Real-time motion estimation for image and video processing applications

Guillermo Botella · Carlos García

Published online: 12 December 2014
© Springer-Verlag Berlin Heidelberg 2014

Abstract This contribution focuses on different topics that are covered by the special issue titled “Real-Time Motion Estimation for image and video processing applications” and which incorporate GPUS, FPGAs, VLSI systems, DSPs, and Multicores, among other platforms. The guest editors have solicited original contributions, which address a wide range of theoretical and practical issues related to high-performance motion estimation image processing including, but not limited to: real-time matching motion estimation systems, real-time energy-based motion estimation systems, gradient-based motion estimation systems, optical flow estimation systems, color motion estimation systems, multi-scale motion estimation systems, optical flow and motion estimation systems, analysis or comparison of specialized architectures for motion estimation systems and real-world applications.

Keywords Motion estimation · Optical flow · Image processing · Video coding · FPGA · GPGPU · DSP · ASIC · Embedded systems · ASIC

1 Introduction

The robustness of the human visual system regarding motion estimation in almost any visual situation is enviable, as it performs enormous calculation tasks continuously, robustly, efficiently and effortlessly. Motion

estimation from image sequences has been deeply analyzed by the scientific community due to its importance in a huge variety of visual tasks.

In a general image, the 3D scene is projected onto the image plane such that each point produces a 2D path with an instantaneous vector velocity. The 2D velocities for all visible surface points are usually denominated as a 2D motion field. The final aim of optical flow estimation is to compute an approximation to the motion field from time-varying image intensity. Several different real-time approaches to motion estimation have been proposed [1–9], and these could preliminarily be classified as belonging to matching domain approximations [10], energy models [11] and gradient models [12, 13]. Despite the number of different models and algorithms [14], none of them covers all the problems associated with real-world processing, such as noise, illumination changes, second order motion, occlusions, etc.

Techniques for estimating optical flow often seek a compromise between accuracy and efficiency [15]. This compromise arises because the most precise techniques tend to have higher computational requirements. Given similar computational resources, some techniques make more precise estimations; albeit more slowly, while others obtain less accurate movement calculations, but faster. The accuracy allows assessment of the quality of the results obtained while the efficiency refers to both the time in which the data are obtained, as well as to the computational resources used for it. In the motion estimation scenario, it is desirable to obtain accurate results efficiently in real-time, using techniques that allow their adaptation to real problems. For this purpose, platforms related to motion estimation accelerators, including those which make use of graphics hardware as specific circuits, and FPGAs are discussed.

G. Botella (✉) · C. García
Department of Computer Architecture and Automation,
Complutense University of Madrid, Ciudad Universitaria s/n,
28040 Madrid, Spain
e-mail: gbotella@ucm.es

1.1 Hardware implementation

One of the types of accelerators which are used for motion estimation is the FPGA (Field Programmable Gate Array), which can basically be described as a reprogrammable silicon chip which is made of pre-built logic blocks and can be programmed thanks to routing resources. An FPGA can be configured to implement custom hardware functionality without soldering or using a breadboard. An FPGA contains millions of connections and logic cells that can be configured to achieve a specific digital logic design. Instead of being restricted to any predetermined hardware function, an FPGA makes it possible to program product features and functions, adapt to new standards, and reconfigure hardware for specific applications, even after the product has been installed in the field, hence the name “field-programmable”.

FPGAs can be programmed in a large variety of low-level and high-level HDL (Hardware Description Languages) [16]. Due to the configurable nature of FPGA devices, a customized piece of hardware can be designed to be included in any sensor. It is possible to design processor features, develop specialized hardware accelerators for intensive computational tasks, and create custom input/output ports to be connected to other physical parts of the sensor. These systems, built together on the same FPGA, are nowadays known as SoPCs (System on a Programmable Chip). FPGAs are configured by developing digital computing tasks in software and compiling them down to a configuration file or bit-stream that contains information on how the components should be wired together. Thanks to these features, FPGAs can be completely reconfigured, and are able to adapt to a new design by only recompiling a different circuitry configuration.

GPUs are another recent development based on a customized processor primarily for graphics rendering, and initially designed for the entertainment market. Nowadays, GPUs have a large number of processors which can be used for general purpose computing. Supercomputers that currently lead the world ranking combine the use of a large number of CPUs with a high number of GPUs. The GPU is especially appropriate for solving computationally intensive problems which can be expressed as parallel data computations. However, implementation on a GPU requires the redesign of the algorithms, focused and adapted to its architecture by taking into account two key characteristics: the complex memory hierarchy optimization used, and the efficient mapping of the problem to be parallelized into blocks of threads, and threads themselves. Programming GPUs automatically involves considering a number of constraints, such as the need for high occupancy in each processor to hide latencies produced by management, the synchronization of different threads running

simultaneously, memory access, the proper use of the hierarchy of memories as mentioned above, among others. Researchers have already successfully applied GPU computing to problems which were traditionally addressed by the CPU.

Regarding motion estimation using embedded micro-processors, the wearable and mobile industry has recently increased the importance of the processing hardware elements. As the market becomes more demanding, many processor manufacturers have specialized in their own solutions, such as Snapdragon, Cortex A8, Tegra, OMAP, ARM11 and many more. Today, device manufacturers equip smartphones with the capacity of high-end computers that are able to perform multimedia coding and transmission tasks, and meet scientific processing requirements [17, 18]. Additionally, there are several teaching approaches for designing specific signal processing microprocessors [19] using architecture description language which makes feasible this paradigm. The ARM instruction set has become the reference architecture in low-power devices, so there are many general CPUs able to run ARM-compatible code; this fact creates a tendency in reference processors to be compatible with these instruction sets, and so the companies produce the chipset following the ARM directives.

1.2 Algorithmic optimization

Motion estimation is a mathematically complex problem which many times is ill-posed and suffers from non-exact solutions. Therefore, it is necessary to adopt many approaches and assumptions. For instance, in the context of block-matching algorithms, the task to perform is to estimate motion vectors for the current frame comparing the current macro-block with each macro-block within a specific and fixed search window in the reference frame [20, 21]. Doing this in the most simple approach, one would run an exhaustive search algorithm, matching all macro-blocks within a search window in the reference frame to estimate the optimal macro-block; i.e., the one with the minimum BME (block-matching error). There are several definitions for BME, but the most commonly used are SAD (sum of absolute differences), for every pixel between an MB (macro-block) in the current frame and an MB in the reference frame, and MSE (mean squared error), which is less conservative due to the square factor. Then, the huge number of computations required to calculate the error by these algorithms is too high, so to optimize motion estimation calculation, many enhanced search algorithms have been proposed. These methods can be divided into two main families: SR (search reduction), which covers techniques based on reducing the number of search points within a search window following the pointed algorithm,

which in most cases moves or changes the search window to improve its accuracy [22–25]; and CR (calculation reduction), which includes algorithms categorized as CR of SAD that are aimed at reducing the number of computations. Since SAD is calculated by adding the differences of each pixel, the computation of the partial SAD is simpler than the computation of the total SAD between two MBs [26]. The idea is thus to reject the invalid motion vectors early by means of not comparing all the macroblocks available.

1.3 Special issue: real-time motion estimation for image and video processing applications

The goal of this special issue has been to bring together researchers and practitioners working in the area of real-time motion estimation for image and video processing applications and using GPUS, FPGAs, VLSI systems, DSPs, and Multicores, among other platforms. The rapid technological developments and advances in recent years are evident. After the call for papers for this special issue, 27 dedicated manuscripts were received, 13 of which were considered to be of sufficiently high quality to be selected in a peer review process conducted by prestigious scientists with a high level of expertise in the different fields covered. Development, tests, and reviews make important contributions to today's information society, which demands the creation, distribution, and processing of information for different purposes, offering some improvements from both the economic point of view, including competitiveness and productivity, as well as for different aspects of human life.

There are still many challenges and problems to be faced, but the references provided in this special issue make a significant step forward in this regard, facilitating subsequent scientific and technological developments for industrial applications. Some considerations about future trends are also provided, on the basis of the evolution of society and the new services required.

2 Specific contributions

Concerning the topics covered in this SI, four papers address the important aspect of multimedia coding and transmission, four papers perform real-time implementations of motion estimation algorithms regarding matching, energy and gradient families, and at last, three papers deal specifically with tracking and navigation. Finally, we close this SI with two visionary papers regarding, on the one hand, intellectual property protection of motion estimation codes on embedded platforms, and on the other, real-time velocity measurement in photogrammetry and computer vision techniques.

2.1 In the context of multimedia coding and transmission

- Ismail et al. [27] present a novel Full Search Motion Estimation co-processor architecture design which reuses search area data to minimize memory I/O while fully utilizing the hardware resources. A smart processing element (PE) and an efficient simple internal memory are the main components of the proposed coprocessor, and as a result, the speed of the co-processor is improved in terms of the throughput and the operating frequency compared with state-of-the-art techniques. The proposed architecture is implemented using both the FPGA and the ASIC flow design tools. For a search range of 32×32 and block size of 16×16 , the architecture can perform motion estimation for 30 fps of HDTV video at 350 MHz and easily outperforms many fast full search architectures.
- Paramkusam et al. [28] propose a new fast search motion algorithm with multi-layer motion estimation (MME) which reduces the computational complexity of each distortion measure instead of reducing the number of search points. A hierarchical quad-tree structure is employed in this paper to construct multiple layers from the reference frame. The effectiveness of the proposed MME algorithm is compared with that of some state-of-the-art fast block matching algorithms with respect to speed and motion prediction quality. Experimental results for a wide variety of video sequences show that the proposed algorithm outperforms the other popular conventional fast search motion estimation algorithms computationally while maintaining the motion prediction quality very close to that of the full-search algorithm. The proposed algorithm can achieve a maximum speed improvement rate of 97.99 % against the fast full-search motion estimation algorithms which are based on the hierarchical block matching process.
- Pastuszak et al. [29] present the architecture of a high-throughput compensator and interpolator used in the motion estimation of the H.265/HEVC encoder. The architecture can process 8×8 blocks in each clock cycle. The design allows a random order of checked coding blocks and motion vectors. This feature makes the architecture suitable for different search algorithms. Synthesis results show that the design can operate at 200 and 400 MHz when implemented on FPGA Arria II and TSMC 90 nm, respectively. The computational scalability enables the proposed architecture to trade the throughput for compression efficiency. If 2160p@30fps video is encoded, the design clocked at 400 MHz can check about 100 motion vectors for 8×8 blocks.

- Finally, the fourth paper selected in the field of multimedia coding and transmission is that of Belghith et al. [30], who speed up the encoding process using efficient algorithms based on fast mode decision and optimized motion estimation. The aim is to reduce the complexity of the motion estimation algorithm by modifying its search pattern, combining it subsequently with a new fast mode decision algorithm to further improve coding efficiency. Experimental results show a significant speedup in terms of encoding time and bit-rate saving with tolerable quality degradation. In fact, the proposed algorithm enables a considerable reduction that can reach up to 75 % in encoding time. This improvement is accompanied with an average PSNR loss of 0.12 dB and a decrease of 0.5 % in terms of bit-rate.

2.2 In the context of real-time implementation of motion estimation algorithms

- Ferreira et al. [31] present a very interesting study regarding the block matching motion estimation of a remotely operated vehicle (ROV). A challenging environment such as being under water is an excellent test bed to evaluate the performance of the various recently developed feature extractors and descriptors. The algorithms were tested using the same open source framework to enable a fair assessment of their performance, especially in terms of computational time. The various possible combinations of algorithms were compared with an approach developed by the authors that showed good performance in the past. A data set collected by the ROV Romeo in typical operations is used to test the methods. Quantitative results in terms of robustness to noise and computational time are presented and demonstrate that the recent trend of binary features is very promising.
- Additionally, Plyer et al. [32] deal with dense optical flow estimation from the perspective of the tradeoff between the quality of the estimated flow and computational cost which is required by real-world applications. They propose a fast and robust local method, denoted by eFOLKI, and describe its implementation on a GPU. It leads to very high performance even on large image formats such as 4 K ($3,840 \times 2,160$) resolution. To assess the usefulness of eFOLKI, they first present a comparative study with currently available GPU codes, including local and global methods, on a large set of data with ground truth. eFOLKI appears significantly faster while providing quite accurate and highly robust estimated flows. They then show, on four real-time video processing applications based

on optical flow, that eFOLKI meets the requirements both in terms of estimated flow quality and of processing rate.

- Another paper, written by García-Rodríguez et al. [33], describes a neural-network-based architecture that represents and estimates object motion in videos. This architecture addresses multiple computer vision tasks such as image segmentation, object representation or characterization, motion analysis and tracking. The use of neural network architecture allows for the simultaneous estimation of global and local motion and the representation of deformable objects. This architecture also avoids the problem of finding corresponding features while tracking moving objects. Due to the parallel nature of neural networks, the architecture has been implemented on GPUs, which allows the system to meet a set of requirements such as time constraints management, robustness, high processing speed and re-configurability. Experiments are presented that demonstrate the validity of the architecture in solving problems of mobile agent tracking and motion analysis.
- Finally, in the field of real-time implementation of motion estimation systems, Tomasi et al. [34] present a co-processing architecture using FPGA and DSP. A portable platform for motion estimation based on sparse feature point detection and tracking is developed for real-time embedded systems and smart video sensor applications. A Harris corner detection IP core is designed with a customized fine grain pipeline on a Virtex-4 FPGA. The detected feature points are then tracked using the Lucas–Kanade algorithm in a DSP that acts as a co-processor for the FPGA. The hybrid system offers a throughput of 160 frames per second for VGA image resolution. They also tested the benefits of the proposed solution (FPGA + DSP) in comparison with two other traditional architectures and co-processing strategies: hybrid ARM + DSP and DSP only. The proposed FPGA + DSP system offers a speedup of about 20 times and 3 times with respect to ARM + DSP and DSP only configurations, respectively. A comparison of the Harris feature detection algorithm performance between different embedded processors (DSP, ARM, and FPGA) reveals that the DSP offers the best performance when scaling up from QVGA to VGA resolutions.

2.3 In the context of real-time tracking and navigation systems

- Migniot et al. [35] address the problem of 3D tracking of human gesture for buying behavior estimation. The top view of the customers, which has not received

much attention in human tracking, is exploited in this particular context. This point of view avoids occlusion except for that of the arms. They propose a hybrid 3D-2D tracking method based on the particle filtering framework, which uses the exclusion principle to separate the observation related to each customer and deals with multi-person tracking. The head and shoulders are tracked in 2D space, while the arms are tracked in 3D space, resulting in a greater descriptive capability and a faster processing time. A thorough quantitative and qualitative experimental analysis, including on-site experiments, is reported and discussed. The results obtained demonstrate that good estimations can be achieved for various cases and situations in real time (~ 40 fps).

- Manzanera et al. [36] tackle two crucial aspects of general-purpose embedded visual point tracking. First, the algorithm should reliably track as many points as possible. Second, the computation should achieve real-time video processing, which is challenging on low power embedded platforms. This paper proposes a new multi-scale semi-dense point tracker called Video Extruder, whose purpose is to fill the gap between short-term, dense motion estimation (optical flow) and long-term, sparse salient point tracking. This paper presents a new detector, including a new salience function with low computational complexity and a new selection strategy that makes it possible to obtain a large number of key points. Thanks to its high degree of parallelism, the proposed algorithm extracts beams of trajectories from the video very efficiently. They compare it with the state-of-the-art pyramidal Lucas–Kanade point tracker and show that, in short-range mobile video scenarios, it yields similar quality results, while being up to one order of magnitude faster. Three different parallel implementations of this tracker are presented, on multicore CPU, GPU and ARM SoCs. On a commodity 2010 CPU, it can track 8 500 points in a 640×480 video at 150 Hz.
- Finally, Nguyen et al. [37] address the need for small light-weight vehicles, such as unmanned ground or air vehicles, to sense their own motion for use in autonomous navigation algorithms. As the processing is ideally performed on-board these vehicles, there are severe restrictions on the processing environment available to perform the optical flow calculations. This has led to the development of FPGA solutions to calculate optical flow. However, the most recent approaches still have extensive on-board memory requirements and make use of complex processing operations such as multiplication and matrix inversion. The authors present an FPGA implementation of a low complexity version of the Lucas–Kanade registration

algorithm. This algorithm operates on one-bit images instead of the standard eight-bit approach and consequently can utilize simple logic operations such as exclusive-or rather than multiplications, and also makes very efficient use of the available internal memory and resources.

2.4 In the context of visionary papers regarding real-time motion estimation systems

- Meyer et al. [38] address the intellectual property protection of multimedia coding for embedded systems, focusing on primitives widely used for motion estimation algorithms. Motion estimation is extensively used in multimedia tasks, video coding standards and home consumer devices, and many FFT-based motion algorithms appear in this field. Furthermore, the intellectual properties of embedded microprocessor systems are typically delivered on HDL and C source code levels. Obfuscating the code is most often the only way to protect and avoid reverse engineering. This paper presents an evaluation of operations widely used for protection purposes in motion estimation for an embedded microprocessor. A set of open source obfuscation tools has been developed that allows the use of very long and hard-to-read identifiers. The implementation of comment methods also allows for the addition of copyright and limited warranty information. The obfuscated code with identifiers of up to 2,048 characters in length is tested for Altera's and Xilinx's FPGAs for a typical HDL example. Compiler penalties as well as FFT runtime results are reported.
- Finally, Almeida et al. [39] presents a methodology and all the procedures used to validate the real-time velocity measurement of the linear motion of a rigid object, executed in a physics laboratory under controlled and known conditions. The validation was based on analyses of registered data in an image sequence and the measurements obtained by high precision sensors. This methodology was intended to measure the velocity of a rigid object in linear motion with the use of an image sequence acquired by a commercial digital video camera. The proposed methodology does not need a stereo pair of images to calculate the object's position in 3D space: it only needs the image sequence obtained by one monocular vision. To do so, these objects need to be detected while in movement, which is achieved by the application of a segmentation technique based on the temporal average values of each pixel registered in N consecutive image frames. The system is low-cost-based and capable of operating in real time.

3 Conclusion

This overview has covered the contributions of accepted manuscripts as well as the topics for the special issue “Real-Time Motion Estimation for Image and Video Processing Applications” in the *Journal on Real-Time Image Processing* (JRTIP). The guest editor hopes that the selected papers will provide the readers with interesting examples of present research on algorithms and architectures for real-time motion estimation. Thanks are due to all the authors for their valuable contributions to this special issue, and to the reviewers for all their comments and suggestions to improve the quality of the accepted papers. Thanks again to Prof. Nasser Kehtarnavaz from the University of Texas in Dallas (USA), and Dr. Matthias F. Carlsohn from Engineering and Consultancy for Computer Vision & Image Communication, Bremen, Germany, for all the help they gave in managing the special issue. Also thanks to all the staff at JRTIP, who offered invaluable support for the organization of this issue.

The Guest Editors wishes to acknowledge the support provided by the Spanish Ministry of Science and Innovation through project TIN 2012-32180.

References

1. Ayuso, F., Botella, G., García, C., Prieto, M., Tirado, F.: GPU-based acceleration of bioinspired motion estimation model, in concurrency and computation: practice and experience, vol. 25, pp. 1037–1056. Wiley, New York (2013)
2. Botella, G., García, A., Rodríguez-Alvarez, M., Ros, E., Meyer-Bäse, U., Molina, M.C.: Robust bioinspired architecture for optical-flow computation. *IEEE Trans. VLSI Syst.* **18**(4), 616–629 (2010)
3. García, C., Botella, G., Ayuso, F., Prieto, M., Tirado, M.: Multi-GPU based on multicriteria optimization for motion estimation system. *EURASIP J. Adv. Signal Processing* **2013**, 23 (2013)
4. Botella, G., Meyer-Bäse, U., García, A., Rodríguez, M.: Quantization analysis and enhancement of a VLSI gradient-based motion estimation architecture. *Digit. Signal Process.* **22**(6), 1174–1187 (2012)
5. Botella, G., Ros, E., Rodríguez, M., García, A., Romero, S.: Pre-processor for bioinspired optical flow models: a customizable hardware implementation. In: Proceedings of the 13th IEEE Mediterranean Electrotechnical Conference, MELECON 2006, Benalmádena (Benalmádena, 2006), pp. 93–96
6. González, D., Botella, G., Meyer-Bäse, U., García, C., Sanz, C., Prieto-Matías, M., Tirado, F.: A Low cost matching motion estimation sensor based on the NIOS II microprocessor. *Sensors* **12**, 13126–13149 (2012)
7. Igual, F.D., Botella, G., García, C., Prieto, M., Tirado, F.: Robust motion estimation on a low-power multi-core DSP. *EURASIP J. Adv. Signal Process.* **2013**, 99 (2013)
8. González, D., Botella, G., García, C., Prieto, M., Tirado, F.: Acceleration of block matching algorithms using a custom instruction-based paradigm on a Nios II microprocessor. *EURASIP J. Adv. Signal Process.* **2013**, 118 (2013)
9. Botella, G., Martín, H.J., Santos, M., Meyer-Bäse, U.: FPGA-based multimodal embedded sensor system integrating low-and mid-level vision. *Sensors* (Basel, Switzerland) **11**(8), 8164 (2011)
10. Oh, H., Lee, H.: Block-matching algorithm based on an adaptive reduction of the search area for motion estimation. *Real Time Imaging* **6**, 407–414 (2000)
11. Huang, C., Chen, Y.: Motion estimation method using a 3D steerable filter. *Image Vis. Comput.* **13**(1), 21–32 (1995)
12. Baker, S., Matthews, I.: Lucas-Kanade 20 years on: a unifying framework. *Int. J. Comput. Vision* **56**(3), 221–255 (2004)
13. Lucas, B.D., Kanade, T.: An iterative image registration technique with an application to stereo vision. In: Proceedings of the Seventh International Joint Conference on Artificial Intelligence, pp. 674–679 (1981)
14. Szelinsky, R.: Computer vision algorithms and applications. Springer, Berlin (2011)
15. Liu, Hongche, Hong, Tsai-Hong, Herman, Martin, Camus, Ted, Chellappa, Rama: Accuracy vs efficiency trade-offs in optical flow algorithms. *Comput. Vis. Image Underst.* **72**(3), 271–286 (1998)
16. Bayley, D.: Design for embedded image processing on FPGAs. Chapter IV Languages. Wiley-IEEE Press eBook Chapters, pp. 73–78 (2011)
17. Seal, D.: ARM architecture reference manual, 2nd edn. Addison-Wesley, Boston (2001)
18. [In-line] <http://www.forbes.com/sites/darcytravlos/2013/02/28/arm-holdings-and-qualcomm-the-winners-in-mobile/>
19. Meyer-Bäse, U., Botella, G., Castillo, E., Garcia, A.: A balanced HW/SW teaching approach for embedded microprocessors. *Int. J. Eng. Educat.* **26**(3), 584–592 (2010)
20. Konrad, J.: Estimating motion in image sequences. *IEEE Signal Process. Mag.* **16**, 70–91 (1999)
21. Sohm, O.P.: Fast DCT algorithm for DSP with VLIW architecture. U.S. Patent 20,070,078,921. 5 April 2007
22. Kappagantula, S., Rao, K.-R.: Motion compensated interframes image prediction. *IEEE Trans. Commun.* **33**, 1011–1015 (1985)
23. Kuo, C.J., Yeh, C.H., Odeh, S.F.: Polynomial search algorithms for motion estimation. In: Proceedings of the 1999 IEEE International Symposium on Circuits and Systems, pp. 813–818. Orlando (2012)
24. Zhu, S., Ma, K.-K.: A new diamond search algorithm for fast block-matching motion estimation. *IEEE Trans. Image Process.* **9**, 287–290 (2000)
25. Zhu, S.: Fast motion estimation algorithms for video coding. MS thesis. Nanyang Technology University, Singapore (1998)
26. Bei, C.-D., Gray, R.-M.: An improvement of the minimum distortion encoding algorithm for vector quantization. *IEEE Trans. Commun.* **33**, 1132–1133 (1985)
27. Ismail, Y., El-Medany, W., Al-Junaid, H., Abdelgawad, A.: High performance architecture for real-time HDTV Broadcasting. *J. Real Time Image Process.* (2014). doi:[10.1007/s11554-014-0430-1](https://doi.org/10.1007/s11554-014-0430-1)
28. Paramkusam, A.V., Reddy, V.S.: An efficient multi-layer reference frame motion estimation for video coding. *J. Real Time Image Process.* (2014). doi:[10.1007/s11554-014-0431-0](https://doi.org/10.1007/s11554-014-0431-0)
29. Pastuszak, G., Trochimiuk, M.: Architecture design of the high-throughput compensator and interpolator for the H.265/HEVC encoder. *J. Real Time Image Process.* (2014). doi:[10.1007/s11554-014-0422-1](https://doi.org/10.1007/s11554-014-0422-1)
30. Belghith, F., Kibeya, H., Loukil, H., Ali Ben Ayed, M., Masmoudi, N.: A new fast motion estimation algorithm using fast mode decision for high efficiency video coding standard. *J. Real Time Image Process.* (2014). doi:[10.1007/s11554-014-0407-0](https://doi.org/10.1007/s11554-014-0407-0)
31. Ferreira, F., Veruggio, G., Caccia, M., Bruzzone, G.: A survey on real-time motion estimation techniques for underwater robots. *J. Real-Time Image Proc.* (2014). doi:[10.1007/s11554-014-0416-z](https://doi.org/10.1007/s11554-014-0416-z)

32. Plyer, A., Le Besnerais, G., Champagnat, F.: Massively parallel Lucas Kanade optical flow for real-time video processing applications. *J. Real Time Image Process.* (2014). doi:[10.1007/s11554-014-0423-0](https://doi.org/10.1007/s11554-014-0423-0)
33. García Rodríguez, J., Orts-Escolano, S., Angelopoulou, A., Psarrou, A., Azorín-López, J., García Chamizo, J.M.: Real time motion estimation using a neural architecture implemented on GPUs. *J. Real Time Image Process.* (2014). doi:[10.1007/s11554-014-0417-y](https://doi.org/10.1007/s11554-014-0417-y)
34. Tomassi, M., Pundlik, S., Luo, G.: FPGA-DSP co-processing for feature tracking in smart video sensors. *J. Real Time Image Process.* (2014). doi:[10.1007/s11554-014-0413-2](https://doi.org/10.1007/s11554-014-0413-2)
35. Migniot, C., Ababsa, F.: Hybrid 3D-2D human tracking in a top view. *J. Real Time Image Process.* (2014). doi:[10.1007/s11554-014-0429-7](https://doi.org/10.1007/s11554-014-0429-7)
36. Manzanera, A., Garrigues, M., Bernard, T.M.: Video extruder: a semi-dense point tracker for extracting beams of trajectories in real time. *J. Real Time Image Process.* (2014). doi:[10.1007/s11554-014-0415-0](https://doi.org/10.1007/s11554-014-0415-0)
37. Nguyen, A.H., Pickering, M., Lambert, A.: The FPGA implementation of a one-bit-per-pixel image registration algorithm. *J. Real Time Image Process.* (2014). doi:[10.1007/s11554-014-0420-3](https://doi.org/10.1007/s11554-014-0420-3)
38. Meyer Baese, U., Meyer-Baese, A., Gonzalez, D., Botella, G., García, C., Prieto-Matías, M.: Code obfuscation using very long identifiers for FFT motion estimation models in embedded processors. *J. Real Time Image Process.* (2014). doi:[10.1007/s11554-014-0421-2](https://doi.org/10.1007/s11554-014-0421-2)
39. Almeida, N., Filitto, D., Kiyoshi, J., Polidório, A.M., Flores, F.C.: Real time velocity measurement to linear motion of a rigid object with monocular image sequence analyses. *J. Real-Time Image Process.* (2014). doi:[10.1007/s11554-014-0472-4](https://doi.org/10.1007/s11554-014-0472-4)