Journal of Real-Time Image Processing (2018) 15:705–707
https://doi.org/10.1007/s11554-018-0839-z

EDITORIAL

Journal of Real-Time Image Processing: fourth issue of volume 15

Matthias F. Carlsohn[1] · **Nasser Kehtarnavaz**[2]

Published online: 19 November 2018

The Journal of Real-Time Image Processing (JRTIP) began its operation in 2006 with 4 issues annually and 80 pages per issue. Since then the technological advances in hardware and new computing paradigms have accelerated the real-time implementation of various image and video processing solutions that were not possible before leading to their actual utilization in real-world products. After 13 years, we are confident to state that JRTIP has now become a comprehensive repository of real-time image and video processing scholarly contributions.

Although the page budget has been increased a number of times since the inception of the journal with the current page count of 200 pages per issue, there has been a steady increase in the flow of submitted manuscripts as original research papers dedicated to the real-time aspects of image and video processing. As a result, the back-log of online first articles (that is, articles accessible and citable online) has been growing and causing delays in the print versions of the articles. In addition to regular submissions, the journal has been offering special issues on focused image and video processing topics within the real-time framework, generating unique sets of articles solely dedicated to real-time aspects of the focused topics on image and video processing. This in turn has contributed to the growth of the back-log.

To address the back-log in a fundamental manner, starting in 2019 with volume 16, it has been decided to increase the number of issues to six per year with 250 pages per issue, making the new annual print page budget 1500 pages per year. It is expected that this major increase in the number of pages will dramatically reduce the delays between online first and print versions of accepted articles. It is worth pointing out that guest editors interested in offering special issues should take into consideration the new 250 page count per issue as this increase in the page count translates into approximately 20–25 accepted articles, which is about 5–10 articles more than the numbers previously making up special issues. We highly encourage guest editors to contact potential authors first for their commitment to submit manuscripts but also to serve as peer reviewers of special issues in order to ensure sufficient numbers of submissions are received and also to make the review process efficient time wise.

All those who are interested in supporting and shaping the real-time image and video processing community by their contributions as guest editors of a special issue within their particular field of expertise are requested to read the "Guidelines For Offering And Managing Special Issues As Guest Editors of JRTIP" at https://static.springer.com/sgw/documents/1488308/application/pdf/SpecialIssueGuidelines_final.pdf and become familiar with the format and typical content before sending proposals to the editors-in-chief. Examples of currently open calls for special issues can be viewed at https://www.springer.com/computer/image+processing/journal/11554/PSE?detailsPage=societies.

Finally, we refer readers to the backmatter of this issue for the technical program of the SPIE Conference on Real-Time Image Processing and Deep Learning that is to be held in April 2019 in Baltimore as part of the SPIE Symposium on Defense and Commercial Sensing. The JRTIP's Editorial Board will be meeting during this conference and the outcome of the meeting to be reported in a future editorial.

This regular issue closes volume 15 with 7 original research papers bringing the total of number of published pages to 1600 in 2018.

The first paper entitled "A fast algorithm for integrating connected-component labeling and Euler number computation" by Lifeng He et al. proposes a fast algorithm for integrating connected-component labeling and Euler number computation. Based on graph theory, the Euler number of a binary image is calculated by counting the occurrences of four patterns for processing foreground pixels in a first scan of a connected-component labeling process, where

✉ Matthias F. Carlsohn
Matthias.carlsohn@t-online.de

Nasser Kehtarnavaz
kehtar@utdallas.edu

[1] Engineering and Consultancy for Computer Vision and Image Communication, Bremen, Germany

[2] University of Texas at Dallas, Richardson, TX, USA

these four patterns can be found without any additional calculation; thus, connected-component labeling and Euler number computation can be integrated efficiently without processing background pixels as demonstrated by experimental results.

The second paper on "Real-time correction of panoramic images using hyperbolic Möbius transformations" by Luis Peñaranda et al. presents a technique to improve the perceptual quality of panorama visualization. It provides a more natural view by addressing artifacts like bent, which has an increasing effect with increasing field of view. The solution considers the viewing sphere as a Riemann sphere, which enables the application of Möbius (complex) transformations to the input image together with a projection scheme that changes depending on the field of view. Additionally, an implementation of the method is presented and compared to imaging results of competing methods, and demonstrating the real-time performance of the transformations making the technique attractive for new settings but also for existing interactive panoramic applications.

The third article by Hamid Bazargani et al. describes "A fast and robust homography scheme for real-time planar target detection" deals with the problem of robust pose estimation for planar targets in the context of real-time mobile vision. It involves low computational costs by employing feature-based methods based on local binary descriptors that allow fast feature matching at run time. The matching set is fed into a robust parameter estimation algorithm to obtain a reliable estimate of a 2D homographic transformation being the essential part of the entire recognition process. This paper presents an optimized and device-friendly implementation of homography estimation through a unified hypothesize-and-verify framework designed for the purpose of fast and robust estimation on power-constrained platforms. It combines simultaneously the run-time efficiency of the recognition software and their optimized implementation tuned to the computational performance of contemporary CPUs leading to speed-up factors of 25 compared to the regular RANSAC homography estimation function in OpenCV.

The fourth article on "Component interconnect and data access interface for embedded vision applications" by Michael Mefenza et al. proposes a novel interface, the Component Interconnect and Data Access (CIDA), and its implementation based on the interface automata formalism that can be used to capture system-on-chip architecture with the primary focus on video processing applications involving a data streaming paradigm with occasional direct memory accesses. The notion of component-interface clustering, targeting resource reduction and a method to automatize this process are discussed in the paper. The discussed approach can reduce the resource usage (#slices) on average by 20% and reduce power consumption by 5% compared to implementations based on off-the-shelf interfaces in video processing applications implemented on FPGAs.

In the fifth paper Jan Bartovský et al. address a fully programmable "Morphological co-processing unit for embedded devices" to deal with the problem that most morphological operations are composed of a (potentially large) number of sequential elementary operators. The demand for robustness and decision liability of industrial applications challenges the resource limitations of embedded systems, especially under real-time constraints answered by a morphological co-processing unit (MCPU) enabling efficient dilation/erosion unit with geodesic units and ALUs to support a larger collection of morphological operations, from simple dilation to serial filters involving a geodesic reconstruction step. The coprocessor is implemented on a FPGA platform running a server that is able to respond to client's requests over the Ethernet. Experiments demonstrate the performance gained by the MCPU for a wider set of operations in the order of a magnitude better than competing embedded platforms based on ARM A9 quad-core processors.

In the sixth paper, Jie Jiang et al. present "An FPGA implementation for real-time edge detection" adopting the Hessian matrix-based edge detection algorithm from Steger because of its accuracy and versatility despite its computational complexity and large-scale Gaussian convolution corresponding to a large number of multipliers in a FPGA implementation. This problem is addressed by a mix of pipeline and parallel architectures at both task and data levels for data stream processing where the original kernels of Gaussian convolution are simplified with box-filters to simplify the multiplication operation of the convolutions by additions, subtractions, or shifts using the concept of an integral image, thereby minimizing the multiplier resources. The proposed FPGA implementation demonstrates a favorable accuracy and anti-noise capability when dealing with different degrees of blur and noise in an image while satisfying real-time requirements for edge detection.

The seventh paper on "Object tracking by mean shift and radial basis function neural networks" by Vahid Rowghanian et al. combines and links two independent trackers. The mean shift algorithm estimates the target's location within two iterations. Scale and orientation of the target are computed by 2-D correlation coefficients between reference and target histograms. A code optimization strategy (multiply–add–accumulate: MAC) removes unnecessary memory occupation and program operations reducing the computational load and speeds-up the tracking process, accordingly. The second tracker "RBFNN" uses as input a vector feature composed of: local contrast, color histogram, gradient,

intensity, and spatial frequency. The neural network is trained to color and texture features of the target and the background; the information is used to detect and track the object in subsequent frames. The neural network employs Epanechnikov activation functions and extracted features are clustered by fuzzy C-means and variances. Experiments demonstrate robustness of the tracker against occlusions, sudden movements, and shape deformations.