Omni-directional recording system for AMI Meeting Room

AMIDA technology package description

Lukáš Polok, Adam Vlček

Graph@FIT
Brno University of Technology
Faculty of Information Technology
Božetěchova 2
612 66 Brno, Czech Republic

Developers:
Igor Potúček, Stanislav Sumec,
Lukáš Polok, Pavel Zemčík

Pavel Zemčík, zemcik@fit.vutbr.cz, responsible for AMIDA project
Adam Herout, herout@fit.vutbr.cz, responsible for Graph@FIT group

Last modified: December 13, 2009

1. Introduction

This document covers our solution of highly mobile low cost meeting recording system for the purpose of the Augmented Multi-party Interaction Project (AMI). It uses a single camera pointed at a hyperbolic mirror, which allows capturing of a large portion of the space angle. The captured image is processed efficiently on any laptop with today's consumer GPU, enabling processing of frame rates well above any consumer camera frame rate. A fast image stabilization algorithm allows for a very simple and cheap camera-mirror holder. Together with a fast image unwrapping algorithm, the system is capable of recording a full 360 degree view of the meeting participants seated around a table. Easy-to-use interface makes integration into existing projects a matter of moment.

2. Motivation

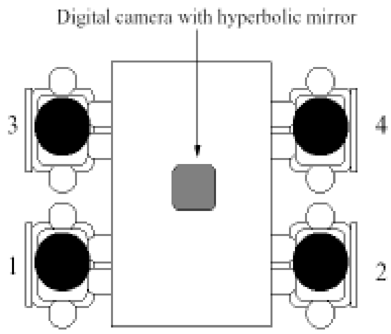

Figure 1: Participant positions

Meetings are important part of everyday human social life. It is generally suitable to retain information contained in the meetings for later use. Traditional approach of manual transcription is time and money consuming. Modern technology can help to automate meeting recording. Various systems, which use one or several fixed cameras, were proposed, but one common important disadvantage of these systems is their installation requirements because cameras are usually mounted to the walls of the meeting room. Also, the size of such systems is unacceptable for mobile applications.

We propose the solution for mobile meeting room with one camera with a special mirror to replace several standard cameras. The system is built from the off-the-shelf products. It is expected that the participants are located around the table, as shown in figure 1. Such case is suitable for use of our system, which is then placed in the middle of the table. Each participant is captured from the frontal view. The processed video output from this system has panoramic view of all 360 degrees. The recording equipment in our system consists of a standard laptop, one camera, hyperbolic mirror, a stand, and a set of microphones. The captured image is not suitable for the human observer without prior processing (the unwrap transformation), which our system performs in real time.

This package is concerned mainly with methods for transformation of the captured picture to the panoramic landscape view and it's stabilization. Other separate packages are concerned with speaker detection / recognition, movement estimation, choosing good view parts from the 360 degree image, automatic cuts, and automatic video summarization.

3. Omni-directional system

The system allows capturing large portion of the space angle - e.g. 120 x 360 degrees. It is based on perspective camera and hyperbolic mirror on the mirror holder. It is possible to use setup with mirror below the camera, or above the camera as well. The system can be improved by using transparent tube, acting as a mirror holder, so that the output image isn't obstructed by it.

Figure 2: Omni-directional system and the hyperbolic mirror, as seen by the camera

3.1. Transformation of panoramic view

The output image has circular native stage and needs to be further processed. The main property of such image is non-uniform resolution in vertical axis. The image needs to be transformed into the panoramic view before detection and recognition. The parameters of transformation depend on the mirror profile equation.

The mirrors are usually designed with angular unit gain in the given distance from vertical axis. This property allows simple unrolling algorithm to be used. Simple transformation can cause distortion in vertical axis of the resulting image in other cases. Therefore, the more sophisticated geometrical transformation is more suitable for mirrors, which do not have unit angular gain. Both transformations use the rotationally symmetric property. The simplest transformation uses just "unwrapping" of the source image. The properties of the input captured omni-directional image which are center and radius of projected circle are known. Knowledge of mirror geometry is not needed.

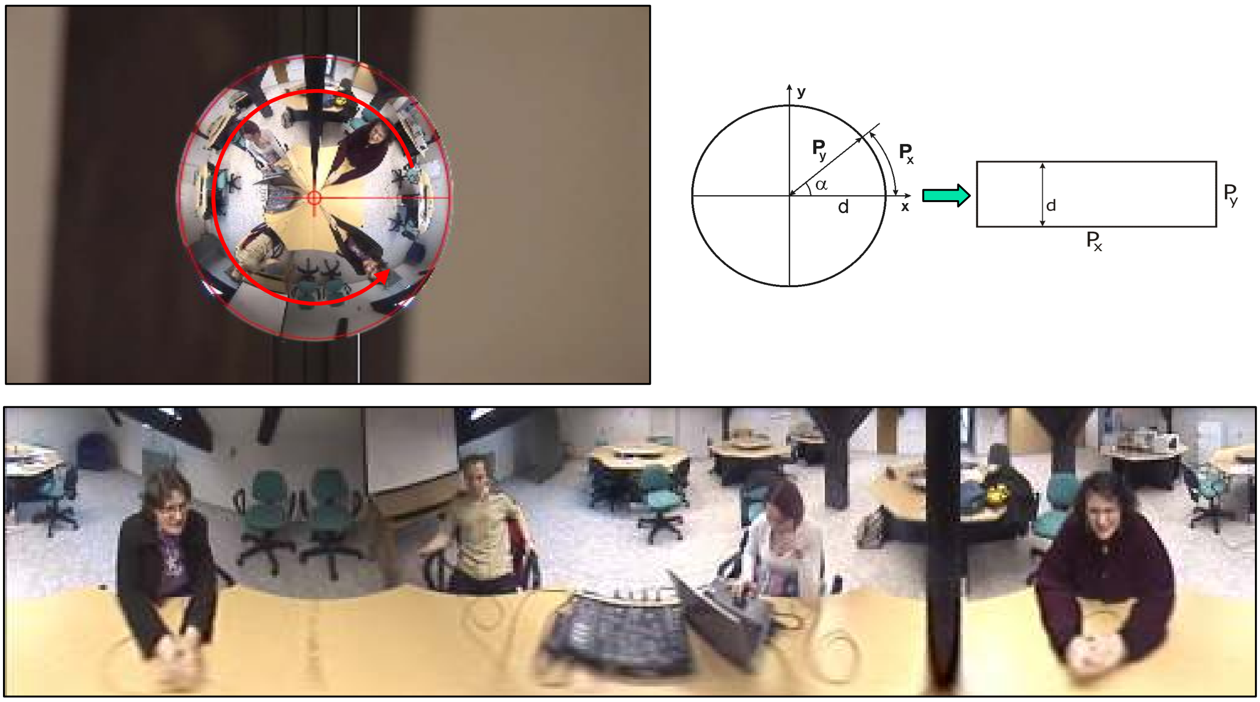

Figure 3: Unwrapping uniform-resolution mirror (notice the deformation of the board)

The main advantage of this transformation is it's low computational cost, but may not be usable for computer processing such as face recognition due to unrolled image deformation, as seen on figure 3. Another approach is to use geometrical properties of the mirror for image projection on the cylindrical plane around the major mirror axis. It results in deformation-less transformation (line in the original space maps to a line in transformed image). Due to the rotational symmetry of the system only the information about the mirror profile is used. The image formation can be expressed as a composition of coordinate transformations and projections. The camera center has to coincide with the second focal point of the mirror to preserve the single effective viewpoint. The geometry of the image formation of the omni-directional catadioptric camera is shown in figure 4.

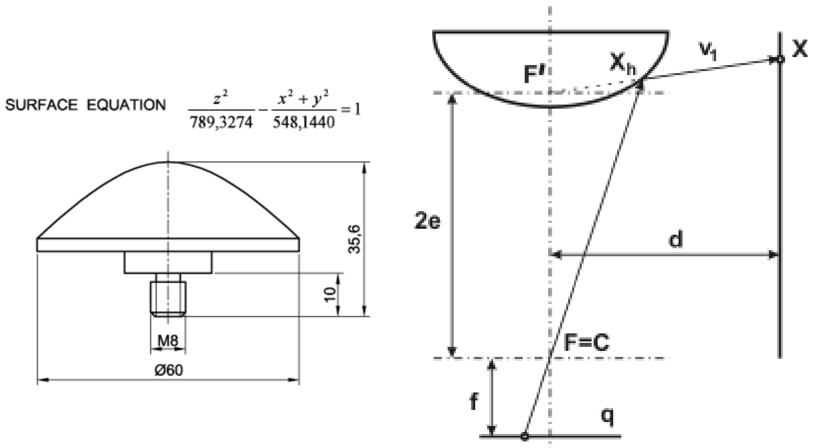

Figure 4: Geometric transformation

Transformation was implemented in GPU shader, for performance reasons - there is need to interpolate image samples to get good quality unwrapped image, which can be quite demanding task for CPU at high resolutions. The transformation itself is also quite computationally intensive, and it can't be precalculated because transformation parameters constantly change because of table vibrations.

There are two ways of calculating the transformation, we can take advantage of raw GPU power and simply calculate transformed coordinates per pixel, or we can take advantage of the fact the transformation of x and y coordinate (in unwrapped image space) is independent, so it is possible to calculate two tables, for transformation of either coordinate and combine them when sampling input image. The disadvantage in this approach is it is necessary to calculate those tables everytime transformation parameters change (every frame), which means expensive rendering context switching. Floating-point precision textures are also needed to store the tables, which may not be available in some low-end hardware. Our implementation elegantly solves these problems.

Some additional post-processing functions were implemented to leverage fragment processing power. There is option to calculate grayscale values of the transformed image, and store them in alpha channel to be used by some following algorithms on CPU. The other option is to evaluate simple Gaussian model for skin locust - based skin color detection [3], skin color probability is stored in alpha channel again.

3.2. Image stabilization

The camera mounted to the holder with mirror is susceptible to the vibrations, caused by meeting attendees writing, leaning on the table and similar. These vibrations create undesirable movements in the transformed image, with magnitude increasing towards upper border in unwrapped image.

Automatic detection of the transformation parameters can solve this problem. The output image from catadioptric system has circular shape, which is given by the mirror top view, as seen in figure 1. The transformation algorithm uses information about circle center and radius.

Because input image consists of simple background and circular “omni image”, it is easy to find border of the “omni image” with it's properties. The goal is to extract pixels laying on the circle border. The first step is to use edge detector for circle border sharpening. The starting position of the detection is around the circle adjusted to the image center. The circle border is tested in the direction from outer to inner area border, because the image background contains almost no edges.

The circle fitting algorithm is used for computing center and radius from pixels positions on the circle border. For accuracy improvement, the iteration method is used. The border interval is decreasing in each cycle until the number of detected points is lower than 1/5 of all possible points, which can be detected on the border. The resulting accuracy of the detected center and radius is ±1.5 pixels, which is enough for automatic initial calibration.

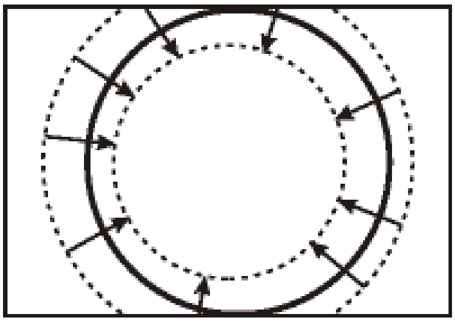

Figure 5: Searching for mirror edge points

There's function to calculate edge points on GPU. Edge points are found as maxima of difference at lines with uniform angular interval around (assumed) mirror center, bounded by minimal and maximal radius, as seen on figure 5. This gives always the same number of detected points. The points are stored with 16:16 fixed point format, providing sufficient precision for the circle fitting algorithm. The only problem with this approach can be higher number of outlier points (edges in image in the mirror, not the edge of the mirror), but that can be solved by choosing contrasting mirror background to increase edge detector response.

Our implementation also contains different, more robust approach to finding precise image border, which aren't burdened by the double-edge problem. It is described in closer detail in publications, which are part of the package.

3.3. Possible extensions

The implementation could be extended to make image transformation on GPU more efficient, using some newer functionality, which wasn't available at the time of development. Expected speedup would only be marginal though.

3.4. Hardware requirements

The development system setup was formed by camcorder Sony HDR-FX1 with HD image output

(1440 x 1080px), hyperbolic mirror H3G from company Neovision (Prague), and PC with GeForce 6600 GPU. The system is able to work with greater resolutions, up to GPU texture size limit (4096 x 4096px on GeForce 6600, up to 8192 x 8192px on today's hardware). Requirements on the GPU are support for fragment shaders, GLSL language version 100, and either P-buffers or frame-buffer objects for off-screen rendering. Asynchronous transfers were implemented to overlap image transfer to GPU with transformation of previous frame. For asynchronous transfers to work, pixel buffer objects and fence synchronization primitives must be supported.

Conclusion

The low cost and mobile meeting room designed for recordings and presentation of relevant information to the human was presented. The methods for image transformation and camera stabilization were used for this task. The video processing is necessary when meeting events need to be presented. Image quality of such system with common digital camera is not same as with fixed mounted camera systems, but hardware requirements are much lower as the price. However, the results can be comparable with standard systems, if the HDTV camera with better resolution is used. This kind of meeting monitoring system is suitable for casual meetings at different places and conditions.

Acknowledgements

This work has been founded by the European Union 6th FWP IST Integrated Project AMI (Augmented Multi-party Interaction) FP6-506811.

References

[1] Nayar, S., Baker, S., A Theory of Catadioptric Image Formation, Department of Computer Science, Columbia university, Technical Report CUCS-015-97.
[2] Svoboda, T.: Central Panoramic Cameras Design, Geometry, Egomotion, Center for Machine Perception, Faculty of Electrical Engineering, Czech Technical University, September 30, 1999.
[3] Jones, M., Rehg, J.: Statistical Color Models with Application to Skin Detection, Cambridge Research Laboratory, Computer Vision and Pattern Recognition (CVPR99), Ft.Collins, CO, 274-280, June, 1999.

List of attachments

1. AMI Demo application binaries and source code. Pictures of AMI Demo application in action, along with some description, can be seen at:

   http://www.amiproject.org/showcase/integrated-systems/mobile-meeting-capture-system

2. TAVFile source code, along with precompiled binaries (TAVFile is C++ class, wrapping audio/video input/output, including input form the HDTV camera).
3. GPU Transformation library source code, along with precompiled binaries
4. Image stabilization code
5. Skin model acquisition tool and source code
6. Documentation, along with collected publications