Detection and Identification of Rare Audiovisual Cues

Insperata accident magis saepe quam quae speres.
(Things you do not expect happen more often than
things you do expect) Plautus (ca 200 (B.C.))

Project no: 027787

# DIRAC

## Detection and Identification of Rare Audio-visual Cues

Integrated Project
IST – Priority 2

DELIVERABLE NO: D3.8
Camera Tracking and
Autocalibration for Detecting and
Correcting Camera De-Calibration

Date of deliverable: 31.12.2009
Actual submission date: 04.02.2010

Start date of project: 01.01.2006 Duration: 60 months

*Organization name of lead contractor for this deliverable*:
**Czech Technical University (CTU)**

Revision [0]

| Project co-funded by the European Commission within the Sixth Framework Program (2002-2006) | | |
|---|---|---|
| Dissemination Level | | |
| PU | Public | |
| PP | Restricted to other program participants (including the Commission Services) | X |
| RE | Restricted to a group specified by the consortium (including the Commission Services) | |
| CO | Confidential, only for members of the consortium (including the Commission Services) | |

# Camera Tracking and Autocalibration for Detecting and Correcting Camera De-Calibration

Akihiko Torii, Michal Havlena, Tomáš Pajdla

December 16, 2009

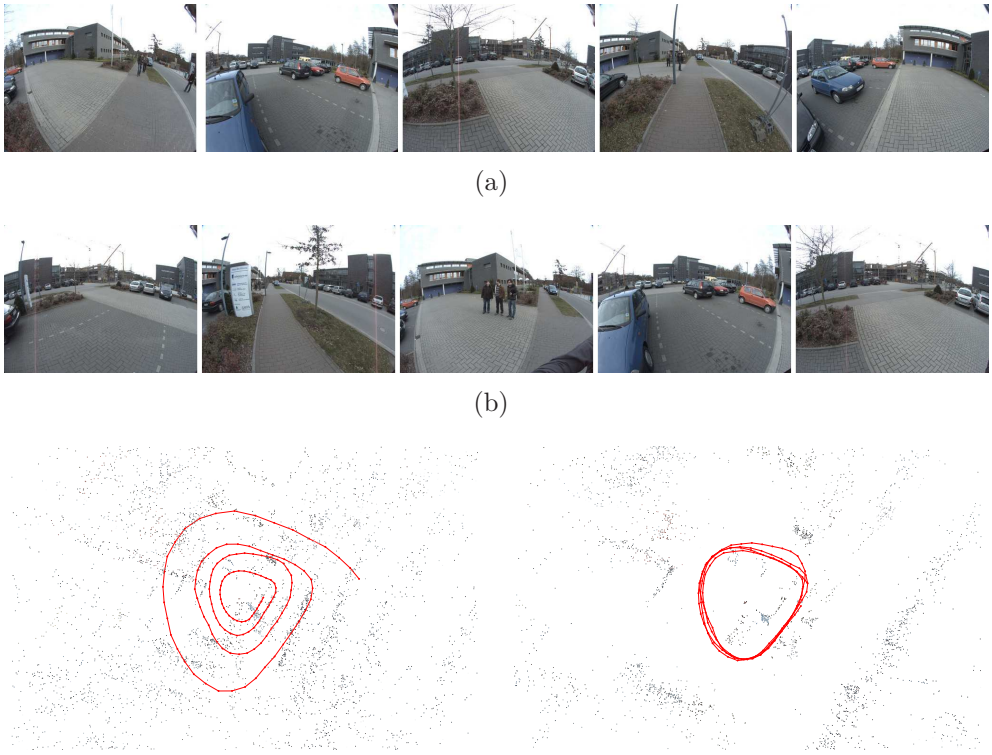
**Abstract**

In this report, we present several contributions to the dynamic 3D scene analysis supported by image and video processing from omnidirectional video data acquired by the AWEAR 2.0 platform. First, we summarize the upgrades of our structure from motion (SfM) pipelines for the autocalibration of the AWEAR 2.0 camera platform. Next, we examine several examples of detecting abnormal situations using statistics resulted from camera tracking: (i) feature detection, (ii) sequential matching, and (iii) stereo matching on the top of SfM. Finally, we demonstrate the detection and classification of abnormal situations, and correction of the contaminated camera calibrations according to the abnormal events on real video sequences.

## 1 Summary of Developments in 2009

There are significant upgrades of our structure from motion (SfM) pipelines for autocalibration of the AWEAR 2.0 camera platform. The contributions are summarized while listing the related and acknowledged publications of our SfM works.

**Calibration**. We focused on the vision aspect and hence the video system of the AWEAR 2.0 platform [2]. For a system aimed at cognitive support, fish-eye lenses are very helpful due to their extended field of view. As their handling requires some care, we will first describe the process of their calibration. Calibration of the lens reveals the transformation between the pixels in the omni-directional image and rays in 3D. As the intended lens



(a)

(b)

(c) (d)

Figure 1: Camera trajectory improved by loop closing on the sequence PARKING. (a) and (b) sample images: every 100th frame in the original video sequence. (c) trajectory before bundle adjustment. (d) trajectory after bundle adjustment with loop closing. Red lines with dots represent camera positions and trajectory. Colored dots are the reconstructed 3D points.

model is equi-angular, we use a two-parameter model [4] which is an extension of the equi-angular model that allows to compensate for small defects of non-expensive lenses due to manufacturing:

$$\theta = \frac{ar}{1 + br^2}. \tag{1}$$

Due to aberrations dependent on manufacturing and mounting, it is necessary to calibrate both lenses independently. For calibration, the entire field of view should be covered by a calibration target, rendering standard planar calibration targets unusable.

**Robust Camera Pose Estimation**. Tracks used for SfM are generated in several steps, (1) detecting SURF features [1], (2) constructing tentative

matches by using an approximate nearest neighbour search [5], and (3) verifying the tentative matches through epipolar geometry (EG) computed by solving the 5-point minimal relative pose problem [6]. The robustness and stability of camera pose estimation is further improved by PROSAC with soft voting, and by scale selection using a visual cone test [10].

**Bundle Adjustment Enforcing Global Camera Pose Consistency**. In a long camera trajectory, there might be some overlaps suitable for constructing loops that can compensate drift errors induced while proceeding the trajectory sequentially. We construct loops by searching pairs of images observing the same 3D structure in different times in the sequence. The candidate image pairs are found by using the image similarity matrix constructed based on visual words and vocabulary [8], and verified by solving the camera resectioning. Global consistency of camera poses and 3D structures are enforced by closing the loops by adding new constraints for the bundle adjustment [9] (see Figure 1).

**Image Stabilization Using Camera Poses and Trajectory**. The recovered camera pose and trajectory can be used to rectify the original images to the stabilized images [10]. If there exists no assumption on the camera motion in a sequence, the simplest way of stabilization is to rectify images w.r.t. the gravity vector in the coordinate system of the first camera and all other images will then be aligned with the first one. This can be achieved by taking the first image with care. When a sequence is captured by walking or driving on the roads, it is possible to stabilize the images w.r.t. the ground plane. For a gravity direction $\mathbf{g}$ and a motion direction $\mathbf{t}$, we compute the normal vector of the ground plane

$$\mathbf{d} = \frac{\mathbf{t} \times (\mathbf{g} \times \mathbf{t})}{|\mathbf{t} \times (\mathbf{g} \times \mathbf{t})|}. \tag{2}$$

We construct the stabilization and rectification transform $\mathtt{R}_s$ for the image point represented as a 3D unit vector such that $\mathtt{R}_s = [\,\mathbf{a}, \mathbf{d}, \mathbf{b}\,]$ where $\mathbf{a} = (0,0,1)^\top \times \mathbf{d} \,/\, \left|(0,0,1)^\top \times \mathbf{d}\right|$ and $\mathbf{b} = \mathbf{a} \times \mathbf{d} \,/\, |\mathbf{a} \times \mathbf{d}|$. This formulation is sufficient because the roads usually go up and down to the view direction.

# 2 Detecting Incongruences on Stereo Image Sequences

In this section, we recapitulate the problem definition as introduced in [7].

One of the goals of low-level processing is to preprocess incoming signals and extract reliable information. In particular, the processing should be able

to detect when some of the incoming information is wrong or when it is too unreliable to be used in further processing. Why should the result of processing be wrong? There may be various reasons. For instance:

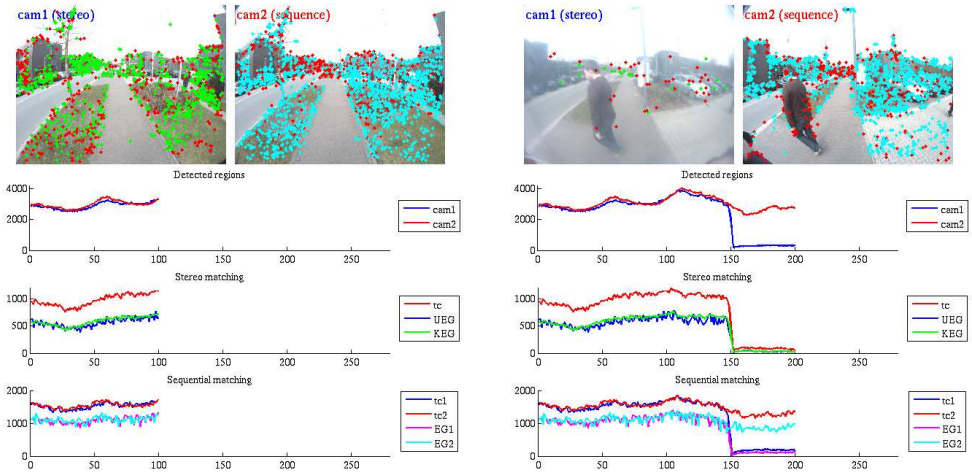
1. one or both cameras may fail to provide images,
2. one or both lenses may be out of focus,
3. one or both cameras may lose their calibration,
4. cameras may get be out of sync,
5. the epipolar geometry (EG) of the cameras may change.

The above items represent a hierarchy of events that can be detected by comparing the results of processing with expectations learned from previous situations.

Another possibility of detecting incongruences comes from the fact that AWEAR 2.0 cameras observe the same scene, only from slightly different viewpoints. Thus, the results on left and right images should, on certain level of processing, be comparable.

As the processing follows the standard path, we may add various detectors based on statistics of the results. The simplest statistics (detectors, classifiers) can be constructed by looking at the number of detected features, tentative matches, and matches verified by epipolar geometries. These numbers can be plotted into graphs as a function of frame number in the sequence. More advanced might be various quality measures, for instance the measure based on apical angles and view field coverage as is used in the randomized SfM [3].

We designed detectors of the above five events using the number of matches. Once events are detected, the next step is to take an action that will remove (if possible) the cause of the abnormality. For instance, we can try to recalibrate individual cameras as well as the camera rig. In fact, this action might be a part of the detection as well. For instance, when we detect that we can successfully track individual cameras but cannot track the rig, we can either expect the problem in rig calibration or in synchronization. Then, we may try to recalibrate or shift frames and choose the action that will better fit to incoming data. However, if none of these actions will improve the results, we will conclude that we are not dealing with this situation, we will start collecting these images, try to learn a model of the situation (e.g. by clustering in the feature space provided by the statistics) as a new phenomenon.



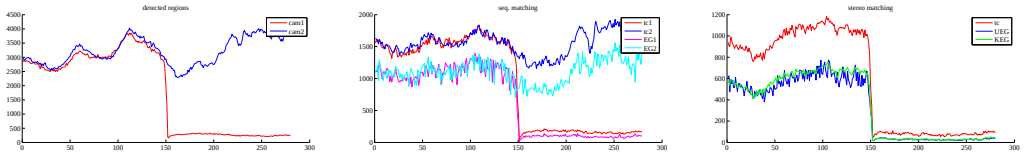

(a) (b)

Figure 2: Detecting the abnormal situation in the sequence POLYBAG. (Top row) input image pairs with matches at frame 100 (a), and frame 200 (b), *i.e.* before and after the abnormal event. (2nd, 3rd, and 4th rows) progress of feature detection, stereo matching, and sequential matching at frame 100 (a), and frame 200 (b). Colors are described in the text.

# 3 Detecting and Correcting Contaminated Camera Calibrations

In [7], it has been shown that the abnormal situations are detectable using statistics of the camera tracking. The remaining problems in [7] are to distinguish the abnormalities, and to correct the camera calibrations contaminated by the abnormalities in practical data. In this section, we demonstrate detecting abnormal situations and correcting camera calibration out of them on real and practical image sequences.

The sequence POLYBAG is a stereo image sequence consisting of 812 $\times$ 617 pixels large images of 280 frames long. When acquiring this sequence, the camera 1 (C1) was covered with a poly bag at frame 150 and thus subsequent frames are blurred. Top left in Figure 2(a) and (b) shows the original image of C1 with stereo matches of TC (red) and KEG (green) at frame 100 and 200, *i.e.* before and after the event, respectively. Top right in Figure 2(a) and (b) shows the original image with sequential matches of TC (red) and KEG (green) at frame 100 and 200, respectively. Furthermore, the statistics in the whole sequence are presented in Figure 3:




(a) (b) (c)

Figure 3: Statistics of camera tracking in the sequence POLYBAG. (a) feature detection, (b) stereo matching, and (c) sequential matching. Colors are described in the text.

(a) The number of detected SURF features. Red and blue colors correspond to C1 and C2, respectively.

(b) The number of matches between C1 and C2 in the stereo pair, i.e. between C1 and C2 of the same frame. The graph shows tentative matches (red), matches supported by an unconstrained epipolar geometry (UEG) computed in every frame (blue), and a known EG (KEG) computed using the precalibrated stereo rig configuration.

(c) The number of matches between consecutive images of C1 and C2 sequences. The graph shows tentative matches of C1 (red) and C2 (blue) and matches supported by an UEG of the consecutive images of C1 (magenta) and C2 (cyan), respectively.

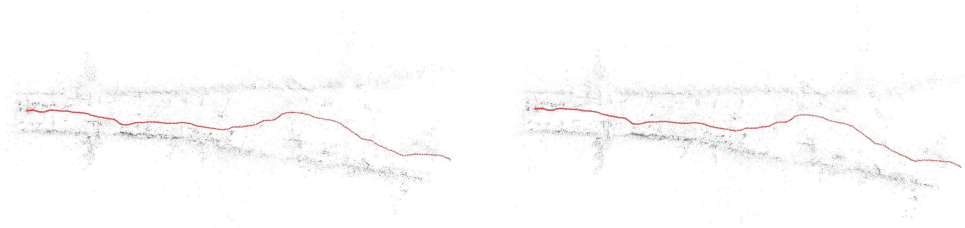
The occurrence of the abnormal event is easily detected by the number of features in (c), and the event is classified into "Camera fails" or "Camera out of focus" by taking into account the congruency of statistics (c), (d), and (e). Further classification between "Camera fails" and "Camera out of focus" can be performed by applying the low-level image processing techniques such as calculating the average intensity of images.

Next, we demonstrate the detection of abnormal situations "DESYNC: Cameras out of sync" and "DECALIB: Camera rig calibration wrong", their classification, and correction of the camera calibrations from these situations.

The camera tracking is first performed on the sequence CITYWALK which consists of every 5th frame of the original video sequence, *e.g.* Figure 4(a) and (b), acquired by well synchronized and calibrated stereo cameras. The camera tracking on the sequence CITYWALK is very challenging due to many objects (people) moving in the scene. The correctness of our camera tracking is implicitly verified by the recovered camera trajectories as shown in Figure 4(c) and (d). Figure 5(a), (b), and (c) shows the statistics of feature detections, stereo and sequential matching.

Figure 5(d), (e), and (f) shows the statistics of the camera tracking performed on the situation "DESYNC" which is generated by dropping first two

(a) (b)



(c) (d)

Figure 4: Camera tracking of the sequence CITYWALK. Input omnidirectional images of left camera C1 (a) and right camera C2 (b). Bird's eye view of the camera trajectory recovered by a single camera SfM using C1 (c) and C2 (d), respectively.

frames of C1 on the sequence CITYWALK. Figure 5(g), (h), and (i) shows the statistics of camera tracking performed on the situation "DECALIB" which is generated by applying a wrong rig calibration: 1 degree of rotation was applied to the precomputed stereo geometry, in other words, 1 degree of rotation was applied to normalized image points of camera 1. Colors in Figure 5 follow in the same as Figure 5 except for a recomputed EG (magenta) in (e), and (h) which indicates the number of stereo matches computed with a stereo rig configuration successively updated by recomputing EG from stereo pairs of neighbouring 5 frames. Clearly, these two situations can be distinguished by comparing the recomputed EG (REG) between Figure 5 (e) and (h).

The misalignment of synchronization is systematically specified in such a way that shift $\pm 1$ frame of left (or right) camera and perform KEG matching until finding a significant peak w.r.t. the number of matches.

The REG in Figure 5(b) validates that decalibration can be recovered by recomputing a stereo rig configuration using neighbouring frames. Furthermore, Figure 6 shows KEGs computed on wrong rig calibrations arranged by

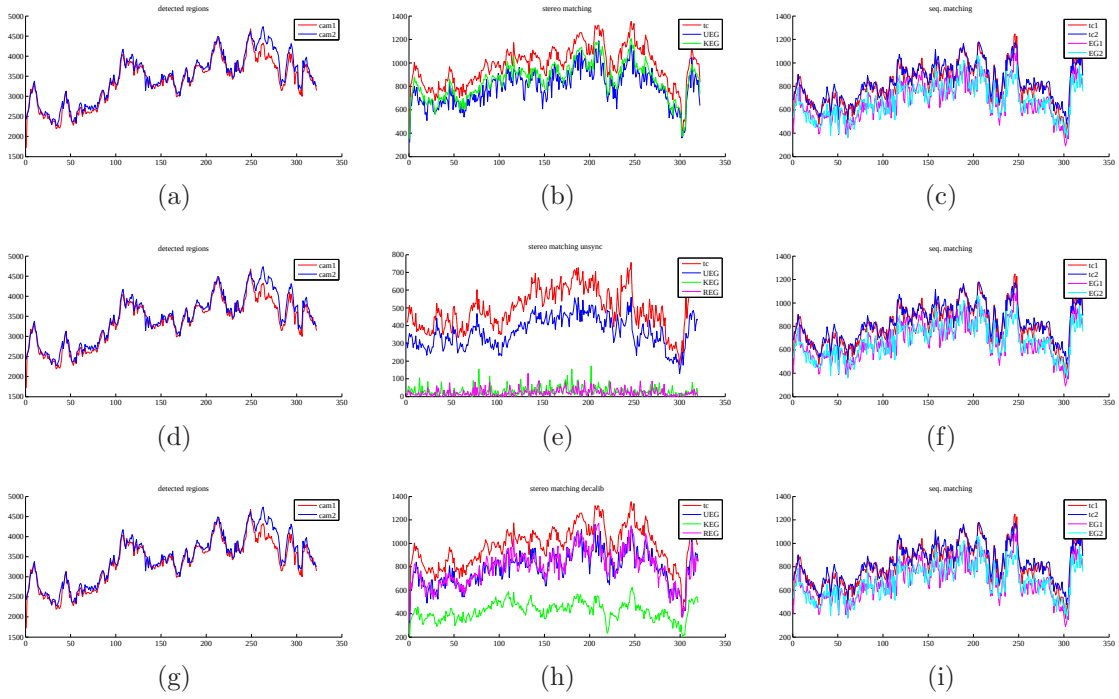



Figure 5: Statistics of camera tracking in the sequence CITYWALK. (a), (d), and (g) feature detection. (b), (e), and (h) stereo matching. (c), (f), and (i) sequential matching. Colors are described in the text.

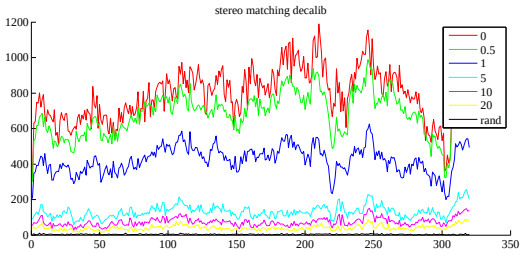



Figure 6: Statistics of KEGs computed with wrong rig calibrations on the sequence CITYWALK. Colors correspond to errors in degree as (red) 0, (green) 0.5, (blue) 1, (cyan) 5, (magenta) 10, (yellow) 20, and (black) 270.

tampering the precomputed stereo geometry, applying rotations varying from 0 to 20 degrees. The graph reveals that the incorrect stereo configuration is sufficiently detectable at 1 degree. On the other hand, KEG computed on 0.5 degree contamination returns a high number of matches and it is difficult to distinguish because the error caused by tampering the stereo calibration is absorbed in the tolerance, which is also 0.5 degree (2 pixels on images),

of evaluating matches in camera tracking. The precision can be improved by using high quality and resolution images and then setting the tolerance tighter.

# 4 Conclusions

We presented several contributions to the dynamic 3D scene analysis on omnidirectional video data acquired by the AWEAR 2.0 platform. First, we summarized the upgrades of our SfM pipeline, and examined the performance by computing the camera poses and trajectory on long video sequences. The experimental results revealed that the accuracy of computing the camera poses and trajectory is significantly improved by the loop closing technique which enforces the global consistency of camera poses when there exists overlaps of trajectory. Next, we reviewed abnormal situations which can be detectable by camera tracking, and examined several examples for their detection, and classification based on the statistics obtained from the camera tracking. Finally, we demonstrated the correction of the camera calibrations contaminated by the abnormal events on the top of detecting and classifying them on real and practical video sequences.

# References


[1] H. Bay, A. Ess, T. Tuytelaars, and L.J. Van Gool. Speeded-up robust features (surf). *CVIU*, 110(3):346–359, June 2008.

[2] M. Havlena, A. Ess, W. Moreau, A. Torii, M. Jančošek, T. Pajdla, and L. Van Gool. Awear 2.0 system: Omni-directional audio-visual data acquisition and processing. In *EGOVIS 2009: Proceedings of the First Workshop on Egocentric Vision*, pages 49–56, Madison, WI, USA, June 2009. Omnipress. CD-ROM.

[3] M. Havlena, A. Torii, W. Moreau, and T. Pajdla. Omnidirectional audio-visual data acquisition and processing. Research Report CTU–CMP–2008–26, Center for Machine Perception, K13133 FEE Czech Technical University, Prague, Czech Republic, December 2008.

[4] B. Mičušík and T. Pajdla. Structure from motion with wide circular field of view cameras. *IEEE Trans. PAMI*, 28(7):1135–1149, July 2006.

[5] M. Muja and D. Lowe. Fast approximate nearest neighbors with automatic algorithm configuration. In *VISAPP09*, 2009.

[6] D. Nistér. An efficient solution to the five-point relative pose problem. *IEEE Trans. PAMI*, 26(6):756–770, June 2004.

[7] T. Pajdla, L. Van Gool, M. Havlena, J. Heller, A. Torii, A. Ess, J.-H. Bach, H. Kayser, J. Anemüller, and P. Van Hengel. Incongruence detection in audio-visual processing. Research Report CTU–CMP–2008–28, Center for Machine Perception, K13133 FEE Czech Technical University, Prague, Czech Republic, December 2008.

[8] J. Sivic and A. Zisserman. Video google: Efficient visual search of videos. In *CLOR06*, pages 127–144, 2006.

[9] A. Torii, M. Havlena, and T. Pajdla. From google street view to 3d city models. In *OMNIVIS '09: 9th IEEE Workshop on Omnidirectional Vision, Camera Networks and Non-classical Cameras*, page 8, Kyoto, Japan, Octorber 2009. IEEE Computer Society Press.

[10] A. Torii, M. Havlena, and T. Pajdla. Omnidirectional image stabilization by computing camera trajectory. In Toshikazu Wada, Fay Huang, and Stephen Y. Lin, editors, *PSIVT '09: Advances in Image and Video Technology: Third Pacific Rim Symposium*, volume 5414 of *Lecture Notes in Computer Science*, page 12, Berlin, Germany, January 2009. Springer Verlag.