OPTIMAL SOFTWARE-BASED PROJECTOR ALIGNMENT

Pablo A. S. Baier   Paulo C. P. Carvalho
IMPA – Instituto Nacional de Matemática Pura e Aplicada
Estrada D. Castorina, 110
22460-320, Rio de Janeiro, Brazil
{baier, pcezar}@visgraf.impa.br

Abstract

We present a software-based method for alignment of projectors, both for side-by-side (mosaic) and stereo projections. The projectors are assumed to be positioned in such a way that their axes are not perpendicular to the screen plane and their projection areas do not adjust exactly to each other. We consider the problem of modifying each projected image in such a way that the projected areas become perfectly adjusted rectangles, without any physical repositioning. We model the problem of finding the largest possible projection rectangles as a linear program and show how to appropriately pre-warp each image so that they project exactly onto these optimal areas.

Keywords

alignment, projective transformation, warping, linear programming

1. INTRODUCTION

The simultaneous use of several projectors, either to produce image mosaics or stereo views, is increasingly present in applications. The resulting projected view should present a sense of continuity. However, distortions due to physical characteristics and positioning of each projector make it difficult to manually adjust the projectors in order to obtain satisfactory results. This problem can be solved by using especially designed projectors, with electro-mechanical controls (as those used in flight simulators, for instance). However, they require constant alignment and calibration; also, they may be too costly for the end-user ([Raskar98], [Raskar02]).

In this work, we present a method for optimal software-based alignment of two (or more) projectors, requiring no physical adjustment. Based on images captured by a camera, we compute projective warpings that are applied to each image to be projected, in order to obtain correct projections that occupy the maximum available area on the display surface, which is assumed to be planar.

No previous knowledge about intrinsic or extrinsic projector parameters is required. Also, the projectors may have different characteristics (such as focal distance or pixel density). However, we restrict ourselves to the geometrical adjustment of the projectors, not treating crominance or luminance correction.

2. CAMERA AND PROJECTOR MODEL

Figure 1 shows what happens, in general, when using two projectors to form an image mosaic (in the examples in this section, we consider the case in which there is no overlap between the images; the general case is treated in section 4). The images were computed to be displayed side-by-side, but they do not ajust precisely due to projector misalignment. Also, each projected area is not, in general, a perfect rectangle, due to deformations introduced by the projectors. We use a pin-hole model for the projectors, which ignores lens-distortion; in this case, the deformations induced by the projector are given by projective transformations Π_{P_1} and Π_{P_2} of the plane. In the remaining sections we will show that we can apply appropriate projective warpings W_1 and W_2 to the original images, in such a way that the projected images (that are obtained by applying transformations $\Pi_{P_1} W_1$ and $\Pi_{P_2} W_2$ to the original images) fit exactly side-by-side.

In order to estimate transformations Π_{P_1} and Π_{P_2}, we project a known pattern through each projector and capture the result in a digital camera. The captured images, however, are distorted themselves, as shown in Figure 2 (a). Assuming again a pin-hole model, the camera deformation is given by a projective transformation Π_C of the plane.

Once we have estimated Π_C, we can apply Π_C^{-1} to the captured image in order to obtain correct, undistorted

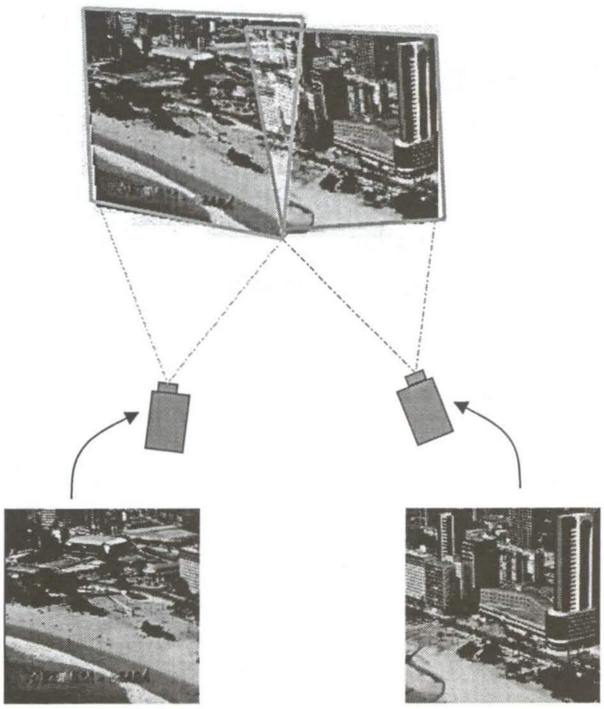

Figure 1. Side-by-side projection without adjustment

images, that in turn can be used to estimate projector deformations Π_{P_1} and Π_{P_2}.

It should be noted, however, that estimating Π_C requires establishing correspondences between points in the image and points of the screen plane. This implies in imposing an absolute frame of reference on the screen plane and performing direct measurements on that plane. This may not be feasible, desirable or necessary. Instead, we may adopt an arbitray frame of reference and consider it to be correct. For instance, we can use a printed chess-board pattern such as the one shown in Figure 2. We may also adopt the image projected by one of the projectors as being correct. In any case, by using such a frame of reference we can compute a projective transformation capable of undistorting the camera image with respect to the chosen frame of reference. This actually introduces a new reference space that we call *corrective space*. Figure 2(b)illustrates the result of mapping the image of part (a) into corrective space.

In summary, the transformations described above involve the following spaces:

- **Screen (or real) space**: It is the space of the projected images, that in turn are captured by the camera.
- **Image space**: It refers to the spaces of the images before being projected or after being captured by the camera. In both cases, images are distorted; in the case of the cameras, this is due to camera positioning; in the case of projectors, these are deformed on purpose, in such a way that the final projected images are correctly displayed.
- **Corrective space**: It is an ideal space, corresponding to captured images after correction and to projected images before deformation is applied.

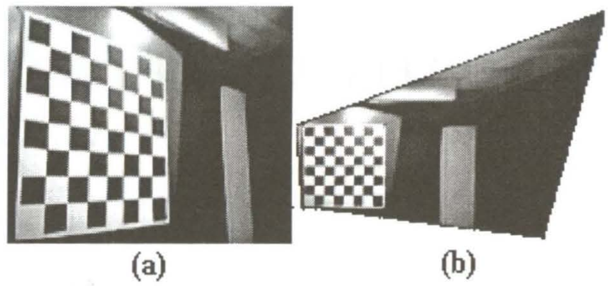

Figure 2. (a) **Captured image (in the image space).** (b) **Image shown in corrective space, showing the correct geometry of objects.**

Figure 3 depicts these spaces and the projective transformations that establish their relationship. Instead of directly estimating Π_P and Π_C, we estimate their proxies T_P and T_C. This process is described in more detail in the next section.

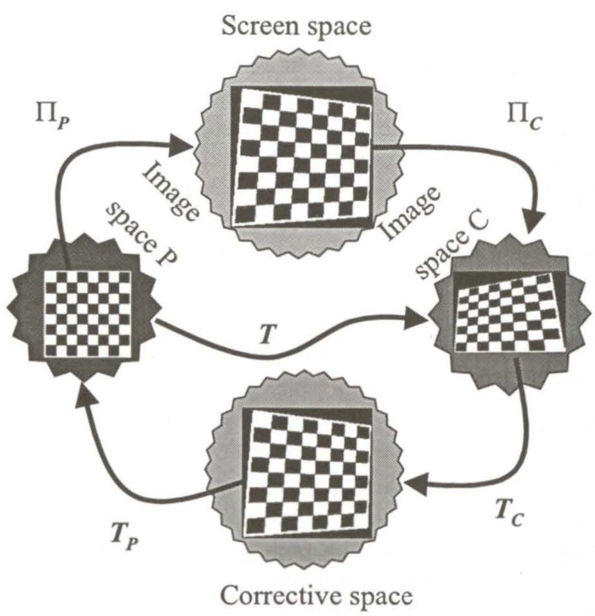

Figure 3. Projection and capture scheme

3. CAMERA AND PROJECTOR CALIBRATION

In general, calibrating a camera (or a projector) means to determine its intrinsic and extrinsic parameters

([Fauger93]). In this work, however, since all objects of interest lie on the plane of the projection screen, projector and camera calibration only involves the determination of projective transformations T_P and T_C, respectively.

To calibrate the camera, we use a set of n points whose coordinates $\mathbf{p_i}$ in corrective space are known and locate the corresponding points $\mathbf{q_i}$ in the image. It is convenient to use the chess-board pattern shown in Figure 2. The corners have known coordinates in corrective space and their locations in the image can be automatically found (we used the Intel Open Source Computer Vision Library ([Intel00])). Observe that the chess-board pattern can be either printed on a surface or projected by one of the projectors. In the latter case, we adopt the projected image as "correct" and align the second projector with respect to it.

In any case, once the n pairs $(\mathbf{p_i}, \mathbf{q_i})$ have been determined, T_C is chosen so as to minimize the total squared distance between $\mathbf{p_i}$ and $T_C(\mathbf{q_i})$. That is, we solve

$$\min_{T_C} \sum_{i=1}^{n} \|\mathbf{p_i} - T_C(\mathbf{q_i})\| \qquad (1)$$

A way of solving (1) is given in [Carval98], based on using a sequence of linear least-square problems.

The same method is used to estimate T_P. We project a chess-board pattern through each projector. By establishing the correspondence between corners in corrective space and their locations in the captured image, we estimate the composite map $T = T_C^{-1} T_P^{-1}$ (see Figure 3). Since T_C has been estimated previously, we can compute T_P using $T_P = T^{-1} T_C^{-1}$.

4. PROJECTOR ADJUSTMENT

Once the projective transformations induced by the projectors have been computed, our problem becomes that of finding appropriate pre-warpings to be applied to each image, in such a way that the two projected images adjust correctly to each other.

Originally, as mentioned in section 1, the images are projected onto quadrilaterals $U = (u_1, u_2, u_3, u_4)$ and $V = (v_1, v_2, v_3, v_4)$, that are not rectangular and do not adjust exactly, as illustrated in Figure 4. In order to obtain the desired effect, we must find suitable rectangles R_1 and R_2, respectively contained in U and V, and apply appropriate projective transformations D_1 and D_2 to the original images so that they project onto R_1 and R_2. This is always possible, since any two quadrilaterals can be mapped to each other by means of a projective transformation. The basic idea is illustrated in Figure 4 for the case of stereo views (where $R_1 = R_2$). Below, we describe how to choose optimal rectangles R_1 and R_2 and how to find the warpings to be applied to each image so that they project onto R_1 and R_2, *respectively*.

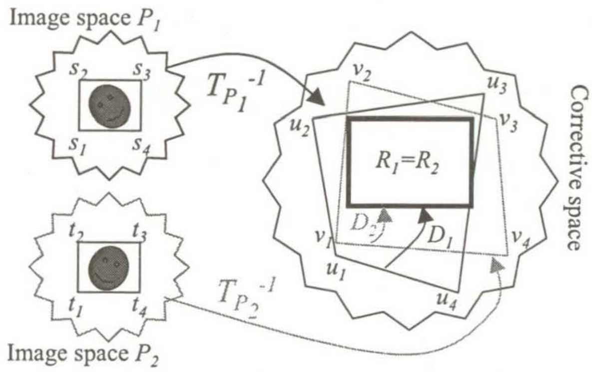

Figure 4. Finding the optimal projection area

4.1. Finding the largest projected rectangle

We assume that the two images are to be projected onto rectangles of the same size, with sides aligned with the horizontal and vertical directions (with respect to the frame of reference used to establish the corrective space) and having a known aspect ratio ρ. We further assume that the top and bottom sides of the rectangles lie on common straight lines and that their widths overlap by a known fraction μ, as illustrated in Figure 5. Observe that the case in which the rectangles exactly adjust to each other side-by-side corresponds to the case $\mu = 0$, whereas the case in which they coincide corresponds to the case $\mu = 1$. Among all pairs of axis-aligned rectangles $R_1 \subset U$ and $R_2 \subset V$, with aspect ratio ρ and overlap μ, we want to find those having maximum area.

The pairs of candidate rectangles may be parameterized by the coordinates (x, y) of the bottom-left vertex of the right-most rectangle and by their common width l (see Figure 5). The right-most rectangle R_2 has vertices (x, y), $(x, y + \rho l)$, $(x + l, y)$, and $(x + l, y + \rho l)$; the vertices of the left-most rectangle R_1 are $(x + (\mu - 1)l, y)$, $(x + (\mu - 1)l, y + \rho l)$, $(x + \mu l, y + \rho l)$, and $(x + \mu l, y)$.

Since the sought rectangle has fixed aspect ratio, maximizing its area is equivalent to maximizing one of its dimensions. Thus, the problem of finding the largest area rectangle can be posed as the problem of maximizing h subject to the conditions that $R_1 \subset U$ and $R_2 \subset V$. These conditions are equivalent to the constraints below:

1. $(x + (\mu - 1)l, y)$ lies to the right of vectors $\overrightarrow{u_4 u_1}$ and $\overrightarrow{u_1 u_2}$;

2. $(x + (\mu - 1)l, y + \rho l)$ lies to the right of vectors $\overrightarrow{u_1 u_2}$ and $\overrightarrow{u_2 u_3}$;

3. $(x + \mu l, y + \rho l)$ lies to the right of vectors $\overrightarrow{u_2 u_3}$ and $\overrightarrow{u_3 u_4}$;

4. $(x + \mu l, y)$ lies to the right of vectors $\overrightarrow{u_3u_4}$ and $\overrightarrow{u_4u_1}$;
5. (x, y) lies to the right of $\overrightarrow{v_4v_1}$ and $\overrightarrow{v_1v_2}$;
6. $(x, y + \rho l)$ lies to the right of vectors $\overrightarrow{v_1v_2}$ and $\overrightarrow{v_2v_3}$;
7. $(x + l, y + \rho l)$ lies to the right of vectors $\overrightarrow{v_2v_3}$ and $\overrightarrow{v_3v_4}$;
8. $(x + l, y)$ lies to the right of vectors $\overrightarrow{v_3v_4}$ and $\overrightarrow{v_4v_1}$.

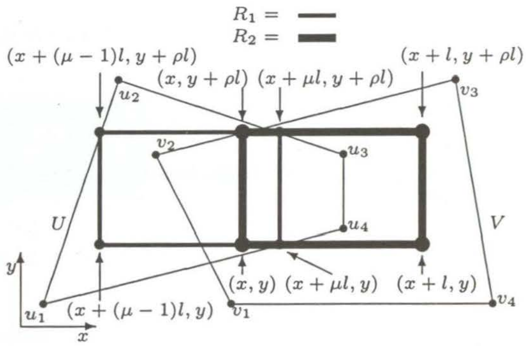

Figure 5. Side-by-side positioning of two rectangles

Observe that each one of the 16 constraints above is expressed as a linear inequality in x, y, and h. Therefore, the problem of finding the largest projected area is a linear program in 3 variables, that can be solved, for instance, using the simplex method ([Dantzig63]). The simplex method works well in practice even though potentially requiring exponencial time. Alternatively, one could use the linear time method of Meggido ([Meggi84]).

Figures 6 and 7 show examples where the formulation above was applied. Figure 6 illustrates a case of stereo view, with $\rho = 1.0$, whereas Figure 7 shows a side-by-side projection, with aspect ratio $\rho = 0.75$, and overlap $\mu = 0.2$.

4.2. Warping the images

Once the rectangles R_1 and R_2 associated with each projector are found, the warpings to be applied to the images are the composite mappings $W_i = T_{P_i} D_i T_{P_i}^{-1}$, $i = 1, 2$, where D_1, D_2 are projective transformations that map quadrilaterals U and V into rectangles R_1 and R_2, respectively (see Figure 4).

Although the method was originally devised for two projectors, it may be useful even for the case in which there is only one projector, and the purpose is to correct the distortion caused by its wrong positioning with respect to the projection screen. Notice, also, that the projective warpings above can be easily inserted at the end of the rendering pipeline, generating

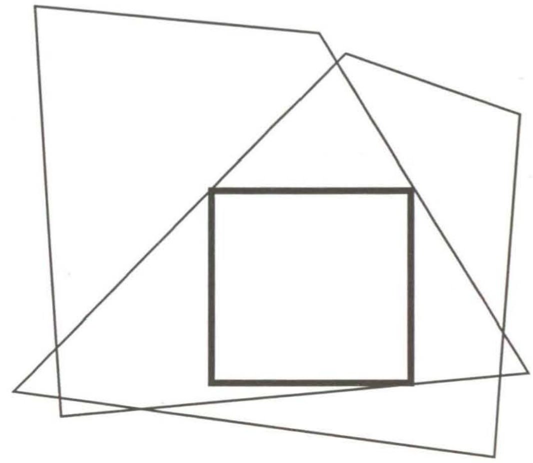

Figure 6. Largest rectangle for stereo case ($\rho = 1, \mu = 1$)

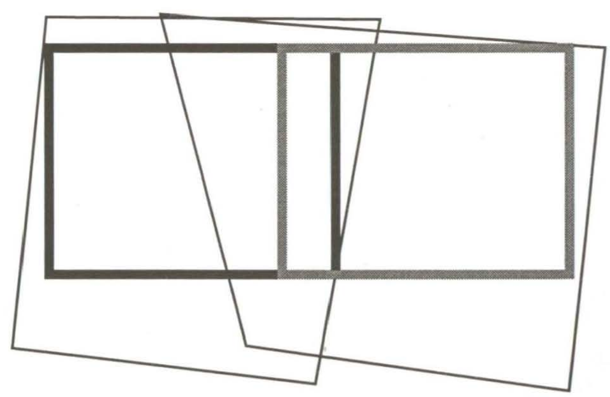

Figure 7. Largest rectangle for mosaic case ($\rho = 0.75, \mu = 0.2$)

geometrically correct projected images at little or no additional cost.

We should point out that straightforward texture mapping works best when the deformation introduced by the projector skew is not too severe (i.e., when the positioning of the projectors is approximately correct). In the case of large deformations, a more sophisticated scheme (for instance, using mipmaps) should be employed.

5. RESULTS

In the examples below, we used a Creative WebCam CT6840 (352 × 288 resolution) and two projectors CTX EzPro610 (800×600 resolution). In the first example, we applied our method in the situation in Figure 8. The images to be projected were obtained by splitting an existing image in two equal-sized parts, without any overlap. The images in Figure 8 are both shown in the corrective space (that is, the images are corrected for camera deformation). The top part of the figure shows the actual projected images. The bottom part shows just the image axes and border, to make it easier to observe the projector misalignment. Figure 9 shows the result of applying the method. The largest rectangles were computed using the algorithm in section 4 and the images were warped so that their new projections coincide with these rectangles. Although the images ajust well to each other, the seam line is quite noticeable. For better practical results, some degree of overlap is necessary, together with an equalization procedure for crominance and luminance.

In the second example (Figure 10), identical images were projected through each projector (in practice, this kind of adjustment would be used for stereo projections, to be seen through special glasses). We show the projections of the actual images on the left and the projections of the axes and borders on the right. All pictures were captured by the digital camera and are shown in corrective space. Figure 11 shows the results provided by the algorithm. We can see that the axes adjust quite well to each other, but the same does not happen to the borders. This is due to the pin-hole model used for camera and projectors. Actually, in both cases we have distortions caused by the lenses, which are more severe at the image border. This slight registration error is not severe for stereo views, since the human brain is capable of compensating for it, when using stereo glasses.

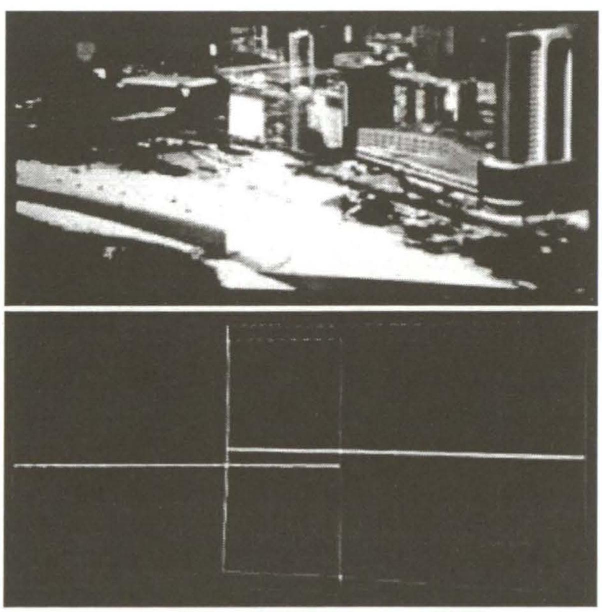

Figure 8. Side-by-side projection without adjustment

6. CONCLUSION AND FUTURE WORK

We presented a method for virtual alignment of two or more projectors that requires little user intervention. In particular, time-consuming and error prone manual adjustment of the projectors is unnecessary. The user has just to position the projectors so that they approximately adjust to each other. The method automatically computes the maximum projection area that can be obtained from both projectors and pre-warps each image in such a way that they project exactly onto this maximum area projection.

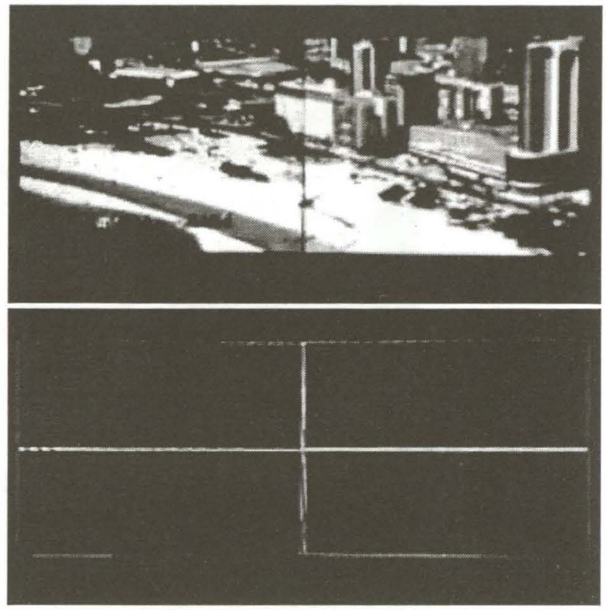

Figure 9. Side-by-side projection with adjustment

The method is based on capturing images on a digital camera. By choosing an appropriate frame of reference, all captured images are mapped to a corrective space, where images are assumed to be correct. Camera and projectors are calibrated by establishing correspondences between a set of points having known coordinates and their locations in the image (in the sense of this paper, calibrating the camera and the projectors means to determine the projective transformations that map the captured or projected images to their appearance on the screen plane).

Once all pieces of equipment are calibrated, a maximum projection area is determined, using a linear programming formulation. The original images are pre-warped in such a way that their final projection occuppy exactly that maximmum area.

The method produces good results, especially for stereo projections. For the case of side-by-side projection, the practical use of the method would require, in most cases, adjusting for differences in the luminance and crominance between the projectors, as in [Pham95]. The method would probably benefit from a more precise camera and projector model, including, for instance, the radial deformation caused by camera and projector lenses.

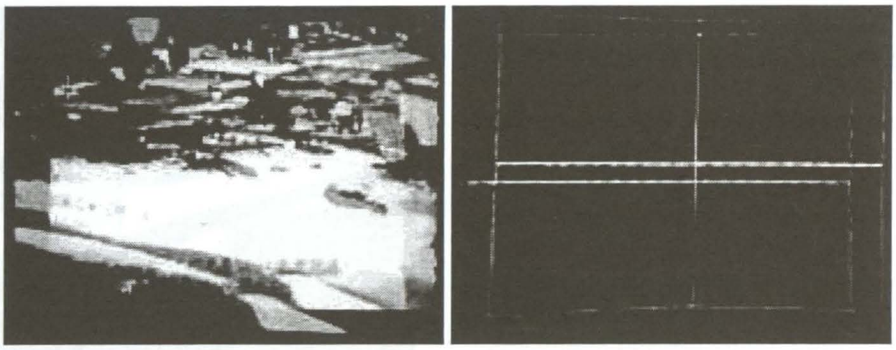

Figure 10. Stereo projection without adjustment

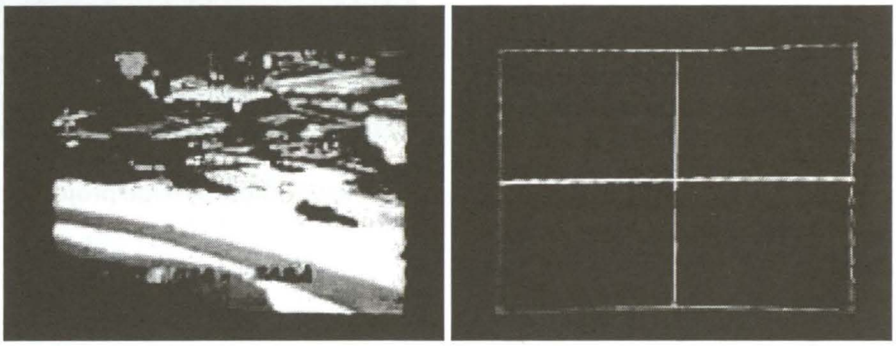

Figure 11. Stereo projection with adjustment

7. ACKNOWLEDGEMENTS

This work was done in the VISGRAF laboratory at IMPA and in the TECGRAF laboratory at PUC-Rio. We thank TECGRAF for providing the environment and the equipment for most of this research. VISGRAF is supported by grants by CNPq, FAPERJ and IBM-Brasil.

8. REFERENCES

[Carval98] Carvalho, P. C. P., Szenberg, F. & Gattass, M.: *Image-base Modeling Using a Two-step Camera Calibration Method.* IEEE Transactions on Robotics and Automation, 3(4), pp. 388–395, 1998.

[Dantzig63] Dantzig, G. B.: *Linear Programming and Extensions.* Princeton University Press, 1963.

[Fauger93] Faugeras, O.: *Three-Dimensional Computer Vision: a geometric viewpoint.* MIT Press, 1993.

[Intel00] Intel Corporation: *Open Computer Vision Library.* `http://www.intel.com/research/mrl/research/opencv/`.

[Meggi84] Meggido, N.: *Linear programming in linear time when the dimension is fixed.* Journal of the Association for Computing Machinery **31** (1984) 114-127.

[Pham95] Pham B. & Pringle, G.: *Color Correction for an Image Sequence.* IEEE Computer Graphics and Applications, 1995.

[Raskar98] Raskar, R., Welch, G. & Fuchs H.: *Seamless Projection Overlaps using Image Warping and Intensity Blending.* Fourth International Conference on Virtual System and Multimedia, Gifu, Japan, 1998.

[Raskar02] Raskar, R., van Baar, J. & Chai, J. X.: *A Low-Cost Projector Mosaic with Fast Registration.* Proceedings of the 5th Asian Conference on Computer Vision, Melbourne, Australia, 2002.

[Tsai87] Tsai, Y.: *A versatile camera calibration technique for high accuracy 3D machine vision metrology using off-the-shelf cameras and lenses.* IEEE transactions on Robotics and Automation, 3(4), 1987.