Camera localization

AMIDA technology package description

Marek Šolony, Pavel Žák

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

Technology developers:
Igor Potůček, Pavel Zemčík, Vítězslav Beran,

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

Last modified: January 4, 2010

Contents

1 Purpose of the technology

Camera view localization means resolving camera position and orientation within three-dimensional space(see figure 1) which has potencial usage in several areas, such as user interfaces, scene reconstruction, robot navigation and so on.

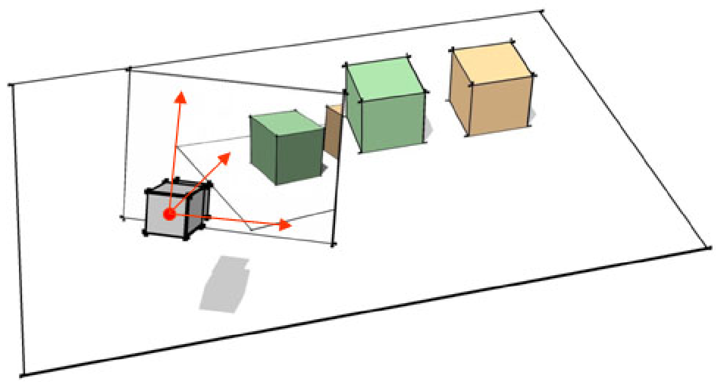

Figure 1: Camera position and orientation within 3D scene.

2 Features

Described technology deals with location-fixed cameras that can change their orientation and/or zoom. These camera parameters are automatically resolved in captured videostream and enables e.g. adding virtual objects into the viewed scene which creates augmented reality and can perform in modern user interfaces(as can be seen in figure 2).

This document contains description of the camera localization technology and its implementation. Next section brings high level technology description and following chapters focus on key parts of computational process.

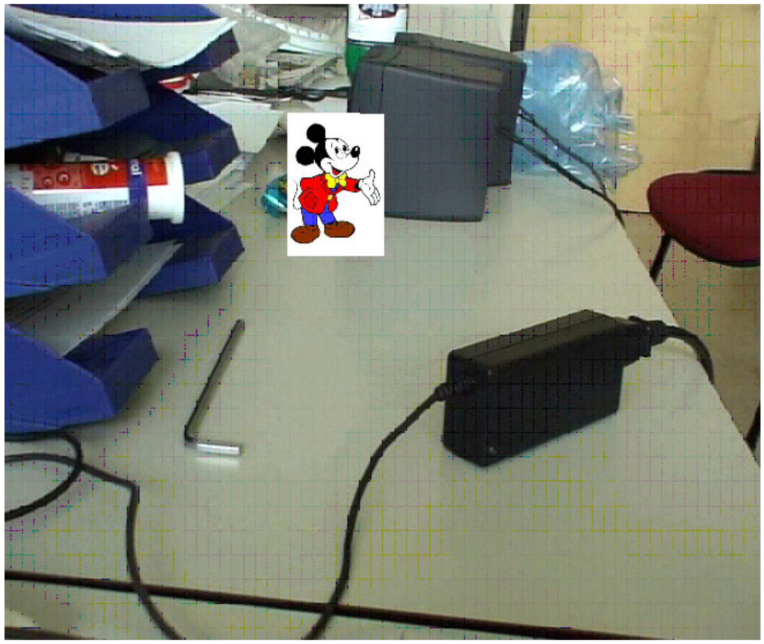

Figure 2: Example of inserting virtual object(2D image of Mickey) into the captured scene.

3 Technical description

3.1 Algorithm overview

Described technology for camera localization process works in a few distinguishable steps:

1. Frame by frame interest point detection
2. Finding point correspondences between frame couples
3. Resolving camera parameters

First distinctive points are located within each frame in video sequence. It could be assumed that in temporal adjacent frames merely the same set of such points are located. That enables us to find correspondences of detected points in close frames. From this information it is possible to discover camera parameters using so called homography properties.

Each step of proposed approach can be solved in several ways, however this text focuses only on existing implementation.

3.2 Interest points detection

As interest points of the image we usually mark such point in the image that has some major and notable characteristic within the image space that enables simplifying of image description and is stable and detectable under local or global image deformations. One of easy to imagine representation of interest points are so called corner points that marks places in images with two dominant and different edge directions in a local neighbourhood of the point.

In current technology, Harris corner detector [2] is used for corner points detection.

3.3 Interest point correspondences

One of the most frequent techniques for finding image point correspondences is to track their position through image sequence. In the reference image of sequence the corner points are located and in the adjacent frames their new positions are searched. To reduce the computational cost the position can be predicted using so called Kalman filter[3].

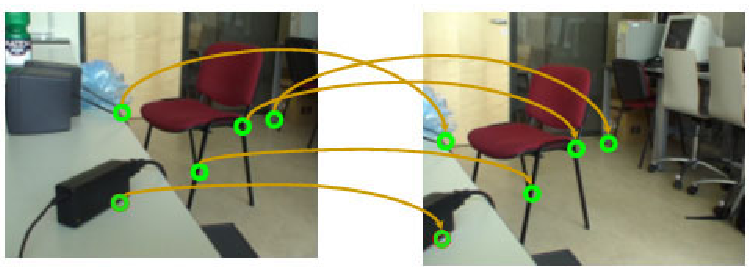

Figure 3: Example of several detected corner points pairs in two images.

3.4 Homography

The process of camera parameters recovery begins with establishing frame-to-frame relation called *homography*. This relation maps *points* from first image to special *lines* in second image and vice versa. Images can be either picked from video sequence of moving camera or from stereo camera pair capturing the same scene. General homography won't be described in this document, since only its special case - *planar homography* is used in this technology. Planar homography occurs in following cases:

- Video sequence is captured from position fixed camera and only rotation and zoom is performed
- Stereo camera pair captures scene with planar surfaces

Planar homography relates *point* in first image to *point* in other, that means knowing position of relevant point in first image can be used to compute its position in another image. In first described case, all points from image, in second case only points from corresponding planar surfaces are related by planar homography. Estimation of homography is part of many computer vision applications, in which the relation between two or more images is used for camera parameter recovery, metric reconstruction, mosaicing, or virtual object insertion. Figure 4 shows different cases of planar homography.

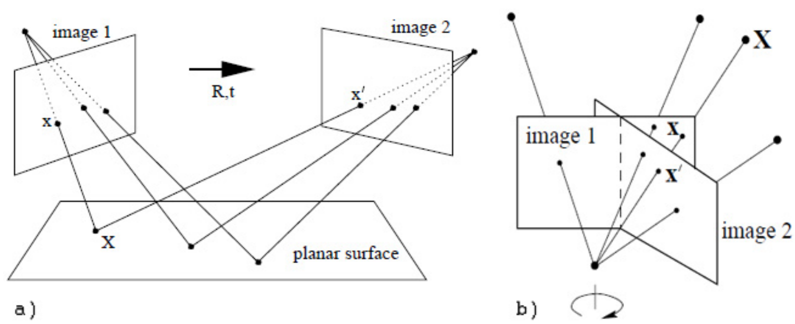

Figure 4: a) Views of planar surface from different positions and with different camera angles. Planar surface and its images are related by homography. b) View with fixed camera center, but different camera angles. Images of all scene points are related by homography.

Our technology supposes the first described scenario - position of camera is fixed. We focuse on estimation of planar homography for further use i.e. correct virtual object insertion.

3.4.1 Estimation of planar homography

Overview of this algorithm is that planar homography is computed between points we know they are corresponding, and then this relation can be applied to whole image. Classical approach begins with extraction of interest points described in section above. Corresponding point pairs are found with help of RANSAC algorithm, which will be described further in text. The planar homography is computed by solving a system of homogenous equations set up from such point pairs.

If x and x' are coordinates of corresponding points in the first view, and in the second, the relation between them is mathematical defined such that:

$$x' = Hx \tag{1}$$

where H is homography represented by 3×3 matrix. This formula can by applied to any point from first image to find its corresponding point.

Feature based methods alone yield number of incorrect point matches, so advanced methods based on RANSAC are applied in order to get better results. RANSAC is short of RANdom SAmple Consesnsus, and basic idea behind this algorithm is following:

1. Choose randomly enough point pairs to compute homography.
2. Compute for H and test how well it fits with rest of point pairs that weren't used for computation.
3. Continue with step 1. until best fitting result is found.
4. Points that satisfy formula (1) are used to compute homography, leaving false matches unused.

The last step is necessary because homography in step 2. was computed to fit the subset of all correspondencies, and if there is any noise, resulting homography will be biased by the noise of those points.

The RANSAC based algorithm described above allows detection of homography between images obtained by camera with fixed camera center, or between images from stereo camera pair if planar surfaces in scenes are present. Figure 5 shows the result of application of this algorithm to a video sequence.

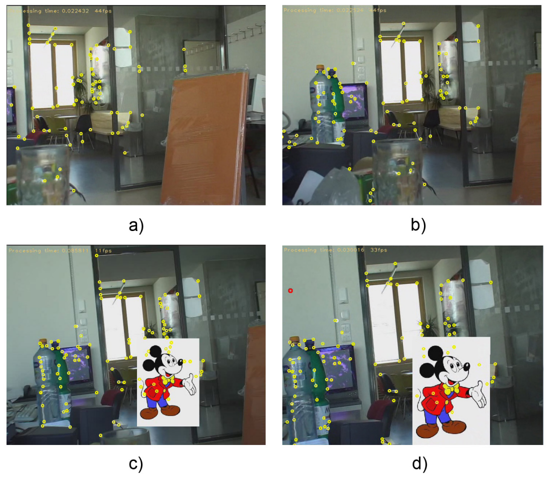

Figure 5: An homography found using RANSAC based algorithm. a) and b) Point matches used to determine homography. c) and d) Virtual object inserted to scene, the motion between views is camera zoom.

4 Limitations

As mentioned earlier in the text this tool deals only with location-fixed cameras. This is the greatest limitation for wider usage areas when there is a moving camera around/in a scene.

5 Technical specifications

This software requires Intel OpenCV [2] (computer vision library package) installed on target computer, or specific OpenCV libraries (cv100.dll, cxcore100.dll, highgui100.dll) located in software directory.

For computation with source video of resolution 640x480 at speed 20-50 fps, we recommend at least:

- Processor: 2.4 GHz or better
- Memory requirements: 512MB RAM
- Operating System: Windows

6 Package content

- Camera localization software
- Software user instruction
- Example of input data
- Demonstration video

References

[1] R. Hartley and A. Zisserman. Multiple View Geometry in Computer Vision. Cambridge University Press, Cambridge, UK, 2000.

[2] Wikipedia: Corner detection, url: http://en.wikipedia.org/wiki/Corner_detection.

[3] G. Welch and G. Bishop. An Introduction to the Kalman Filter, University of North Carolina, 1995.

[2] Intel OpenCV, available at http://opencv.willowgarage.com/wiki/