Real-time Image Registration with Region Matching

Charles Beumier and Xavier Neyt
Signal and Image Centre, Royal Military Academy, Avenue de la Renaissance 30, 1000 Brussels, Belgium

Keywords: Image Registration, Region Segmentation, Connected Component Labeling, Region Descriptors.

Abstract: Image registration, the task of aligning two images, is a fundamental operation for applications like image stitching or image comparison. In our project in surveillance for route clearance operations, a drone will be used to detect suspicious people and vehicles. This paper presents an approach for real-time image alignment of video images acquired by a moving camera. The high correlation between successive images allows for relatively simple algorithms. We considered region segmentation as an alternative to the more classical corner or interest point detectors and evaluated the appropriateness of connected component labeling with a connectivity defined by the gray-level similarity between neighboring pixels. Real-time processing is intended thanks to a very fast segment-based (as opposed to pixel-based) connected component labeling. The regions, even if not always pleasing the human eye, proved stable enough to be linked across images by trivial features such as the area and the centroid. The vector shifts between matching regions were filtered and modeled by an affine transform. The paper discusses the execution time obtained by this feasibility study for all the steps needed for image registration and indicates the planned improvements to achieve real-time.

SCITEPRESS
SCIENCE AND TECHNOLOGY PUBLICATIONS

1 INTRODUCTION

Image registration, a very important field of image processing and computer vision, is the task of aligning pictures, a fundamental step for applications like image stitching, medical image alignment or camera motion compensation.

Images are usually registered by intensity-based matching or feature-based pairing (Zitova and Flusser, 2003; Goshtasby, 2005). The common intensity-based matching consists in image patch cross-correlation to find corresponding areas in both images, a time consuming process due to the large space of search (image dimension and transform parameters). The feature-based approach consists in extracting in both images remarkable points, lines or contours and in pairing them. The small memory need and computational load of the latter approaches have given rise to many successful and efficient methods like SIFT (Lowe, 2004) and ORB (Rublee et al., 2011).

We are currently active in a European Defence Agency project of the Research and Technology programme IEDDET for countering Improvised Explosive Devices (EDA, 2017). It addresses the topic of future route clearance operations for which an early warning phase is in charge of pre-screening the area to highlight any suspicious presence of people or vehicles. To realize this, a test area will be flown over by an Unmanned Aerial Vehicle equipped with visible and thermal infra-red cameras. The thermal camera has been selected for its capacity to detect individuals and vehicles thanks to its temperature sensitivity while the visible camera is more appropriate for image registration.

For image registration, we propose to match uniform regions as an alternative to the more classical corners or interest points. Due to the similarity of images taken from a sequence, regions can provide for several simple and robust features, obtained with little development and for small computational effort. They can bring geometrical and radiometric information or mix local (contour) and regional characteristics. They also represent a useful description for object tracking, after image registration.

Real-time responses in the context of security or rapid processing in the case of automatic detection in hours of video footage impose fast algorithms. For the sake of estimating local shift between images to be registered, most fast approaches detect interest points and match them across images (Lowe,

Beumier, C. and Neyt, X.
Real-time Image Registration with Region Matching.
DOI: 10.5220/0006655304390445
In *Proceedings of the 13th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2018) - Volume 4: VISAPP*, pages 439-445
ISBN: 978-989-758-290-5
Copyright © 2018 by SCITEPRESS – Science and Technology Publications, Lda. All rights reserved

2004; Rublee et al., 2011). Many works preferred to optimize the image intensity comparison of local areas (blocks). For instance (Puglisi and Battiato, 2011) relied on efficient integral projections while (Kim et al., 2008) limited the number of blocks and sub-blocks to be analyzed, and estimated the best correlation from the number of matching edge points in sub-blocks. A more recent trend for acceleration consists in exploiting parallel computing from the central or graphics processing unit (Zhi et al., 2016; Shamonin et al., 2014). In this work we planned to explore the region approach in terms of speed, registration potential and code simplicity.

The rest of the paper first outlines the methodology in section 2, then details how images are segmented into regions in section 3, and how these are matched in order to model the image transform for registration, subject of section 4. Registration results are presented in Section 5 and time figures are discussed for this feasibility study and for the planned developments with the suggested improvements. Section 6 draws conclusions and outlines our future work.

2 METHODOLOGY

We were motivated to show that for image registration, region extraction and matching is a valid alternative to the traditional feature-based approaches in terms of speed and precision, and this for a software implementation easy to code and control.

Our development is based on the segmentation of images into regions thanks to a very fast detection of connected components. Instead of considering pixel connectivity, horizontal segments are first detected thanks to a fast horizontal connectivity check. Then the vertical connectivity is used to link segments. The representation of regions exploits directly the segments and is coded as a list of segment leftmost and rightmost x coordinates. This representation allows for memory compactness and very fast computation of classical geometrical features.

With such a speed for region segmentation, the difficulty for choosing a threshold can be alleviated by testing several threshold values for the reference image (done once) and for the images to be registered. The number of regions can be used as selection criterion but some applications may prefer to use all detected regions (for all thresholds tested). In this feasibility study, only one threshold was necessary, due to the high correlation of images taken from a short sequence.

The regions extracted in images are matched by features so that provisional shift vectors (Dx,Dy) are collected all over the image. These vectors are filtered and modeled by an affine transform. This image transform made of 6 coefficients is used to align the image to the reference so that image differencing can highlight objects in motion.

3 IMAGE SEGMENTATION

The segmentation of images follows the approach of connected component labeling, with a connectivity rule based on the gray-level difference of neighboring pixels. The implementation employs an efficient representation of regions by segments to offer speed and to optimize memory accesses and size.

3.1 Connected Component Labeling

Connected Component Labeling, the process of assigning a unique label to each group of connected pixels, is usually applied to binary images. Refer to (Grana et al., 2010) and (Lacassagne and Zavidovique, 2011) for a detailed review of pioneering and recent approaches.

Most algorithms use a 2-pass procedure that first finds connected pixels and marks them with a provisional label, storing possible equivalence of labels when branches with different labels meet. They then scan the image a second time to give a final label, result of equivalence resolution.

The improvements brought to this general approach concern the way the equivalence of labels is resolved, how memory accesses are optimized to reduce memory cache misses and how much conditional statements are minimized to avoid stalling the processing pipeline in RISC computers.

One of the fastest published methods on RISC architectures is called LSL (Light Speed Labeling, Lacassagne and Zavidovique, 2011) and consists in the storage of foreground regions (in a binary image) as run length codes (RLC) and not as an image. It is exactly the way we improved our pixel-based segmentation by connected component. The very good results and thorough evaluation of LSL make us confident that once our development for segment-based (RLC) image segmentation will be finalized, it will offer a fast and valid solution, as preliminary tests already showed.

3.2 Pixel Connectivity

Two pixels are considered connected if they touch (in 4- or 8-Neighbor connectivity) and if their gray-level difference complies to some rule. We adopted a constant threshold. This definition for connectivity allows regions to climb or descend hills of limited slopes to form large areas that are bordered by edges with a minimum contrast.

The choice for the threshold is crucial to avoid a myriad of useless small regions or a reduced set of very large areas. For speed reasons, we did not choose for an adaptive solution with varying threshold, such as the Maximally Stable Extremal Regions (Matas et al., 2002). Region growing methods usually perform non-contiguous memory accesses that may result in cache misses. Instead, we observed that 256-level images are well segmented with a fixed threshold value between 2 to 8, depending on the edge strengths. Values 3 or 4 are often appropriate values.

One good threshold value can be obtained automatically from a rough estimation of the gradient histogram. Alternatively our fast segmentation algorithm can be run several times to select the best threshold when matching two images, or even to use all obtained regions (for all thresholds) if more candidates are needed. Mention that the images in the sequences are captured within a short time interval and from a similar point of view. A threshold good for one image is likely to be fine for the others.

3.3 Region Detection

As soon as two pixels are connected horizontally, a segment is initiated by storing the first x position into the array of segments xT. The x position of the last horizontally connected pixel of this segment is stored in the next value of xT. The array xT is filled progressively during the image scan from top to bottom. Thanks to the increasing addresses of the accesses to the image and xT, memory cache misses are minimized.

At the beginning of each row during the scan process, the index of the first free value in xT is stored in a small table yT that contains h (image height) elements. This table offers a simple way to access the segments of any image line and in particular the line preceding the currently processed one. yT also gives a compact and inexpensive way to keep the y position of a segment without explicitly storing y values for each segment.

3.4 Region Labeling

Subsection 3.3 explained horizontal connectivity. The vertical connectivity is checked with stored segments (xT) of the previous image line. Again, memory accesses are efficient as xT values of the previous line are probably still in the cache. As shown in Figure 1, a new segment S may link segments with different labels Li, when for instance two or more branches get connected. This calls for label equivalence and its resolution.

All segments of the first image line receive a unique label assigned in increasing order. From the second line, a comparison is made between segment ends of the current line and the previous one to see if a label can be propagated. Since xT values are increasing along each image line, the comparison between segments of two consecutive lines is done efficiently. For a label to be propagated from segment L on line y-1 to segment S freshly detected on line y, there must exist at least one pixel from L touching one pixel of S, with a gray-level difference under the threshold.

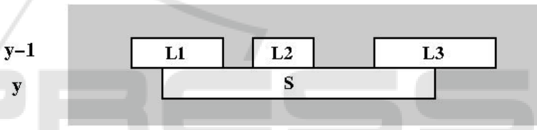

Figure 1: Segment labeling for new segment S with equivalences for L1, L2 and L3.

SCITEPRESS SCIENCE AND TECHNOLOGY PUBLICATIONS

Several label propagation cases may happen. If there is no segment L touching S, a new (increasing) label is given to S. If there is just one, its label is assigned to S. If there are several segments L, all the corresponding labels Li have to be connected in an equivalence table.

The equivalence table contains the provisional final label (called parent) for each label. Each table entry (label) is initialized with its table index. Once equivalences are found, the minimum value (so, the oldest assigned one) of the parent labels of all labels connected by segment S is used as new parent label for all connected labels.

At the end of the image scan, all segments are found and compactly stored in the xT array and easily accessed line by line thanks to the yT array. A label array called labT (indexed by simplicity the same way as xT, or half its index to gain some memory) contains the segment provisional label values. To resolve equivalences, the table values are replaced by their parent label and compacted since non-parent labels become useless. labT values are updated accordingly so that at the end, the remaining regions have the minimum number of labels from 1 (0 is reserved for no_region) to the number of regions, by order of appearance when scanning the image.

Figure 3 shows the result of image segmentation into regions for two images of the sequence separated by 4 seconds of Figure 2.

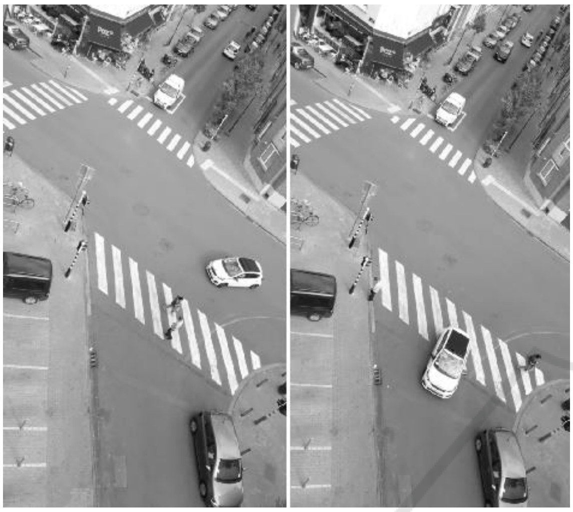

Figure 2: Two images of a sequence separated by 4 seconds.

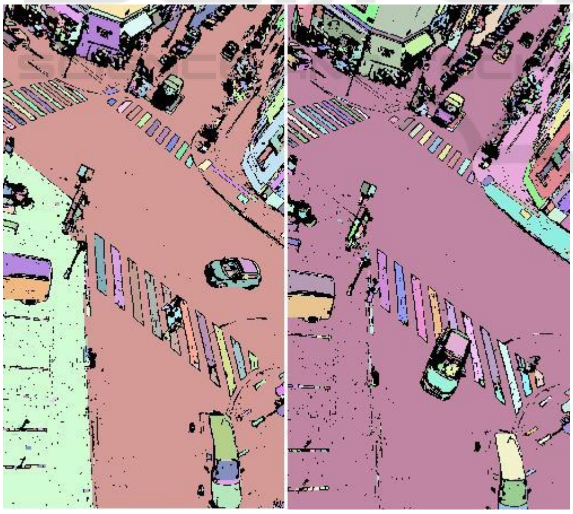

Figure 3: Region extraction and labeling for the reference image and the image to register.

4 IMAGE REGISTRATION

Image registration is realized by a 4-step procedure. First, features are extracted for the regions detected during image segmentation. Secondly, region features of an image pair are compared to identify possible matches. Each match defines a shift vector (Dx,Dy), probable displacement of a region. Thirdly, shift values are used to fit an affine transform modeling the local shift all over the image. Finally, the image to register is warped by the affine transform to be aligned to the reference.

4.1 Region Features

Several region features are easily and efficiently extracted from the way regions are stored as a collection of segments. The most direct feature is the area in pixels, computed very quickly for all regions by scanning once xT, and summing the segment lengths for each region. Region x and y value averages, also accelerated by the segment-oriented representation with xT and yT, give the centroid coordinates Cx, Cy and offer a robust localization for regions.

Like the first order moment Cx and Cy, the 2nd order moments Mxx, Mxy, Myy, physically related to inertia, can be efficiently computed. They also directly lead to the maximum and minimum inertia axes, and give a hint to the region orientation. Other easy geometrical features are the bounding box and the region contour, with possible corner detection. These last features should be included when regions are not numerous or when the centroids are not sufficiently precise, usually for medium or large size regions.

Aside from these geometrical characteristics, some obvious radiometric values can be rapidly evaluated (e.g. minimal and maximal gray values, average, standard deviation).

4.2 Region Matching

In this feasibility study, we implemented feature matching by a direct comparison of only two features (area, centroid position) with quite a large tolerance. The first image of a sequence is taken as reference to register any of the following images, one at a time.

Two regions of similar area (up to 10% difference) constitute a matching pair if their centroid lies within a distance D, by default set to 1/10 of the image largest dimension.

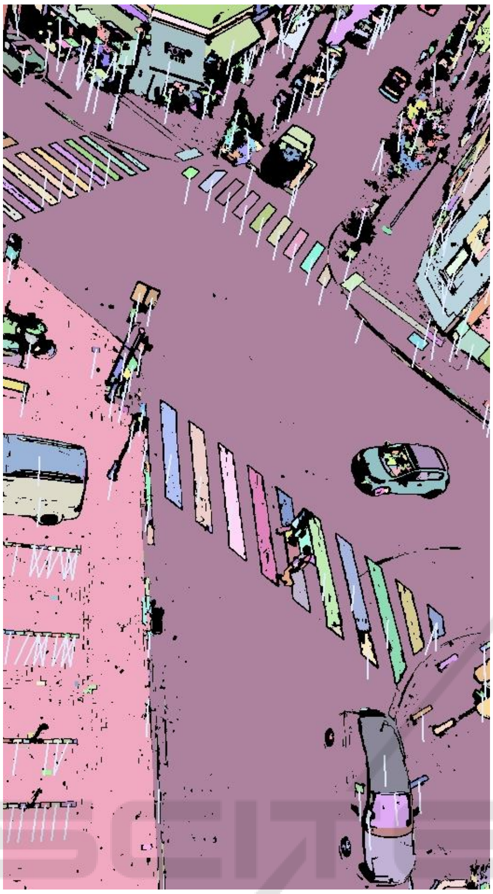

Figure 4: Selected shift vectors.

The shift vectors (Dx,Dy) between matching regions centroids are collected to later derive a motion vector field. Even with the current elementary region matching with two features, a dominant peak clearly appears in the 2-D histogram of (Dx,Dy). The distribution around this peak corresponds to the dependence of the local shift values with image position since the camera movement may induce a perspective transformation or rotation. In our tests, the false candidates due to wrong matches were distributed sparsely in the histogram and did not challenge the dominant peak. The selected shift vectors after histogram peak selection are shown in Figure 4. A majority of vectors are coherent in size and direction.

We will have to evaluate in practice, for limited movements corresponding to fast frame rates (10 or 25 Hz), and depending on drone motion patterns, if we need to consider multiple peaks, for instance for the case of a strong rotation. One possible implementation then consists in dividing the frames into tiles in which the local apparent motion is closer to a translation, resulting in a dominant peak if there are enough matching regions in the tile and few moving objects.

If the precision of Dx or Dy from the centroids is not sufficient, other points may be searched for, either from the region contours, or from the gradient peaks near region borders.

4.3 Shift Modeling

The candidate list of (Dx,Dy) values was restricted in the previous subsection to the histogram peak since the area feature (and the maximal centroid distance D) was not constraining enough to filter out most of the false matches. To further fight against erroneous shift estimations but also to compensate for the possible lack of shift values in some image area and to capture the dependence of shift values with image position, a global model for (Dx,Dy) is looked for in terms of the image coordinates. We opted for an affine transform:

$$X = Ax+By+C \quad (1)$$

$$Y = Dx+Ey+F \quad (2)$$

where x,y are the coordinates of the image to be registered and X,Y are the reference image coordinates.

The coefficients of (1) and (2) are currently estimated by least mean squares with the function getAffineTransform from the openCV library. As this function is called from our C program with a process launching Python, shift modeling represents a slow step in the current implementation of this feasibility study.

4.4 Image Warping

An image warping operation is applied to register an image of the sequence to the reference image. This operation typically scans the result frame to write the bilinear interpolation of 4 pixels from the source surrounding the coordinates projected by the inverse transformation of equations (1) and (2).

Although easy in concept, this operation is slow (40 msec for a 2 Mpixel image) since all image pixels are considered.

5 RESULTS AND DISCUSSION

The main goal of the presented research is to offer camera motion compensation. Figure 5 shows the difference between a registered image and the reference. We see that the correction is globally fine.

A residual error of 2 or 3 pixels exists in some areas. This is mainly due to the approximated localization of regions by their centroid. An approach based on region contours would be more precise but is not necessarily needed as for the detection of large and fast moving objects.

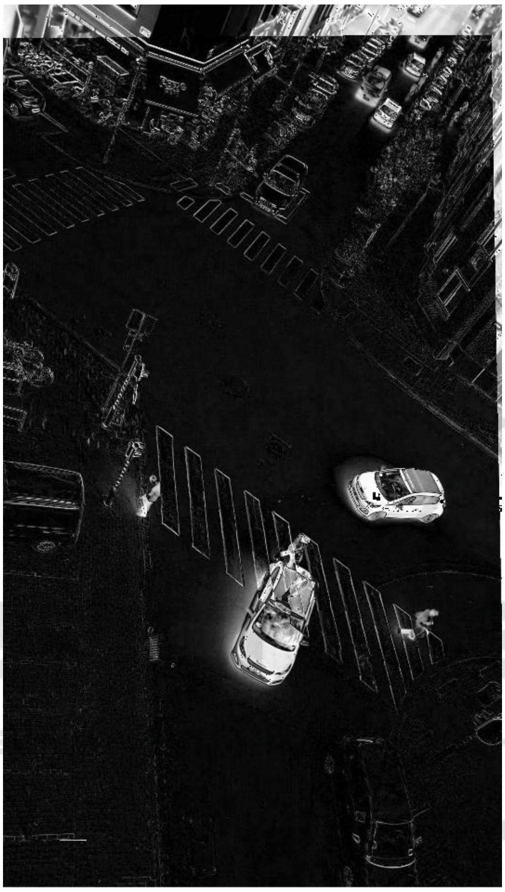

Figure 5: Difference between the reference and the registered images.

A second objective of our development is to offer fast processing. We intend to analyze the video flow in real-time or to process stored sequences as fast as possible. This is less of a challenge with nowadays computers, but the standard image resolution has increased.

For this feasibility study we recorded sequences with a Samsung A5 (2016) in the MPEG 1080p format. Each image has 1920 x 1080 pixels (2 Mpixel). The given execution times were obtained by a computer equipped with an Intel i5-4590 at 3.3 GHz (27 Gb of RAM), using a single core.

Table 1 gives an overview of the current and prospected execution time for the different steps of the proposed registration approach in the case of 2 Mpixel images.

Table 1: Timing figures for a 2 Mpixel image.

Processing	Current [ms]	Prospected [ms]
(Pre-processing)	(20)	(20)
Segmentation	140	15
Region Features	1	5
Region Matching	2	10
Shift Modeling	100	10
Image Warping	40	20
Total	**283 (+20)**	**60 (+20)**

Some pre-processing might be needed, for instance in the case of noisy images. We have indicated an optional time of 20 ms to account for simple low-pass filtering or equivalent processing.

Our implementation for this feasibility study used a pixel-based region segmentation that runs in about 140 ms. The segment-based version, not yet finalized, currently detect similar regions in less than 15 ms. This impressive timing is comparable to published works about connected component labeling from binary images (Grana et al., 2010), considering that gray-level comparison needs extra work. Only the regions with a pixel count in the range of 50 to 5000 pixels were kept. For the considered sequence, this represents more than 500 regions.

The computation of features used in section 4 (area and centroid) is really fast (less than 1 ms) thanks to the storage of regions as a list of segments. We will explore additional features to increase the region discriminative power. Some extra time has been foreseen in Table 1 for possibly more computationally demanding features.

Feature matching is also very fast (about 2 ms in our tests). About 3000 matching candidates were reduced to roughly 200 ones by the histogram peak selection. The impact on time for increasing the number of features is quite difficult to estimate since more discrimination will speedup histogram processing.

The estimation of the affine transform is a bottleneck in the current implementation because it relies on a Python library called as a separate process from a C program. About 100 ms are required to find the model coefficients thanks to roughly 200 vectors (Dx,Dy), from which about half will be rejected during refinement. Due to the large proportion of valid region pairs, the solution can benefit in execution time from a RANSAC procedure (Fischler and Bolles, 1981). From preliminary tests we believe in a 10 times speedup compared to the current implementation.

The current warping operation by the affine transform is also a heavy step (about 40 ms), since all pixels are processed and require the access of 4 neighbors for bilinear interpolation. A possible speedup for motion detection applications consists in warping first at a lower resolution, and/or with the nearest neighbor pixel, and to apply warping at full resolution only where differences with the reference are significant at low resolution.

According to Table 1, if we target an application with 2 Mpixel image sequences, 60 ms (or 80 with pre-processing) are likely to be needed for all the processing steps. At a rate of 10 images per second, 40 ms (or 20) are left to handle moving object detection and tracking, a task possibly helped by the available regions extracted for image registration.

6 CONCLUSIONS

We presented a feasibility study for real-time image registration that exploits fast image segmentation into regions based on pixel connectivity along and across horizontal segments. These segments form a compact representation of the regions, appropriate for the fast extraction of classical features such as the area, the centroids and the 2nd order moments.

According to preliminary tests, video sequences of 2 Mpixel images can be registered at 3 Hz. Based on the discussion about identified slow operations, the same sequences are likely to be registered and analyzed for object tracking at 10 Hz.

Some refinements and improvements mentioned in the discussion of section 5 are our future concern. We will first finalize the segment-based region extraction algorithm. We will then analyze the potential of additional region features and adapt region matching accordingly. We will look for another model fitting algorithm, directly callable from C. And finally, we will test other sequences, and evaluate the influence of parameters.

ACKNOWLEDGEMENTS

We would like to thank the Belgian MoD and in particular the Royal Higher Institute for Defence for supporting this research.

REFERENCES

Zitova, B., and Flusser, J. (2003). Image registration methods: a survey. *Image Vision Computing 21*, pages 977-1000.

Goshtasby, A. (2005). 2-D and 3-D Image Registration, for Medical, Remote Sensing and Industrial Applications. *Wiley Press*.

Lowe, D. (2004). Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision*, 60 (2):91-110.

Rublee, E., Rabaud, V., Konolige, K., and Bradski, G. (2011). ORB: an efficient alternative to SIFT or SURF. *In IEEE International Conference on Computer Vision (ICCV)*, pages 2564-2571.

EDA, (2017). EDA programme launched to improve IED Detection. https://www.eda.europa.eu/info-hub/press-centre/latest-news/2017/01/12.

Puglisi, G., and Battiato, S. (2011). A Robust Image Alignment Algorithm for Video Stabilization Purposes. *IEEE Transactions on Circuits and Systems for Video Technology*, 21 (10):1390-1400.

Kim, N.-J., Lee, H.-J., and Lee, J.-B. (2008). Probabilistic Global Motion Estimation Based on Laplcian Two-Bit Plane Matching for Fast Digital Image Stabilization. *EURASIP Journal on Advances in Signal Processing*, Volume 2008, pages 1-10.

Zhi, X., Yan, J., Hang, Y., and Wang, S. (2016). Realization of CUDA-based real-time registration and target localization for high-resolution video images. *Journal of Real-Time Image Processing*, May 2016, pages 1-12.

Shamonin, D., Bron, E., Lelieveldt, B., Smits, M., Klein, S., and Staring, M. (2014). Fast parallel image registration on CPU and GPU for diagnostic classification of Alzheimer's disease. *Frontiers in Neuroinformatics*, Vol 7.

Grana, C., Borghesani, D., and Cucchiara, R. (2010). Optimized Block-based Connected Components Labeling with Decision Trees. *IEEE Transactions on Image Processing*, 19(6):1596-1609.

Lacassagne, L., and Zavidovique, B., (2011). Light Speed Labeling. *Journal of Real-Time Image Processing*, 6(2):117-135.

Matas, J., Chum, O., Urban, M., and Pajdla, T. (2002). Robust wide baseline stereo from maximally stable extremal regions. *In British Machine Vision Conference 2002*, pages 384-393.

Fischler, M., and Bolles, R. (1981). Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381-395.

33545395.395.

SCITEPRESS SCIENCE AND TECHNOLOGY PUBLICATIONS