

Mobile 3D Visualization

Balázs Tukora

Department of Information Technology, Pollack Mihály Faculty of Engineering,
University of Pécs
Rókus u. 2, H-7624 Pécs, Hungary, e-mail: tuxi@morpheus.pte.hu

Abstract: Recent developments make it possible for the mobile devices without GPS system (such as mobile phones and PDAs) to determine their spatial situation on the visual information given by the embedded camera, complete the environment with virtual objects, and display the created world. Another feature of them is the ability of the continuous on-line connection with remote databases or expert systems. On the basis of all this we can create portable virtual instruments which give the technicians helpful additional visual and technical information on the spot, even outdoors, far from any wired connections. The idea of amplifying the environment with virtual objects is not a new concept, let just think of the countless 3D designing and visualizing computer applications. However the limited performance of the wireless devices requires developing faster and more effective algorithms for this purpose. This paper discusses the arising problems during the development and gives proposals for solving them.

Keywords: mobile technology, camera calibration, 3D reconstruction, single view modeling

1 Introduction

Technicians and engineers often visit working places with only a plan in hand and make their job relying on their imagination and spatial vision. Up to now there haven't been such portable visualization devices which help them localize hidden equipments and building parts at the place, i.e. underlying pipes or electric systems.

In the Virtual Instrument Project wireless devices (mobile phones or PDA-s) make the hidden objects visible. The exact situation of the objects is indicated by a reference figure. When the built-in camera of the phone is pointed to the figure, the equipments appear on the display with correct orientation and position (see Fig. 1, where a simple pipe-node is displayed, with helpful signs which show the flowing direction of the fluid). They can be walked round, and seen from every sides.



Figure 1
Virtual Instrument Project

In addition, the technicians can intervene in the functioning of the displayed equipments, as the mobile phones can set up remote connection with the system managing servers. The process of changing can be displayed as well (see Fig. 2, where the path of the fluid is changed).

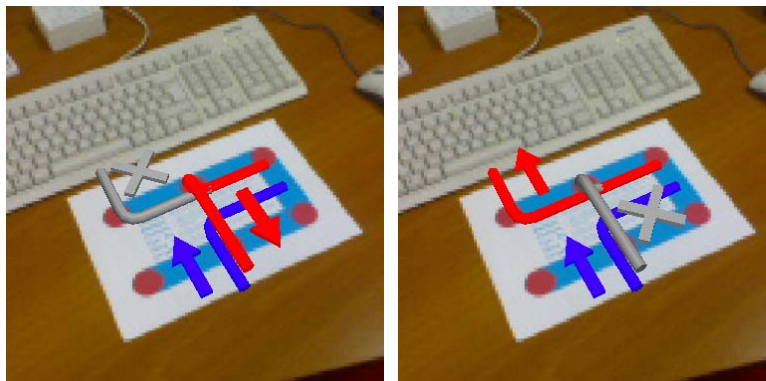


Figure 2
Changing parameters

2 The Application

For the introduced purposes a wireless Java application has been created. The two main tasks of the application are: visualization and communication. The visualizing part merges the real and virtual worlds according to the view of the reference figure, in details:

- it safely identifies some reference points on the picture taken by the camera,

- relying upon these findings computes the spatial location and orientation of the device (i.e. does the camera calibration),
- puts the 3D virtual objects into the scene,
- and visualizes the results on the display.

The communicating part keeps the connection with remote servers and takes care of the data input-output processes.

2.1 Locating the Reference Points

The application's first task is taking a snapshot of the surroundings and recognizing some elements on the picture which give information about the spatial relation of the device and the environment. It would be the best method if we could recognize the common objects of the locality, e.g. building parts. At the present time this would be too big work, especially for such limited performance devices like mobile phones. The solving of this problem is placing reference points into the surroundings, which must comply with the requirements of easy identifying and distinguishing from other parts of the picture.

The color of an object in its genuine neighborhood is continuously altering. According to the current part of the day and the weather it's getting lighter and darker, and it is changing the shade of color. The mobile phones' cheap automatic embedded cameras just increase this symptom, as they make averaged light adjustment on the whole picture: thus the same red dot can be seen almost black on a white piece of paper, whereas bright pink on dark background.

As the Java specifications define, the snapshots of the embedded camera are presented in RGB format. This means that the color of every pixel is given by three values: the volumes of red, green and blue color components. The most distinguishable colors are in which one of these components is much stronger or weaker than the other two. These are the primary colors: the clear red, green, and blue, and their mixtures, as yellow, cyan and magenta. Besides there are two other well recognizable colors, in which the volume of the components are nearly equal and easily measurable: white and black [1]. If the reference points are from the listed colors, we have good chance finding them on the picture even in different lighting conditions.

Other objects, such as clothes or billboards can have the same color as the reference points, of course. With the object of clear identifying, the reference points should be ordered into well distinguishable patterns. The shape of the most convenient pattern can be define just after we have found the right camera calibration algorithm, and we know the needed amount of the reference points as well as their criteria of spatial positioning.

2.2 Camera Calibration

Camera calibration means that we compute the camera's spatial orientation and location on the basis of the 3D coordinates of the reference points and the 2D coordinates of the snapshots of the reference points. Let's assume that the spatial position of the points are defined in the *world* coordinate system, and meanwhile the virtual objects are stored in the device's memory in relation to the *virtual* coordinate system. If we get the position of the camera in the world coordinate system, and set these parameters as its position in the virtual coordinate system, the two coordinate systems become congruent, the "outer" and "inner" worlds are merged.

The image of the reference points on the display proceed by the rules of the perspective projection (see Fig. 3). In this case the spatial points, their images on the projection plain (e.g. on the display) are on the same line which is originating at the focal point of the camera. As the depth information is getting lost, the perspective projection is theoretically not an invertible process, if only we have more than one reference points and we know the spatial relation of them [2].

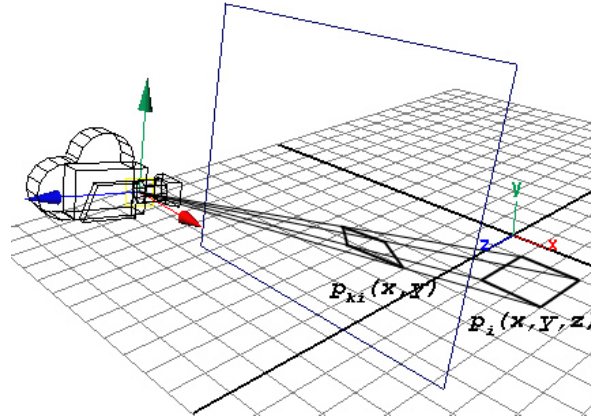


Figure 3

Perspective projection

The initial parameters of the calculation are the followings (according to Figure 3):

$$\underline{\underline{P}} = [\underline{P}_1, \underline{P}_2, \dots, \underline{P}_n] = \begin{bmatrix} P_{1x} & P_{2x} & \dots & P_{nx} \\ P_{1y} & P_{2y} & \dots & P_{ny} \\ P_{1z} & P_{2z} & \dots & P_{nz} \end{bmatrix}, \quad (1)$$

i.e. the 3 dimensional coordinates of the reference points in the world coordinate system,

$$\underline{\underline{P}}_k = \left[\underline{p}_{k1}, \underline{p}_{k2}, \dots, \underline{p}_{kn} \right] = \begin{bmatrix} p_{k1x} & p_{k2x} & \dots & p_{knx} \\ p_{k1y} & p_{k2y} & & p_{kny} \end{bmatrix}, \quad (2)$$

i.e. the 2 dimensional coordinates of the reference points in the display coordinate system.

The parameters which define the position of the camera are searched in the form of a homogenous transform matrix:

$$\underline{\underline{H}} = \left[\begin{array}{ccc|c} \underline{\underline{R}} & & & \underline{k} \\ 0 & 0 & 0 & 1 \end{array} \right] = \left[\begin{array}{ccc|c} \underline{e}_1 & \underline{e}_2 & \underline{e}_3 & \underline{k} \\ 0 & 0 & 0 & 1 \end{array} \right] = \begin{bmatrix} e_{1x} & e_{2x} & e_{3x} & k_x \\ e_{1y} & e_{2y} & e_{3y} & k_y \\ e_{1z} & e_{2z} & e_{3z} & k_z \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (3)$$

in which $\underline{\underline{R}}$ is the rotating matrix of the camera coordinate system in relation to the world coordinate system, in details:

$\underline{e}_1, \underline{e}_2, \underline{e}_3$ are the unit vectors of the camera coordinate system in relation to the world coordinate system, and \underline{k} is the vector from the origin of the world coordinate system to the origin of the camera coordinate system (see Fig. 4).

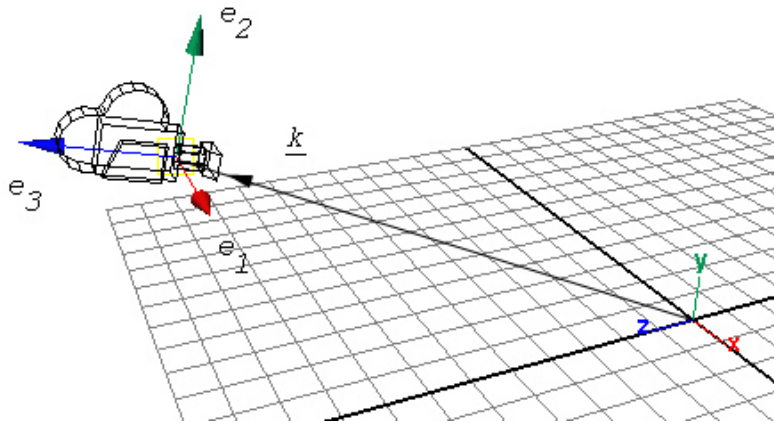


Figure 4
Camera coordinate-system

Thus the main task is finding such a minimal set of reference points with which the elements of the $\underline{\underline{H}}$ homogenous transform matrix can be computed.

2.3 The Algorithm

The elements of the \underline{H} matrix are not independent. We can write down all the 16 values as the function of seven parameters: the focal length, the 3 spatial coordinates and the rolling, pitching and yawing angles of the camera in relation to the world coordinate system. These values can be computed as the iterative solution of a non-linear equation-system with seven unknown parameters [3], however the capacity of the mobile devices is still too weak for this purpose. We must find a better way.

The searched parameters can be determined if we know the position of a given rectangle which is placed on the x-y plain of the world coordinate system [4]. For this computation we need the knowledge of the focal length as well (i.e. the distance of the focal point and the projection plain). This fact prevents using this algorithm for our aim, because the focal length values of every potential mobile devices are not available. Previous camera calibration would solve the problem (e.g. the snapshotting of a reference figure from a given distance), but in this case the system would lose its flexibility and portability.

The popular 3D design applications (such as 3D Studio Max or Maya) compute on the basis of six reference points, without the knowledge of the focal length. There is nevertheless a strong restriction: the points must be located on two different plains – thus *one* reference figure on the pavement would not be enough.

If we want to write an algorithm which needs reference points on only one plain and doesn't require the knowledge of the focal length, we have to exploit some hidden information of the picture taken by the camera. The *vanishing line concept* is used for a long time for determining the plain of texts or patterns snapshotted by non-fixed cameras [5]. This kind of method could be used for our purposes.

It is well known that the parallel lines on the same plane are getting converging in case of perspective projection (see Fig. 5). They meet at the horizon-line (or vanishing line) of the plain. The meeting points are called vanishing points. In general two vanishing points mark out the horizon.

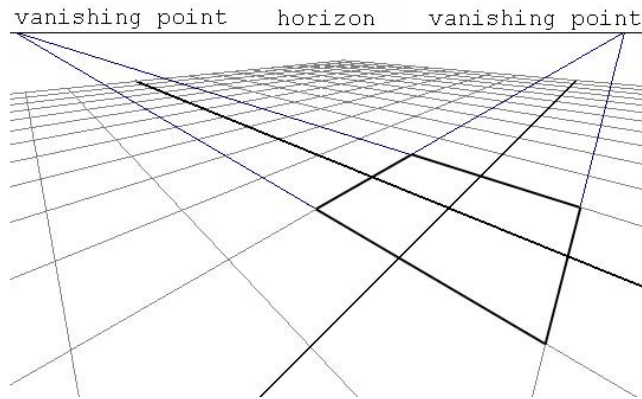


Figure 5
Vanishing points

As the location of the horizon-line depends only on the orientation angles of the camera, the rolling, pitching and yawing angles can be computed. The result is independent from the focal length, since the changing of the focal length doesn't change the inner relations of the picture. (The alternating distance between the projection plain and the focal point causes just scaling). And besides if we know the distance of two reference points on the pavement, we can compute the position of the device according to the figure.

Relying upon these findings only four reference points are needed on the same plain. The four points are equal, thus we don't know which space-quarter we are observing from. If we want to know, one of the vertices must be marked.

When the bottom and upper sides of the rectangle are getting parallel to the horizon, one of the vanishing points is moving into the infinity aside. The very little angles on the picture can cause big error in reading, so we should avoid this situation, in spite of the fact that the algorithm can be adapted to a single vanishing point as well.

Though the results are independent from the focal length, the precision of the computing is not so. In case of little view of angle (big focal length) the converging of the lines are getting weaker, the vanishing points are getting farther, increasing the possible error in reading. Fortunately the fixed-focal-length automatic cameras use big angle of view.

2.4 The Reference Figure

As mentioned above, four reference points give adequate result, but in some cases considerable error can occur. This is why such a pattern has been chosen which consists of six points instead (see Fig. 6). Six dots produce two pairs of converging lines (marked with “A” on the figure) and two other pairs rotated with 45 degrees (marked with “B”) in the world coordinate system. This arrange results at least two vanishing points close to the horizontal center of the image in every case, even if 2 sides of the rectangle are perpendicular to the viewer and thus one of the vanishing points is in the infinity aside.

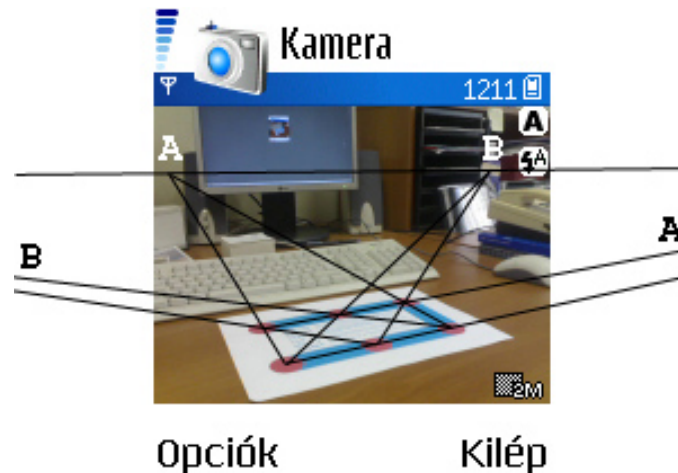


Figure 6
Reference pattern

The reference points are arranged into a U-like shape. This layout let us clearly identify each vertices from any viewpoints. Red and white colours are used, because they are easily recognizable even in various light conditions.

2.5 The 3D Virtual World

The 3 dimensional virtual world consists of the 3D model of the visualized objects, two directional lights and a camera (see Fig. 7). The objects are made from about 1000 polygons. Bigger number of polygons would far too reduce the speed of the devices. The world was created with Maya and 3D Studio Max and then it was exported into .m3g file-format. This type of file can be processed by the mobile phones, as it's defined by the JSR-184 Java Specification Request. Recent 3D developing tools let us convert objects and worlds from the main CAD systems, thus basic industrial and architectural models are importable easily into the application.

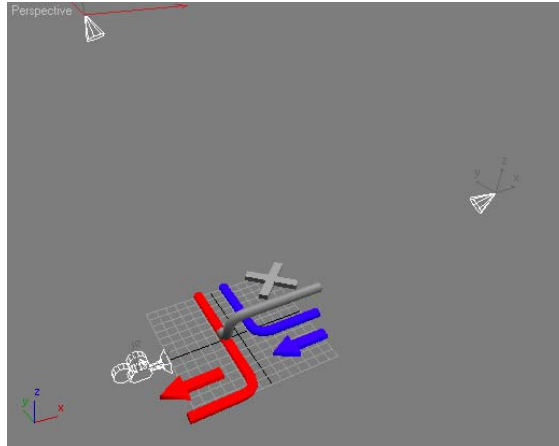


Figure 7
The virtual 3D world

2.6 Composition and Displaying

The final tasks of the application are the composition of the inner and outer worlds, and the displaying of the results. At first the computed parameters are assigned to the virtual camera, thus the objects gets seen from the right direction and distance. Secondly the snapshot picture taken by the camera is put into the background and the objects are projected onto it, so they look like the part of the surroundings (see Fig. 8).

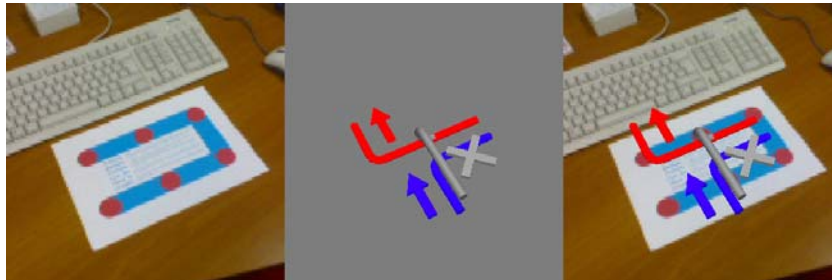


Figure 8
Composition and displaying

2.7 Communication

The JSR-172 Java Specification Request defines the methods of the client-server communication in wireless environment, let the developers create XML-based

communicating parts for their applications. Java and XML are device independent, so it's quite easy to create the needed client and server functions with any of the common software development tools.

Conclusions

The graphical quality of the wireless devices is still very limited. Though the application is continuously checking the movement of the device and refreshing the display contents, less than 0.8 – 1 frames per second can only be achieved. The main reason is the slowness of the built-in camera, since not a video-streaming but serially snapshotted pictures are processed. The virtual objects must consist of limited number of polygons, and the fine details often disappear on the little display. The technology however is developed rapidly: since the first début of the 3D-compliant mobile phones in 2004, the benchmark index which shows the 3 dimensional representation abilities increased tenfold.

Acknowledgement

The research work has been developed in the frame of the project "Utilization of the infocommunication technologies in the region" No. GVOP-3.1.1.-2004-05-0125/3.0.

References

- [1] Gonzalez, R. and Woods R.: Digital Image Processing, Second Edition, Pearson Education, 2002
- [2] Szirmay-Kalos L. (editor): Theory of Three Dimensional Computer Graphics, Publishing House of the Hungarian Academy of Sciences, 1995
- [3] Glaskó G.: Vision-based Camera Matching Using Markers, 4th Central European Seminar on Computer Graphics (CESCG 2000)
- [4] Haralick R. M.: Determining Camera Parameters from the Perspective Projection of a Rectangle. Pattern Recognition, Vol. 22, No. 3, 1998, pp. 225-230
- [5] Clark P., Mirmehdi M.: On the Recovery of Oriented Documents from Single Images. Proceedings of ACIVS 2002 (Advanced Concepts for Intelligent Vision Systems), pp. 190-197