AWEAR 2.0 System: Omni-directional Audio-Visual Data Acquisition and Processing

Michal Havlena[1] Andreas Ess[2] Wim Moreau[3]
Akihiko Torii[1] Michal Jančošek[1] Tomáš Pajdla[1] Luc Van Gool[2,3]

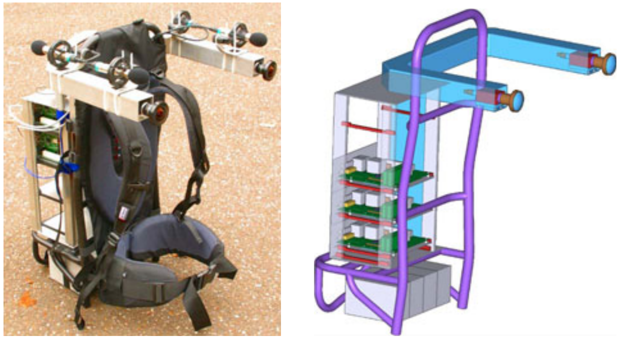
[1]Center for Machine Perception, Department of Cybernetics
Faculty of Elec. Eng., Czech Technical University in Prague
{havlem1,torii,jancom1,pajdla}@cmp.felk.cvut.cz

[2]Computer Vision Laboratory, D-ITET
ETH Zürich, Switzerland
{aess,vangool}@vision.ee.ethz.ch

[3]PSI-VISICS, ESAT
KU Leuven, Belgium
{Wim.Moreau,Luc.Vangool}@esat.kuleuven.be

Abstract

We present a wearable audio-visual capturing system, termed AWEAR 2.0, along with its underlying vision components that allow robust self-localization, multi-body pedestrian tracking, and dense scene reconstruction. Designed as a backpack, the system is aimed at supporting the cognitive abilities of the wearer. In this paper, we focus on the design issues for the hardware platform and on the performance of the current state-of-the-art computer vision methods on the acquired sequences. We describe the calibration procedure of the two omni-directional cameras present in the system as well as a Structure-from-Motion pipeline that allows for stable multi-body tracking even from rather shaky video sequences thanks to ground plane stabilization. Furthermore, we show how a dense scene reconstruction can be obtained from the data acquired with the platform.



(a) (b)

Figure 1. (a) The AWEAR 2.0 system is comprised of 3 computers, 2 cameras heading forwards, and a Firewire audio capturing device with 4 microphones (2 heading forwards and 2 backwards). Thanks to 4 lead-gel batteries, the autonomy is about 3 hours. (b) AutoCAD model of the system.

1. Introduction

We present a wearable audio-visual system aimed at helping the impaired, e.g. elderly, with their cognitive functions in everyday living. Care for the aging is becoming rapidly one of the major issues for the health care locally and globally. The escalating proportion of individuals over 65 with chronic diseases and with declining cognitive functions confronts the caregivers with economical, medical, and social challenges on a global scale, and the only economically feasible solutions rely increasingly on sophisticated technology including monitoring and cognitive assistant devices.

Our system can be seen as a first prototype of an assistive device that analyzes the environment and gives appropriate feedback to the wearer. Sensor-wise, it is equipped with two high resolution cameras with fish-eye lenses, as well as four microphones. A total of three computers (two for video, one for audio) process the incoming multi-modal data streams, powered by a battery pack that can sustain the system for up to 3 hours.

After introducing the actual hardware platform in Section 2, we will focus on the video processing pipeline. Especially in the intended supportive application, an enlarged field of view is of prime importance. We will therefore first study the calibration of the cameras in Section 3 before outlining an algorithm that can deliver robust and accurate camera pose estimates together with needed pre-processing steps in Section 4. When filming from a walking observer, the ground plane usually does not remain constant, we

978-1-4244-3993-5/09/$25.00 ©2009 IEEE

therefore propose a suitable stabilization approach. Based on this, we will show how to use the camera information together with a pedestrian detector to obtain multi-body tracking-by-detection and a dense 3D reconstruction of the scene in Section 5. The output of our system could then be used in further processing stages to give cognitive feedback to the wearer. In its current embodiment, the wearable device is mainly used as a recording platform. Our aim is to use the current state-of-the-art computer vision methods to process this type of image data in order to explore the potency and/or limits of egocentric vision, mainly in the means of constructing an assistive device.

The main benefits of the presented system are in particular (i) high resolution stereo, (ii) large field of view, and (iii) synchronization with multichannel high quality audio. When searching for similar devices, one can find separate solutions, but not the combination. Two experimental wearable devices closest to our platform are [13] and [14], both of them using just low resolution cameras and no audio. Existing commercial products offering broadcasting quality are generally not wearable.

The need for large field of view is demonstrated in [14], where two perspective cameras on each side of a walking person had to be used in order to cover the whole 360° horizontal field of view when solving a vision-based navigation task. Full resolution of the PointGrey Firefly MV cameras used was 752×480 pixels but as the system comprised of four Firewire-400 connected cameras, the frame rate could not go beyond 8*fps* due to bandwidth limitations.

2. AWEAR 2.0 Acquisition Platform

As AWEAR 2.0, shown in Figure 1, is constructed with the processing of multi-modal data streams at high frame rates in mind, the platform has to provide sufficient processing power while keeping both price and weight at acceptable levels. The constructed system comprises three computers, two for capturing video and one for capturing audio, the latter also acting as the controller of the entire system. The system is powered by a battery pack that allows autonomous operation for up to 3 hours. On the sensor side, two cameras heading forwards and a Firewire audio capturing device with four microphones (two heading forwards and two backwards) are used, triggered by a separate controller. All components, along with further supporting mechanic hardware, are mounted on a rigid frame. The weight of the presented system is 20kg (10kg of that for batteries).

2.1. Overview of the Major Components

1. Video hardware
   - 2 video recording computers, each with 2GB RAM, 570GB diskspace, and a Turion64 X2 TL-56 dual core processor
   - 2 AVT Stingray F-201C 2Mpixel 14*fps* color IEEE-1394b video cameras producing 1624×1234 RAW images
   - 2 Fujinon FE185CO86HA1 fish-eye lenses having approximately $150 \times 114°$ (H×V) FoV
2. Audio hardware
   - 1 audio recording computer with 1GB RAM, 250GB diskspace, and a Mobile Sempron 2100+ processor
   - 1 Focusrite Saffire Pro 10 I/O 10-channel 96kHz audio recording device with microphone preamps and phantom power supplies
   - 4 T. Bone EM700 condenser microphones with freq. range 50–20,000Hz and sensitivity -42dB
3. Trigger logic sending a synchronization signal to both cameras and the soundcard
4. Power dist. logic with 4 7.5Ah 12V lead-gel batteries
5. Ethernet connection and wireless adapter
6. Rigid frame backpack and other mechanic hardware

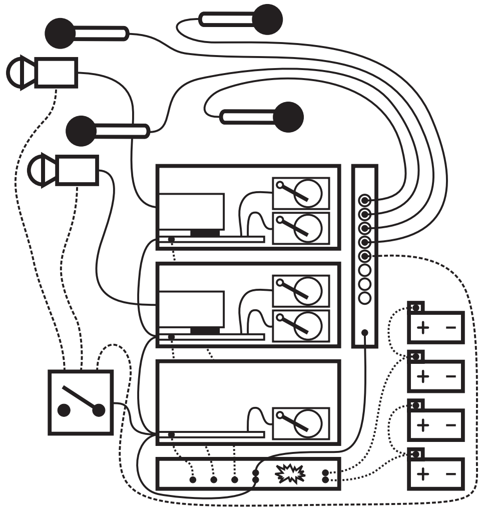
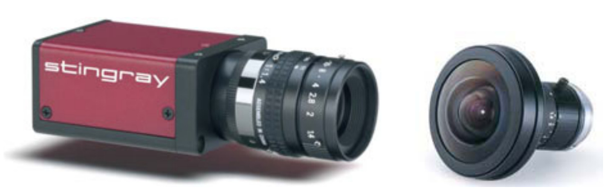


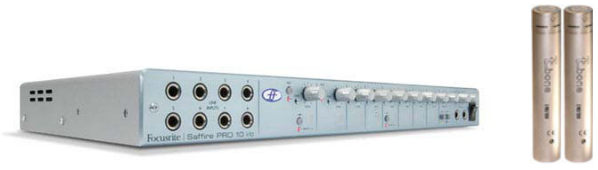
Figure 2. AWEAR 2.0 system consists of three computers, two Firewire-800 cameras connected via PCIe cards, four condenser microphones connected via a Firewire-400 audio capturing device, and a power distribution logic with four batteries. A hardware trigger is used to synchronize cameras and the audio. Data links are denoted by solid lines, trigger links by dashed lines, and power links by dotted lines.

Since not all the parameters were clearly known at design time, a modular design had been chosen. The platform can be extended to accommodate for four instead of two cameras or have the cameras replaced with faster ones capturing at double frame rate without having to modify the computing platform itself. Furthermore, up to four additional microphones can be added by just plugging them in. The computing platform has a margin in both bandwidth and processing power (see Figure 2).



(a) (b)

Figure 3. (a) AVT Stingray F-201C 2Mpixel 14*fps* color IEEE-1394b video camera producing 1624×1234 RAW images. Image courtesy of [1]. (b) Fujinon FE185CO86HA1 fish-eye lens with approximately $150 \times 114^\circ$ (H×V) field of view. Image courtesy of [9].



(a) (b)

Figure 4. (a) Focusrite Saffire Pro 10 I/O 10-channel 96kHz audio recording device with microphone pre-amps and phantom power supplies. Image courtesy of [8]. (b) T. Bone EM700 condenser microphones with range 50–20,000Hz and sensitivity -42dB. Image courtesy of [26].

2.2. Video Hardware

The video computers are two Siemens Mini-ITX industrial board computers D2703-S [25] with 2GB of RAM and 2 SATA 2.5” harddisks of 250 and 320GB each. The processor is an AMD mobile Turion 64 X2 Dual Core TL-56 which is equivalent to a dual core 1800MHz Intel processor.

The main design difficulty in the recording system is the bandwidth from the cameras to the disk on this mobile and hence relatively power-deprived system. As the bandwidth needed is 14*fps* $\times$ 1Byte/pixel $\times$ 1624×1234 pixels = 28MB/s for each camera, this has to be reflected in most design choices. We chose 2 harddisks in order to double the bandwidth to them and to alleviate possible higher recording rates (given appropriate cameras) or to include more cameras. SATA was chosen to increase the interface speed. Two 2.5” harddisks were preferred to one 3.5” harddisk (which typically has 1.5–2× higher bandwidth) for the weight and mainly for power consumption reasons.

The cameras are Firewire-800 (IEEE-1394b), connected via a PCIe card to the mainboard. Firewire-400 would in principle suffice but would nearly entirely saturate both Firewire buses, as only 80% of 400Mbps is reserved for isochronous transport. Cameras used are AVT Stingray F-201C [1] cameras (see Figure 3(a)). Triggering is performed via hardware through a USB general purpose 32-bit IO device. The trigger command is issued by the control computer connected to this USB device.

The employed lenses are Fujinon FE185CO86HA1 [9] high-resolution fish-eye lenses (see Figure 3(b)) which yield a field of view of approximately $150 \times 114^\circ$ (H×V) on the 2/3” CCD sensors of the cameras. The use of these high-resolution lenses results in relatively minor color aberration, so image data is free of artifacts across the entire field of view, and thus usable for processing.

2.3. Audio Hardware

The audio and control computer is about the same as the video nodes except the, in comparison to video, limited processing requirements which enable the use of a less powerful processor, an AMD mobile Sempron 2100+, using about 1/3 of the power of a video node under full load, thus increasing autonomy considerably. The audio recording device, a Focusrite Saffire Pro 10 I/O [8] 10-channel 96kHz interface with microphone pre-amps and phantom power supplies (see Figure 4(a)), is connected to this computer via the onboard Firewire-400 port.

Microphones used are T. Bone EM700 [26] condenser microphones (see Figure 4(b)) with a frequency range of 50–20,000Hz and a sensitivity of -42dB.

2.4. Power Distribution Logic

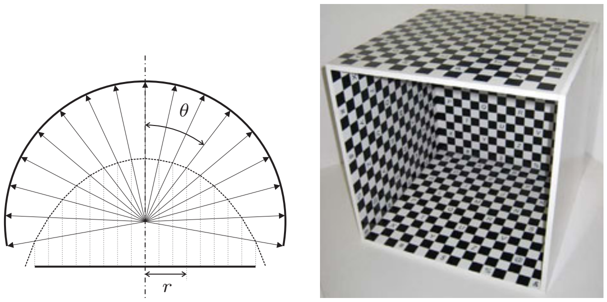
The computers’ onboard 24V DC power supply is connected to a power supply board which draws power from either the battery packs of 2×12V, 7.5Ah lead-gel batteries each, or an external power supply. Cameras are powered by the attached computer via the Firewire bus. The audio device is powered directly from the power board since the mainboard could not supply enough power over Firewire. This is accomplished by connecting the Firewire cable through the power board and using it as a power inserter.

Under full load, the video computers draw 2.1A @ 24V each (with cameras operating), the audio computer 0.7A @ 24V (with USB trigger attached), and the audio interface 0.25–1A @ 24V, totaling at about 5.5A. The batteries supply 15Ah @ 24+V, resulting in autonomy of slightly under 3 hours at full load. In normal operation, the actual consumption of the computers is about 1.1A for video and 0.35–0.5A for audio, resulting in a longer autonomy.

2.5. Software & Control

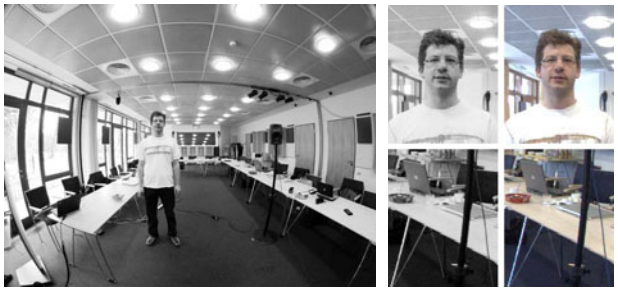
We use Ubuntu 8.10 as the operating system and several applications for video and audio capture. Video is captured in RAW (bayered) format into streams of 1000 files each, audio is saved as a 5-channel file, with the fifth channel containing the trigger pulses for video-audio synchronization.

The control computer is connected to the other computers via gigabit-Ethernet, and wireless access allows remote control of the entire system.



(a) (b)

Figure 5. (a) Equi-angular projection model. (b) Calibration target used for fish-eye lens calibration.



(a) (b)

Figure 6. (a) Sample RAW image from the camera. (b) Examples of debayered image data on two cutouts from the RAW image.

3. AWEAR 2.0 Calibration

In the following sections, we will focus on the vision aspect and hence the video system of the AWEAR 2.0 platform. 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 model is equi-angular (see Figure 5(a)), we use a two-parameter model [19] 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.

We thus chose a cube with a side length of 40cm as our calibration target. It is covered by a checkerboard pattern, with some cells labeled with letters and symbols (see Figure 5(b)). The top left corners of the labeled cells are selected manually, yielding a total of up to 160 2D points when the whole cube is visible. As the currently employed lens has a slightly smaller field of view, a part of the cube was not visible, but we could still measure more than 120 points. Based on the correspondence between 2D and 3D points, we optimize for the two parameters of the lens model along with the camera principal point and the unknown relative pose between the calibration object and the camera.

Next, in order to find the transformation between the left and the right camera, we recorded a short sequence of 808 frames while walking in a room and then recovered the epipolar geometry as follows.

In every 10th image pair a variety of features were detected, including MSER intensity +/- [17] and Laplacian-Affine [20]. Tentative correspondences between the left and the right image were found independently for each stereo pair captured at the same time. Finally, tentative correspondences from all pairs of images were concatenated and epipolar geometry was found thanks to the previously obtained lens calibration using RANSAC with the 5-point relative pose algorithm [22] as hypothesis generator.

To obtain a fully metric reconstruction, the real-world distance between the camera pair, in our case 45cm, needs to be applied to the found transformation.

4. Low-level Visual Data Processing

In our vision system, we differ between two levels of visual data processing. The lower level recovers camera positions for each stereo pair and provides the higher level, i.e. object detection and dense reconstruction, with appropriate images. Depending on the high-level module, this can be stabilization of the image w.r.t. a ground plane, or perspective cutouts.

4.1. Debayering

As the cameras use only a single CCD chip, the images first have to be debayered. For optimal results, we use an adaptive homogeneity-directed interpolation method suggested by [12]. An additional gamma correction is used to make the images more amenable for feature detection (see Figure 6(b)).

4.2. Cutout Generation

Based on the lens calibration, we are able to generate perspective, cylindrical, or any other projection model cutouts from the original omni-directional images. A given omni-directional image can be mapped onto a surface of a unit sphere as the lens calibration transforms image pixel positions into unit vectors (directions). This surface can be then projected to any other surface using the desired projection model. Technically, we do inverse filtering from the

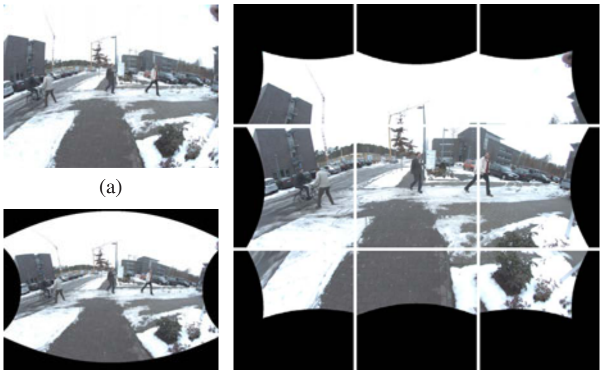


Figure 7. (a) Original omni-directional image. (b) Non-central cylindrical projection image. (c) 3×3 perspective cutouts. Notice the image overlap of the corner images with the other.

resulting image into the source image, the final pixel values are obtained by bilinear interpolation.

Non-central cylindrical projection images. To generate images more suitable for object recognition while keeping the full field of view, we use non-central cylindrical projection [27], see Figure 7(b). Optionally, images can be stabilized during the process using the estimated camera poses. This is especially helpful for fixing the ground plane from shaky images, thus constraining pedestrian detection to meaningful locations. See Sections 4.3 and 4.4 for details.

Perspective cutouts. Alternatively, we can also generate a group of nine 600×600 perspective images with field of view $60 \times 60°$ by projecting the surface of the sphere to its tangent planes in nine different directions. First is the direction of the optical axis (z-axis), other directions are computed by rotating the optical axis around the x-axis (top and bottom images), the y-axis (left and right images), or both axes (corner images) by 60 degrees. Using these settings, the resulting images cover the whole half sphere with image overlaps present in corner images only. If corner images are not used, there are no image overlaps present but some parts of the original image are missing (see Figure 7(c)).

4.3. Structure-from-Motion

Structure-from-Motion (SfM) computation recovers the unknown camera poses needed for image stabilization, multi-body tracking, and dense 3D reconstruction. We will demonstrate the pose computation based on the indoor sequence used above.

Sequence OLDENBURG is 808 frames long and the distance between consecutive frames is 0–0.2 meters as it was captured at 10*fps* while standing still and walking inside a room. For computing the camera poses by SfM robustly,

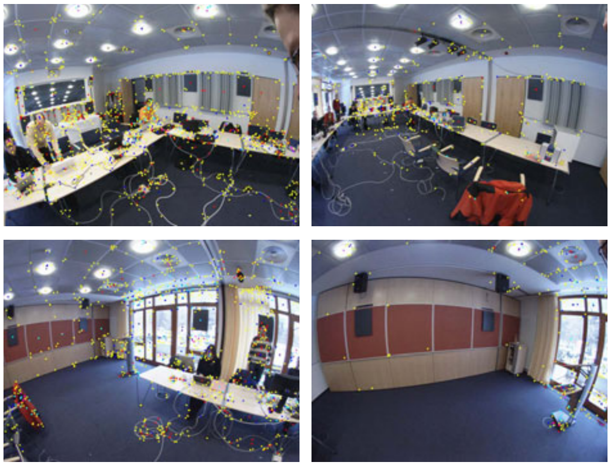


Figure 8. Features detected on images from sequence OLDENBURG. MSER int+ (red), MSER int- (blue), MSER sat+ (magenta), MSER sat- (cyan), and SURF (yellow). Notice the lack of feature regions on the last image acquired in a difficult turn.

129 keyframes are selected using the algorithm described in [27]. This algorithm computes the dominant, i.e. the most frequent, apical angle, which is the angle under which the camera centers are seen from the perspective of the reconstructed scene points [28]. A new keyframe is selected if the dominant apical angle between the current frame and the previously selected keyframe is greater than one degree. Otherwise, current frame is skipped and processed during gluing as described below.

In brief, the computation proceeds in several steps: First, different affine covariant feature regions including MSER [17] and SURF [2] are detected in input images (see Figure 8). The detected regions are assigned local affine frames (LAF) [23] and described by discrete cosine descriptors [24]. Secondly, tentative feature region matches are constructed from mutually closest descriptors in the feature space using FLANN [21] which performs fast approximate nearest neighbour search based on a hierarchical k-means tree. The 5-point minimal relative pose problem for calibrated cameras [22] is used for generating camera pose hypotheses and PROSAC [4], an ordered variant of RANSAC [7], together with voting similar to that used in [16] is used to find the largest subset of the set of tentative matches that is geometrically consistent. Finally, inliers of the geometry test are triangulated into 3D points [11] and the dominant apical angle is measured. Obtained relative camera poses are chained through the sequence resulting in the absolute poses of all keyframe cameras.

Using the recovered camera poses of the keyframe images, the camera poses of the remaining images are computed by the technique of gluing described also in [27]. If the scene is difficult and gluing fails for some cameras, their poses are obtained by interpolation from neighbouring cam-

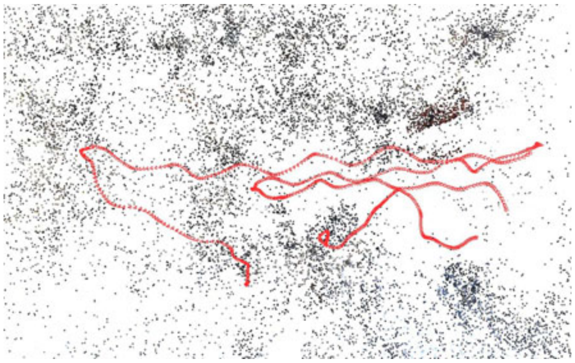


Figure 9. Camera trajectory of sequence OLDENBURG. The bird's eye view of the resulting 3D model. Small dots represent the reconstructed world 3D points. Red cones represent the camera positions of the whole sequence including glued and interpolated camera positions.

eras. Figure 9 shows the camera positions for the whole sequence and the recovered 3D points.

4.4. Image Stabilization

The recovered camera poses and trajectory can be used to rectify the original images to the stabilized ones. 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. Successful stabilization can be achieved by taking the first image with care.

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

$$\mathbf{a} = \frac{(0,0,1)^\top \times \mathbf{d}}{|(0,0,1)^\top \times \mathbf{d}|} \tag{3}$$

and

$$\mathbf{b} = \frac{\mathbf{a} \times \mathbf{d}}{|\mathbf{a} \times \mathbf{d}|}. \tag{4}$$

This formulation is sufficient because the pavements usually go up and down to the view direction.

Figure 10 shows several frames of the original images in sequence OLDENBURG (a), the corresponding panoramic images without camera stabilization (b), and the panoramic images stabilized w.r.t. the gravity vector in the coordinate system of the first camera using the recovered camera poses and trajectory (c).

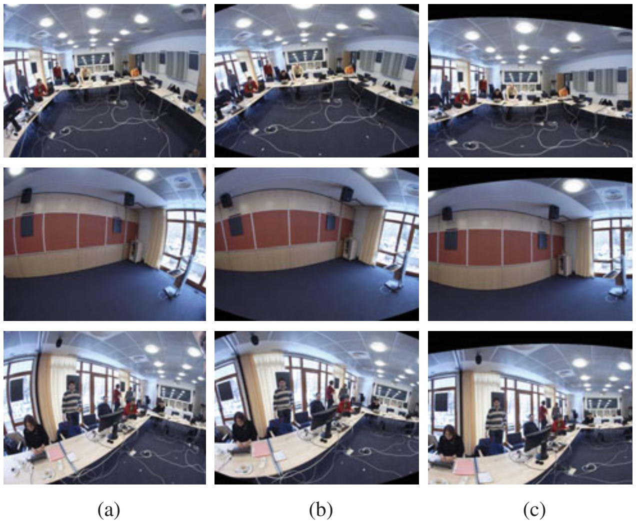


Figure 10. Result of our image stabilization and transformation in sequence OLDENBURG. (a) Original images. (b) Non-stabilized non-central cylindrical projection images. (c) Stabilized images w.r.t. the gravity vector in the first camera coordinates.

5. High-level Visual Data Processing

Given the data preprocessed by the lower levels of the system, we can now apply high-level visual data processing. In this paper, we will give two examples of high-level visual data processing: multi-body tracking and dense 3D reconstruction.

Both tasks are demonstrated on the sequence PEDCROSS, which is taken while walking on a pavement and observing several pedestrians. The sequence is 228 frames long, captured at 12*fps*, resulting in a distance of 0.05–0.15 meters between consecutive frames.

10 keyframes were selected using the algorithm described in Section 4.3 in order to have a sufficient dominant apical angle. Using the recovered camera poses of the keyframe images, the remaining camera poses are computed by gluing. Figure 11 shows the camera positions for the whole sequence and the world 3D points. Using the algorithms described above, we generate non-central cylindrical projection images that are stabilized w.r.t. the gravity vector in the coordinate system of the first camera.

5.1. Multi-body Tracking-by-Detection

The stabilized panoramic images form the basis for a tracking-by-detection approach that follows an earlier work [6]. In short, we detect pedestrian bounding boxes using a state-of-the-art detector [5] and place these detections into a common world coordinate system using the known ground plane and the camera positions.

The actual multi-body tracking system then follows a multi-hypotheses approach. For this, we accumulate the de-

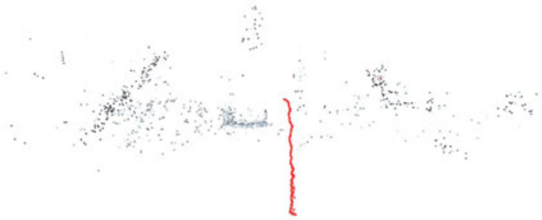


Figure 11. Camera trajectory of sequence PEDCROSS and bird's eye view of the resulting 3D model. Small dots represent the reconstructed world 3D points. Red cones represent the camera positions of the whole sequence including glued and interpolated camera positions.

tections of the current and past frames in a space-time volume. This volume is analyzed by generating many trajectory hypotheses using independent bi-directional Extended Kalman filters (EKFs) with a constant-velocity model.

To deal with data association uncertainties, we generate an overcomplete set of trajectories by starting EKFs from detections at different time steps. The obtained set is then pruned to a minimal consistent explanation using model selection. This step simultaneously resolves conflicts from overlapping trajectory hypotheses by letting trajectories compete for detections in the space-time volume. For the mathematical details, we refer to [15]. The most important features of this method are automatic track initialization (usually, after about 5 detections) and the ability to recover from temporary track loss and occlusion.

Figure 12 shows several frames of the sequence PEDCROSS before processing (a), the panoramic images without camera stabilization (b), and the results of the multibody tracking [6] performed on the sequence of panoramic images stabilized using the recovered camera poses and trajectory (c).

Obtained pedestrians and their tracks can be used as input to even higher processing levels, e.g. action recognition, which classifies pedestrians by their movement and detects pedestrians that behave unusually.

5.2. Dense 3D Reconstruction

Knowing the camera poses, one can reconstruct a dense 3D model of the captured scene. We have used a Scalable Multi-View Stereo (SMVS) pipeline which works with an unordered set of perspective images and corresponding camera poses [27], therefore we had to convert the input set of omni-directional images into perspective cutouts using the method described in Section 4.2.

The pipeline follows the reconstruction paradigm used in work [18], which can deal with large video sequences working with a few neighbouring frames of each actual frame to compute and fuse the depth maps. We build upon the work of [10]. In particular, we modify the reconstruction process to be scalable by accumulating reconstructed scene and avoiding unnecessary computations and improve the filtering step by using MRF filtering formulation [3]. Four different views of the resulting model are shown in Figure 13.

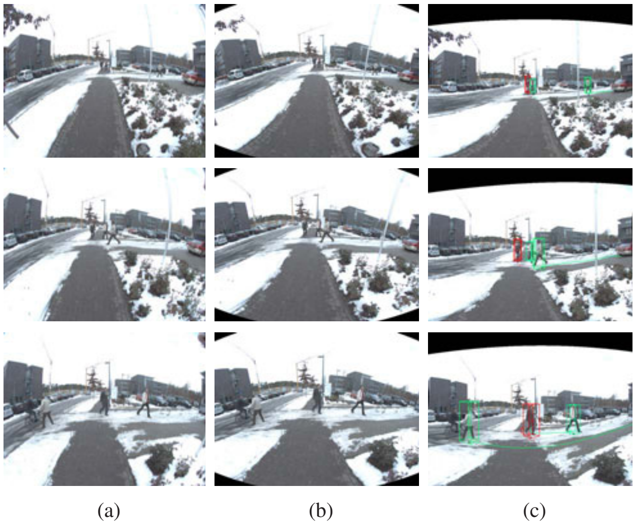


Figure 12. Results of our image stabilization and transformation in sequence PEDCROSS. (a) Original images. (b) Non-stabilized non-central cylindrical projection images. (c) Stabilized images w.r.t. the gravity vector in the first camera coordinates together with pedestrian tracking.

The obtained result can be used for depth map generation, which facilitates image segmentation and also allows for better understanding of the scene geometry utilized by other higher processing levels, e.g. obstacle detection, warning the wearer from unwanted hits.

6. Conclusions

We have described a wearable audio-visual sensor platform that is aimed at cognitive supportive applications for the elderly. Both aspects of hardware and software were considered. As we used fish-eye lenses for extending the field of view, particular image data processing methods were required. To this end, we proposed low-level methods for calibration, projective mapping, SfM, and image stabilization from such data. Based on the robust output of these components, we demonstrated two high-level tasks: multibody pedestrian tracking and dense 3D reconstruction.

In its current embodiment, we use the hardware platform as capturing device only, with video stream processing done off-line. While the low-level processing steps are not far from real-time performance: debayering takes 1s per image, cutout generation 2s per image, and SfM with stabilization 7s per image, high-level processing steps are slower: pedestrian detection takes 40s per image and dense 3D reconstruction 60s per image.

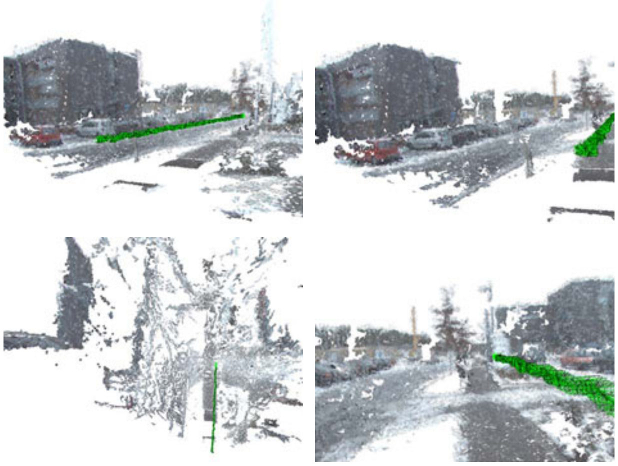


Figure 13. Four different views of the final dense 3D reconstruction of sequence PEDCROSS with given camera positions and orientations (green).

Future work will encompass bringing the algorithms up to speed in order to run on the platform. Then, additional higher level components can provide actual cognitive feedback to the wearer.

Acknowledgements

This work was supported by EC project FP6-IST-027787 DIRAC and by Czech Science Foundation under Project 201/07/1136. A. Ess was supported by Toyota Motor Corporation, M. Jančošek was supported by the Grant Agency of the Czech Technical University under project CTU0908713, and T. Pajdla was supported by Czech Government under the research program MSM6840770038. Any opinions expressed in this paper do not necessarily reflect the views of the European Community. The Community is not liable for any use that may be made of the information contained herein.

References

[1] AVT – Stingray. *http: // www.alliedvisiontec.com / avt-products / cameras / stingray / f-201-b-bs-c-fiber.html*, 2008.

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

[3] N. Campbell, G. Vogiatzis, C. Hernandez, and R. Cipolla. Using multiple hypotheses to improve depth-maps for multi-view stereo. In *ECCV'08*, 2008.

[4] O. Chum and J. Matas. Matching with PROSAC – progressive sample consensus. In *CVPR'05*, pages 220–226, 2005.

[5] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR'05*, pages 886–893, 2005.

[6] A. Ess, B. Leibe, K. Schindler, and L. Van Gool. A mobile vision system for robust multi-person tracking. In *CVPR'08*, 2008.

[7] M. Fischler and R. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Comm. ACM*, 24(6):381–395, June 1981.

[8] Focusrite – Saffire. *http: // www.focusrite.com / products / saffire / saffire_pro_10_io /*, 2008.

[9] Fujinon – FE185. *http: // www.fujinon.com / security / product.aspx ? cat=1019 & id=1072*, 2008.

[10] Y. Furukawa and J. Ponce. Accurate, dense, and robust multi-view stereopsis. In *CVPR'07*, 2007.

[11] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, second edition, 2003.

[12] K. Hirakawa and T. Parks. Adaptive homogeneity-directed demosaicing algorithm. *IP*, 14(3):360–369, March 2005.

[13] R. Kelly and R. Green. Camera egomotion tracking using markers. In *IVCNZ'06*, 2006.

[14] O. Koch and S. Teller. A self-calibrating, vision-based navigation assistant. In *CVAVI'08*, 2008.

[15] B. Leibe, K. Schindler, N. Cornelis, and L. Van Gool. Coupled object detection and tracking from static cameras and moving vehicles. *PAMI*, 30(10):1683–1698, October 2008.

[16] H. Li and R. Hartley. A non-iterative method for correcting lens distortion from nine point correspondences. In *OMNIVIS'05*, 2005.

[17] J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide baseline stereo from maximally stable extremal regions. *IVC*, 22(10):761–767, September 2004.

[18] P. Merrell, A. Akbarzadeh, L. Wang, P. Mordohai, J. Frahm, R. Yang, D. Nister, and M. Pollefeys. Real-time visibility-based fusion of depth maps. In *ICCV'07*, 2007.

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

[20] K. Mikolajczyk et al. A comparison of affine region detectors. *IJCV*, 65(1-2):43–72, 2005.

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

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

[23] Š. Obdržálek and J. Matas. Object recognition using local affine frames on distinguished regions. In *BMVC'02*, pages 113–122, 2002.

[24] Š. Obdržálek and J. Matas. Image retrieval using local compact DCT-based representation. In *DAGM'03*, 2003.

[25] Siemens – D2703. *http: // www.fujitsu-siemens.com / products / prof_accessories_mainboards / mainboards / industrial / d2703.html*, 2008.

[26] Thomann – T. Bone. *http: // www.thomann.de / gb / the_tbone_em700_stereo-set.htm*, 2008.

[27] A. Torii, M. Havlena, M. Jančošek, Z. Kúkelová, and T. Pajdla. Dynamic 3d scene analysis from omni-directional video data. Research Report CTU–CMP–2008–25, CMP Prague, December 2008.

[28] A. Torii, M. Havlena, T. Pajdla, and B. Leibe. Measuring camera translation by the dominant apical angle. In *CVPR'08*, 2008.