Real-Time Stitching of Panoramic Video on Intel Edison IoT Platform

Mikhail Burov
Saint Petersburg State University of
Aerospace Instrumentation (SUAI)
Saint Petersburg, Russia
m.burov@vu.spb.ru

Anton Sergeev
Saint Petersburg State University of
Aerospace Instrumentation (SUAI)
Saint Petersburg, Russia
slaros@vu.spb.ru

***Abstract*—The paper describes a system enabling real-time stitching of panoramic video on Internet-of-Things (IoT) micro-platform Intel Edison. Usually creating panoramas in real-time requires high-performance computers. This work shows that nowadays it is possible to perform real-time panoramic video even on small sized IoT devices with very low power consumption.**

I. Introduction

For today the creation of a panoramic video is usually applied inside the entertainment industry, while this technology can find much more applications. The main restriction to the popularisation lies in the duration of the process of creating such content. The captured data sets are sending to the processing centre where on high-performance computers panorama videos are created. The number of solutions for doing this in real-time is still very limited. Such a system can be adopted in the applications, where an instant response is required. For example, in the entertainment industry (video games etc,), surveillance and military applications.

The aim of this work was to make of a system for creating panoramic videos (captured from 2 or more cameras simultaniously) in real-time with acceptable FPS (>10). The criterion of real-time is the delay between obtaining the raw set of frames and result panorama. We assume that real-time is considered a system that provides an average delay (time between event happening and showing the resultant panoramic picture) less than 100 ms.

II. Overview of Interesting Results in Real-Time Stitching of Panoramas

Several similar works should be mentioned. Group of researchers of Graz University of Technology (Austria) implemented and presented very effective method for the real-time creation and tracking of panoramic maps on mobile phones in 2010 [1]. They announces creation of 1920x240 panoramic pictures (6 images 320x240) using Asus P565 smartphone (and embedded camera) rotation. CPU was XScale ARM 800 MHz (vs Intel Atom 500 MHz of Intel Edison in our experiments). The authors does not build panoramic video (because they use 1 camera rotation) but present and estimate an effective algorithm that theoretically do this at 10 FPS. The work [2] is very close to ours but uses Raspberry Pi 2 IoT platform, show a little bit better resolution (400x800) and much worse FPS (1-3 vs 22 in this work).

The criterion of real-time is the delay between obtaining the raw set of frames and result panorama. We assume that real-time is considered a system that provides a delay less than 100 ms.

III. Panorama Creation

If we have two frames, video panorama can be taken by projective plane transformation of one frame relative another [3][4]. The required deformation of the frame describes by homography matrix. To obtain this matrix, it is necessary to find common points between pairs of images.

As the hardware part used IoT platform Intel Edison, which has a very compact size (less that 3x3 cm), although gives high performance (Intel Atom 500 MHz, 2 cores) and has low power consumption (less than 1W). Edison was connected with two web cameras Microsoft HD3000, which reaches a 1280x720p resolution and allow to obtain up to 30 frames per second. The camera was attached (Fig. 1) on a wooden bar by a metal plate.

Fig. 1. Prototype

The initial idea was to use computer vision library OpenCV 3.1. It contains a class *Stitcher* designed for panoramas creation and even on the desktop shows very low FPS. The slowest part of the algorithm (Fig. 2) is the search for common points between pairs of images.

ISSN 2305-7254

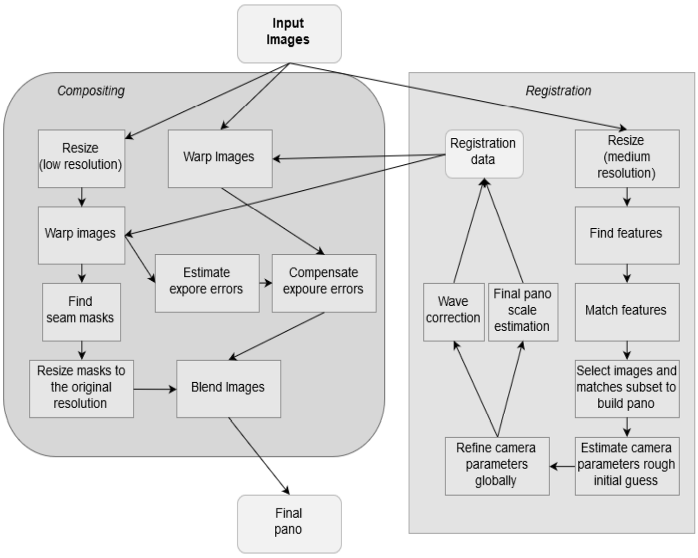

Fig. 2. OpenCV panorama creation algorithm

IV. SPEED IMPROVEMENT

To improve the speed, we can fix the cameras. Now it will be enough to find key points using SIFT or ORB and calculate the homography matrix just once. It will be applicable for each subsequent pair of the obtained images. The experiments show that this obvious solution results in enormous complexity decreasing and significant gain in speed.

Algorithm 1 Panoramas video creation

- **Initialization**

0: Obtain a set of images

1: Find image descriptors

2: Find key points between pairs of images

3: Build the homography matrix h.

- **Creating panoramas**

1: Capture next available set of frames

2: Do projective transformation using matrix h

3: Merge into a single panorama image

4: Repeat steps 1 – 4

Mongoose web-server with implemented mjpg-stream allow observing the result of real-time stitching in any browser.

For first, measurements of system performance were taken on Quad-core Intel Core i7-4702MQ with a clock frequency of 2.20 GHz. At SD 720p resolution, it shows on average 17 FPS.

Intel Edison achieved 22 FPS for video 200x200 pixels. For bigger frame size the results were much worse. This can be explained by poor (in comparison with i7) computational resources of Ediso's Atom CPU and low bandwidth of USB 2.0. As it was shown above a simple modification of the algorithm (see section III) highly reduces the time spent on data processing.

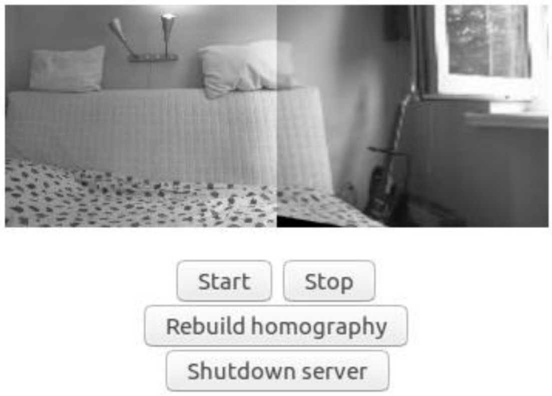

Fig. 3. Web interface with panoramic video (captured and stitched from 2 cameras)

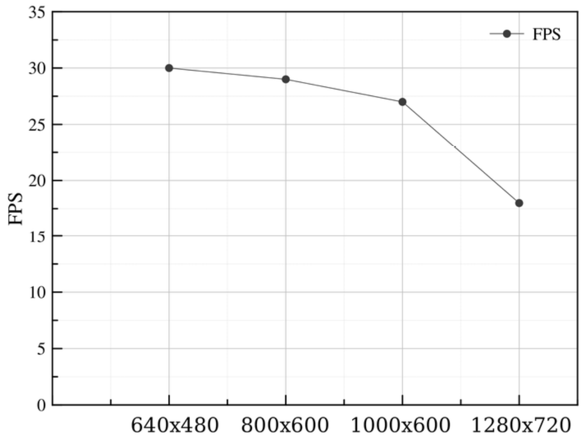

Fig. 4. Performance on Intel Core i7-4702MQ

V. CONCLUSION

We hope that in our future work we can improve this result by using more effective image stitching algorithms and optimizing data flow form camera to IoT platform. It's also planned to estimate the accuracy of the approach (mapping errors in degrees etc.) and apply transcoding methods described in [5] and [6] to improve the compression/quality ratio of the resultant video and mean transmission delay .

REFERENCES

[1] Daniel Wagner etc al, *Real-time panoramic mapping and tracking on mobile phones,* VR '10 Proceedings of the 2010 IEEE Virtual Reality Conference, Pages 211-218, 2010

[2] Adrian Rosebrock, Real-time panorama and image stitching with OpenCV, Web: http://www.pyimagesearch.com/2016/01/25/real-time-panorama-and-image-stitching-with-opencv/

[3] Elan Dubrofsky. *Homography Estimation.* Carleton University, 2007.

[4] Matthew Brown and David G. Lowe. *Automatic Panoramic Image Stitching using Invariant Features.* Department of Computer Science, University of British Columbia, Vancouver, Canada.

[5] Anna Ukhanova, Anton Sergeev, Soren Forchhammer, Extending JPEG-LS for low-complexity scalable video coding, in proc. of the IS&T/SPIE Electronic Imaging 2011, San Francisco, US, 2011.

[6] Dmitrii Malichenko, Eugine Krouk, Estimation of the Mean Message Delay for Transport Coding, *Intelligent Interactive Multimedia Systems and Services*, pp.239-249, DOI: 10.1007/978-3-319-19830-9_22, 2015