposed mismatch removal method is hardware friendly enough, and both the resource utilization and processing delay are suitable for the goal of high frame rate and ultra-low delay. Besides, the experiments prove that the matching system is able to perform real-time matching without mismatches.
4.4 Comparison and Discussion on Computational Time

As analyzed above, RANSAC and HPT are hardware-unfriendly because of iteration and complicated operations. It is not easy to obtain processing time of conventional works on hardware. We therefore tested their software computational time by programming with C++ language and OpenCV on a PC with 2.6GHz CPU. Experimental results show that, detecting 64 keypoints, extracting ORB features and performing brute-force matching takes about 2 seconds; RANSAC takes about 0.003 seconds to remove mismatches, but its robustness is low; HPT takes about 40 seconds to remove mismatches with high robustness. As reported in Sect. 4.3, the proposed method takes only 0.858ms to finish the mismatch removal task. From these results and Table 1, one can conclude that the proposed method achieves both high robustness and ultra-low delay.

5. Conclusion

In this paper, a temporal constraints and block weighting judgement based high frame rate and ultra-low delay mismatch removal system has been proposed. The matching result shows that the proposed method has much better performance than baseline ORB matching. Compared with conventional mismatch removal methods, the performance of the proposed method with much more complexity reduction and higher parallelism is better than the widely used RANSAC and is close to HPT. For hardware evaluation, hardware experiment on FPGA indicates that the proposed system is capable of supporting the real-time matching system in processing high frame rate videos (784fps, 640×480 pixels/frame) with an ultra-low delay speed (0.858 ms/frame). In our future research, we will consider more spatial constraints and the combination of spatial and temporal constraints for further improvement of the accuracy without dynamic increase of complexity.

Acknowledgments

This work was supported by KAKENHI (16K13006) and Waseda University Grant for Special Research Projects (2019C-581).

References

[1] P. Zhao, H. Zhu, H. Li, and T. Shibata, “A Directional-Edge-Based Real-Time Object Tracking System Employing Multiple Candidate-Location Generation,” IEEE Trans. Circuits Syst. Video Technol., vol.23, no.3, pp.503–517, March 2013.
[2] E.L. Schwartz, “Spatial mapping in the primate sensory projection: analytic structure and relevance to perception,” Biological Cybernetics, vol.25, no.4, pp.181–194, 1977.
[3] M.W.M.G. Dissanayake, P. Newman, S. Clark, H.F. Durrant-Whyte, and M. Csurka, “A solution to the simultaneous localization and map building (SLAM) problem,” IEEE Trans. Robotics and Automation, vol.17, no.3, pp.229–241, June 2001.
[4] M. Heesen, M. Dziennus, T. Hesse, A. Schieben, C. Brunken, C. Lüper, J. Kelsch, and M. Baumann, “Interaction design of automatic steering for collision avoidance: challenges and potentials of driver decoupling,” IET Intelligent Transport Systems, vol.9, no.1, pp.95–104, 2014.
[5] T. Hu and T. Ikenaga, “FPGA Implementation of High Frame Rate and Ultra-Low Delay Vision System with Local and Global Parallel based Matching,” Proc. IAPR Int. Conf. Mach. Vis. Appli., pp.286–289, 2017.
[6] T. Hu and T. 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, May 2018.
[7] Y. Xu, T. Hu, S. Du, and T. 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 Recognit., pp.1827–1832, 2018.
[8] S. Du, Y. Xu, T. Hu, and T. Ikenaga, “Adaptive-Partial Template Update with Center-Shifting Recovery for High Frame Rate and Ultra-Low Delay Deformation Matching,” IEICE Trans. Fundamentals, vol.E102-A, no.12, pp.1872–1881, Dec. 2019.
[9] M.A. Fischler and R.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, 1981.
[10] X. Li and Z. Hu, “Rejecting Mismatches by Correspondence Function,” Int. J. Comput. Vis., vol.89, no.1, pp.1–17, 2010.
[11] S. Du and T. 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.
[12] X. Jiang, J. Ma, and J. Chen, “Progressive Filtering for Feature Matching,” IEEE Int. Conf. Acoust., Speech, Signal Process., pp.2217–2221, 2019.
[13] Z. Wang, S. Du, and T. Ikenaga, “Temporal coherence based mismatch removal for high frame rate and ultra-low delay system,” Proc. Int. Workshop on Smart Info-Media Systems in Asia, 2019.
[14] C. Harris and M. Stephens, “A combined corner and edge detector,” Proc. Alvey Vision Conf., pp.147–151, 1988.
[15] M. Calonder, V. Lepetit, C. Strecha, and P. Fua, “Brief: Binary robust independent elementary features,” Proc. European Conf. Computer Vision, pp.778–792, 2010.
[16] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, “ORB: An efficient alternative to SIFT or SURF,” Proc. Int. Conf. Comput. Vis., 2011.
[17] K. Mikolajczyk and C. Schmid, “A performance evaluation of local descriptors,” IEEE Trans. Pattern Anal. Mach. Intell., vol.27, no.10, pp.1615–1630, Oct. 2005.

Songlin Du received the Ph.D. degree from the Graduate School of Information, Production and Systems, Waseda University, Kitakyushu, Japan. He is currently with the School of Automation, Southeast University, Nanjing, China. His research interests include visual feature representation and related hardware implementation. He received the Best Paper Award at IS-PACS2017. He is a member of the IEEE.
Zhe Wang received the B.E. degree in Computer Science and Technology from Beihang University, China, in 2017, and the M.E. degree from Graduate School of Information, Production and Systems, Waseda University, Japan, in 2019. His research focuses on FPGA implementation of computer vision related algorithms.

Takeshi Ikenaga received his B.E. and M.E. degrees in electrical engineering and Ph.D. degree in information & computer science from Waseda University, Tokyo, Japan, in 1988, 1990, and 2002, respectively. He joined LSI Laboratories, Nippon Telegraph and Telephone Corporation (NTT) in 1990, where he had been undertaking research on the design and test methodologies for high performance ASICs, a real-time MPEG2 encoder chip set, and a highly parallel LSI & system design for image understanding processing. He is presently a professor in the system integration field of the Graduate School of Information, Production and Systems, Waseda University. His current interests are image and video processing systems, which covers video compression (e.g. H.265/HEVC, SHVC, SCC), video filter (e.g. super resolution, noise reduction, high-dynamic range imaging), and video recognition (e.g. sport analysis, feature point detection, object tracking). He is a senior member of the Institute of Electrical and Electronics Engineers (IEEE), a member of the Institute of Electronics, Information and Communication Engineers of Japan (IEICE) and the Information Processing Society of Japan (IPSJ).