IMM
DEPARTMENT OF MATHEMATICAL MODELLING

Technical University of Denmark
DK-2800 Lyngby-Denmark

15.12.96

CORRECTING FOR OFF-CENTRE, ANAMORPHIC, NON-PLANAR PROJECTION

Per Skafte Hansen
Svend B. Sørensen
Ole Mærsk-Møller

TECHNICAL REPORT

IMM-REP-1996-30

IMM

ISSN 0909 6264

Trykt af IMM - DTU

Correcting for off-centre, anamorphic, nonplanar projection

IMM, Numerical Analysis Group

Per Skafte Hansen, IMM
Svend B. Sørensen, Dept. of Physics
Ole Mærsk-Møller, Dept. of Planning

Technical University of Denmark

Abstract:

Projecting an image onto a large spherical dome becomes a data-fitting problem when the projection equipment no longer behaves like the ideal pinhole lens, *i.e.* producing a simple perspective. For a specific dome and lens, placed in the Planetarium in Copenhagen, the image shown is distorted by the combined effects of the curvature of the dome, the anamorphism of the lens and the off-centre position of the projector. To correct the image seen, a pre-distortion must be calculated. In the present work, the effects of the projection are split in an *ad hoc* but geometrically sound fashion, and the pre-distortion is modelled as a perturbation. A specific expression for the perturbation function is obtained by a least-squares fitting to data obtained by theodolite measurements.

Authors' Preface:

The problem discussed in this report was posed by the Copenhagen Planetarium and soon turned out to require cross-disciplinary work. The solution method is almost purely practical: stereophotogrammetric measurements were made in the dome after closing hours — the first attempt foundered owing to the extreme lighting requirements — and the computational analysis subsequently carried out mostly as an experiment to verify the *ad hoc* model of the projection: the lens is treated as producing an ideal perspective of a slightly distorted image, the distortion taking place in an intermediate parameter space obtained by splitting the transformation.

The actual result of this extended exercise is simply two sets of coefficients. Using these, pre-distortions can be computed and their visual effect can be estimated, a problem not considered here.

The wider applicability of the method will largely depend on the geometry of whatever object serves as a screen: anything approximately spherical, cylindrical or conical allows for split transformations as simple as those employed here; and the least-squares fitting of tensor product Chebyshev polynomials is an elementary and robust process. On the other hand, detailed analyses of error distributions and the sources of errors are likely to become unduly involved

1 Introduction

The "real" or "physical" problem to be discussed in the following is this: in the dome hall of Tycho Brahe Planetarium (in Central Copenhagen) is a wide-angle projector, capable of projecting a circular slide of approximately 85mm in diameter onto the dome. However, the image undergoes a noticeable distortion, for which a computational method of correction is sought.

Indeed, the lens system of the projector spreads the light rays of the lamp over an angle of approximately 200°, which implies that the projection is not a perspective — another way of expressing this is that in the construction of the lens system, the conventional pin-hole ideal has been abandoned in favour of an *anamorphic projection*.

No specific information is available about the lens system, and it is considered very difficult to measure or compute its anamorphic effects directly.

However, the anamorphic effects are partly compensated for, partly enhanced by the screen, which is not planar but, to a relatively high degree of accuracy, spherical. "Curved as a sphere" would be a more precise statement, since the dome is slightly less than a hemi-sphere. Henceforth, we shall use the expression "the dome sphere" to refer to a virtual sphere, of which the actual dome is the physical part.

To top this, the projector is not placed in the centre of the dome sphere, for the simple reason that this is reserved for the planetarium proper, *i.e.* the highly complex projection equipment that reproduces the motion of planets and stars.

The problem is therefore defined by very practical conditions and considerations: the lens and the dome are there, images can be projected and the distortion measured directly from the dome. Given an equally practical definition of an "undistorted projection", is it then possible to pre-distort the planar image is such a fashion that a spectator gets the undistorted view?

2 The data and conditions

The problem will be solved using simple least squares data-fitting techniques, combined with a bit of low cunning (the reader may prefer to use phrases such as "common sense" or "mathematical maturity" in oral communication of the results). The actual data consist of high precision (5 decimal digits assumed correct) measurements made on the slide — using a microscope — and in the dome — using theodolites:

A slide was prepared, app. 85mm by 85mm, with a coarse grid in white (*i.e.* clear) on black on it. The grid itself was not drawn with great precision — in fact, it is not even regular in any sense; but the lines were kept as thin as possible to give the most precisely defined ("sharpest") points upon projection. Around 50 grid intersection points were distributed inside a circle with diameter app. 80mm. The slide was placed on the glass plate that serves as a slide mount for the projector and its position manually adjusted, so that the 80mm circle, upon projection, appeared as a parallel to the rim of the dome. The visible grid intersection points were then measured. The raw data thus consist of: a set of two-dimensional cartesian coordinates, henceforth denoted X, Y for the grid intersection points on the slide (with a reference point relatively far outside the slide area); and a set of three-dimensional cartesian coordinates, henceforth denoted x, y, z for the projected grid intersection points with a reference point in an arbitrary position inside the dome sphere.

The intersections on the slide were measured, relative to a two-dimensional coordinate system defined by points outside the slide area, using a ZeissJena stereo comparator to a precision of 1 micron .

The points on the dome were measured by a three-dimensional intersection using theodolites. Under normal conditions, this would be a rather trivial task, but owing to the special circumstances, great care was necessary:

Although the slide projector is lit by a high-powered, highly efficient HMI-lamp, the intersections on the dome were very difficult to see. Light in the theatre had to be switched off completely in order to make the thin lines of the slide visible, and this caused an unforeseen additional problem: The diaphragm ("the crosshair") of the theodolite is normally black and hence visible on a typical background, such as a landscape in daylight; but here the diaphragm was overlaid on a black background, and consequently became invisible. Most theodolites have means of illuminating the diaphragm to allow measurement of stars in the night for astronomical azimuth determination. However, when the diaphragm was thus made visible, this light was too strong and completely blocked the view of the intersections on the dome.

Another approach had to taken. A small laser pointer pen (of the type that a university lecturer would use to point out details of a slide presentation) was fitted to the telescope of the theodolite instead of the diopter and aligned in such a way that the laser beam grossly coincided with the crosshair of the telescope. This mounting was of course highly eccentric relative to the vertical axis of the theodolite and therefore an error source in the measurement of both the horizontal and vertical angles. However, remembering the basics in measuring with theodolites, this eccentricity will be annulled if the measurements are carried out with readings of the theodolite both facing right and facing left. Normally, a difference between the two readings of 0.01 gon is to be expected, but owing to the eccentricity differences of up to 0.9 gon were observed. Taking the mean of the two measurements pr. point eventually produced three-dimensional Cartesian coordinates having an RMS error less than 5.0 mm.

As for the concept of "undistorted", at least two interpretations suggest themselves: one might *either* attempt to let the projected image imitate a picture painted directly on the dome, in which case the original slide — before any correction for anamorphism and the off-centre position of the projector — should probably be a genuine perspective of this spherical picture; *or* to let the projected image simulate that of a very large planar picture, say, painted onto a tangent plane of the dome.

In the latter case it is quite obvious from the geometry of the problem, that only *one* spectator in the dome can see the resulting image as completely distortion-free. This would be equally true even if the picture were indeed painted onto such a plane, since a perspective can only be seen correctly from one point, a fact that is often forgotten and in practice (*e.g.* the cinema) overruled by elementary commercial imperatives.

The algorithm presented below aims at the second interpretation of "undistorted" and places the distiguished spectator in the centre of the dome sphere, a democratic solution in every way since, as explained above, that particular position is occupied by immobile and expensive equipment. However, with very modest adjustments the algorithm can cater for the first interpretation as well, should the need arise.

3 The algorithm(s)

The general problem of data-fitting is extremely ill-posed in the sense that it involves fitting a continuum to a discrete set. Therefore, it is conventionally assumed that the underlying *model* is smooth and that the errors in the data distributed in a fashion concordant with the *norm* used to estimate their magnitude during the fitting process. So conventional have these assumptions become, that they are probably better characterized as "implicit". Here, we shall use least squares techniques and make no attempt to justify this particular choice.

There are two (groups of) algorithms: one is the final projection(s), to be performed repeatedly; the other is the computation of certain sets of coefficients to be used *in* the projection process(es) — this need only be done once.

In the description of the algorithms, the following conventions have been used throughout:

- The dome is scaled to a unit sphere

- A right-handed spatial coordinate system is introduced with

  x representing "right"
  y representing "down"
  z representing "forwards"

- The virtual image plane is the tangent plane to the sphere in the point with spatial coordinates (0, 0, 1)

- A virtual image plane coordinate system is introduced with

  origin in the point of tangency
  X aligning with x
  Y aligning with $-y$

- Parameters (u, v) are introduced, so that on the dome

  $x = \sin u \cos v$
  $y = - \sin v$
  $z = \cos u \cos v$

Hence, for numerically small values, u represents X and v represents Y. Within the range of validity of the inverse trigonometric functions, we thus have $u = \text{atan}(x/z)$ and $v = \text{asin}(-y)$.

- The film image lies within $[-1;1]^2$ with coordinates ("parameters")

  p representing "right"
  q representing "up"

3.1 The projection algorithms

It may not be immediately obvious, but it is necessary — or at any rate highly advantageous — to have two different projection algorithms:

If the final image on the film is to be drawn using lines or curves defined by vertices, which can *a priory* lie "anywhere" within the limiting circle, then these vertices are projections *onto the film* of vertices found in the virtual image. In other words, this projection takes values *from* the virtual image in space *onto* the film.

If the final image on the film is to be drawn as a collection of *pixels* in a *raster*, then their coordinates are integer pairs and their positions bound to the raster grid. Therefore, it must be possible to project from such a raster point *onto the virtual image* in order to sample the intended colours there, as is customary in *ray-casting* — see any good book on computer graphics. (An attempt to project from virtual image to film may miss out some pixels on the film and draw others several times over).

Needless to say, these two projections should ideally be each other's inverses; in practice, a small error must be considered tolerable. It is possible to reduce the inversion error to the level of the computer's numerical precision by implementing only one of the projections as an explicit process, and turn the other into a problem of minimization or the solution of non-linear equations; but since the physical distortion remains and the number of image points to be computed is likely to be very large indeed, this is not worthwhile.

Another important aspect is the composite nature of the projections: disregarding the off-centre position of the projector, we can argue as follows:

If the lens were an ideal pin-hole (and the projection thus a perspective), the light rays through the circular film image would form a cone, passing through a circular cap of the dome and intersecting the virtual image plane in a circle. — So one part of the projection (in either direction) is the passage between the circle in the virtual image plane and the circular cap of the dome. With the above choice of parametrizations, this part can be performed explicitly using simple trigonometric functions.

The anamorphic effect of the lens is really a non-uniform widening of the cone of light rays; and when modelled as such, in terms of the trigonometric parameters, it is not likely to differ drastically from a scaling. Finally, the off-centre effects can be considered as small perturbations.

In the algorithms actually implemented, anamorphic and off-centre effects are treated together, but the very significant contributions from the plane-dome interactions are taken separately:

3.1.1 Line primitives: virtual image to film

The planar virtual image is assumed scaled to a size that fits with the dome-as-unit-sphere convention. Under a perspective with eye point at the origin, a point (X, Y) in the image then projects onto the dome point

$$(x, y, z) = (X, -Y, 1)/\mathrm{d}, \text{ where } \mathrm{d}^2 = X^2 + Y^2 + 1$$

and we can compute

$$u = \mathrm{atan}(x/z) \text{ and}$$
$$v = \mathrm{asin}(-y)$$

and finally

$$p = p(u, v)$$
$$q = q(u, v)$$

where the functions p, q are determined by the data fitting process

3.1.2 Raster primitives: film to virtual image

With assumptions and notation as above, we compute

$$u = u(p, q)$$
$$v = v(p, q)$$

again using functions determined by data fitting. Then

$$x = \sin u \cos v$$
$$y = -\sin v$$
$$z = \cos u \cos v$$

on the dome, and hence

$$X = x/z$$
$$Y = -y/z$$

3.2 The data fitting

The suggested data fitting process is very closely related to the line primitive projection of 3.1.1:

First, an approximation to the dome sphere is determined. If data were noisy, this would require, say, a non-linear least-squares fit. However, simple numerical experiments verified that the precision in the measurements is far higher than the degree of approximation to a mathematical sphere obtained in the actual dome, which is built from short, narrow aluminium plates. Hence, four suitable points were chosen among those measured, and an *interpolating* sphere computed:

The identities

$$(x_k - c_x)^2 + (y_k - c_y)^2 + (z_k - c_z)^2 = r^2, \ \mathrm{k} = 1, 2, 3, 4$$

form a system of non-linear equations for $\mathbf{c}$, r, easily solvable by the Newton method. The deviation of the remaining data points from this sphere is in the millimeter range.

Next, the data point values are transformed by translation, rotation and scaling (in that order) to give points on the unit sphere with all z-values positive. For these points, (u, v)-values can be computed as described above.

The (u, v)-values thus found are then translated to make a selected point near the centre of the image the new origin. Similarly, the measured (p, q)-values are balanced and scaled to fit inside $[-1;1]^2$. The two sets of mappings $p = p(u, v)$, $q = q(u, v)$ and $u = u(p, q)$, $v = v(p, q)$ can now be determined in any fashion deemed suitable. One choice is to use products of Chebyshev-polynomials as basis functions and determine the coefficient sets by a linear least squares fit. The only immediate disadvantage of doing so in this case is that data is scarce near the corners of the p,q-square, so errors (in particular errors in the reciprocity of the two mappings) will be much larger there than elsewhere. On the other hand, these areas are not projected.

Properties of the Chebyshev polynomials can be found in ref. [1], and the details of setting up the equations are given in appendix 1 in the form of a MatLab-script. The over-determined linear system is solved in the least squares sense using the MatLab division operator. Simple tests show the system to be well-conditioned, *i.e.* the use of normal equations, QR-factorization or singular value decomposition should lead to identical results.

Some simple graphical illustrations of the resulting mappings are given in appendix 2. The reader should be aware that the MatLab plots are not isotropic, so some additional (but non-essential) distortion is introduced in the figures presented. Practical use of the mappings would require that the coefficients be written to a file and read in by or textually copied to a special-purpose program. At the time of writing, no such program has been implemented, but as long as simple graphics formats are used for input and output images, this is not a particularly demanding task.

4 References

[1] Snyder, Martin Avery:
Chebyshev Methods in Numerical Approximation
Prentice-Hall 1966

Appendix 1: A MatLab program

The MatLab script presented below implements the data-fitting algorithm described in the text. The data files `kuppel3d.dat` and `billed.dat` contain the original measurements. Note, that this version does not store the coefficients, as the script was used directly in the interactive MatLab environment for various experiments.

```
%
%   File:    fit200.m
%
%   An attempt to construct the functions to
%   data-fit the measurements of spots on the
%   Planetarium dome and the corresponding
%   spots on the image placed under the mighty
%   200+ degree wide-angle projection lens.
%
%   The final outcome are the coefficients
%   needed to match (u,v)-values on an exact
%   sphere to (p,q)-values on the film strip.
%
%   Projection from the (x,y,z)-values on the
%   virtual (BIG, planar) image onto the exact
%   sphere is done "without errors":
%
%      (x,y,z) = (x,y,z)/sqrt(x.*x + y.*y + z.*z)
%
%      u = atan(x/z); v = asin(-y)
%
%   Then, p = P(u,v), q = Q(u,v), where P and Q
%   are 5.-degree polynomials.
%
%   A full report will appear elsewhere
%
%   This version: 15.12.96   (C) PSH, IMM
%
% ---------------------------------------------
%
```

```
%   Stage 1:
%
%   Reading the 3D data, discarding the last two
%   values, known to be references off the dome:
%
fid          = fopen('g:\mat\kuppel3d.dat');
[XYZ,count] = fscanf(fid, '%f %f %f\n', [3 inf]);
XYZ          = XYZ';
n            = (count/3)-2;
%
% ----------------------------------------------
%
%   Stage 2:
%
%   At this stage, a non-linear least squares
%   (or other norm) fitting should be performed.
%   However, the measurements are so accurate
%   that a little less work will do:
%
%   Four points were chosen by inspection, and
%   an interpolating sphere is now calculated:
%
x1 = XYZ(34,:)'; x2 = XYZ(25,:)';
x3 = XYZ(22,:)'; x4 = XYZ(3,:)';
%
%   Starting guesses for centre C and radius r:
%
m = (x1+x3)/2; d = m - x2;
C = x2 + 2*d;  r = 3*sqrt(d'*d);
%
%   Newton iteration, using functions for f and J:
%
w = [C;r];
for i = 1:6
   w = w - plaj(x1,x2,x3,x4,C,r)\plaf(x1,x2,x3,x4,C,r);
   C = w(1:3); r = w(4);
end
%
```

```
%   Bringing the dome points down to the unit sphere:
%
x = (XYZ(1:n,1)-C(1))/r;
y = (XYZ(1:n,2)-C(2))/r;
z = (XYZ(1:n,3)-C(3))/r;
%
%   Rotating points to obtain all-positive z-values:
%
%   (Further rotations are done implicitly in stage 3,
%    by manipulation of (u,v)-values)
%
phi = -3*pi/4;
Y = cos(phi)*y - sin(phi)*z;
Z = sin(phi)*y + cos(phi)*z;
%
%   Computing the (u,v) values, taking into account
%   the wonderful upside-down coordinate system:
%
u = atan(x./Z); v = asin(-Y);
%
% -----------------------------------------------
%
%   Stage 3:
%
%   Reading the measurements from the physical
%   image, taking into account that there is a
%   zero'th image point (which is discarded):
%
fid         = fopen('g:\mat\billed.dat');
[XY,count] = fscanf(fid, '%f %f\n', [2,50]);
XY = XY';
Xb = XY(2:count/2,1);
Yb = XY(2:count/2,2);
%
%   Balancing and scaling the two sets of variables:
%
% dd = 85
mx = Xb(25); my = Yb(25);
```

```
X  = Xb-mx;   Y = Yb-my;
dx = dd/2;    dy = dd/2;
X  = X/dx;    Y = Y/dy;

mu = u(25);  mv = v(25);
U  = u-mu;    V = v-mv;
du = pi/2;   dv = pi/2;
U  = U/du;    V = V/dv;
%
%   Forming the observation equations, using
%   tensor products of Chebyshev polynomials:
%
F = zeros(n,21);
%
%   Constant term:
%
F(:,1)   = F(:,1)+1;
%
%   Degree 1:
%
F(:,2)   = U; F(:,3) = V;
%
%   Degree 2:
%
F(:,4)   = 2*U.*U - 1;
F(:,5)   = U.*V;
F(:,6)   = 2*V.*V -1;
%
%   Degree 3:
%
F(:,7)   = 2*U.*F(:,4) - U; F(:,8)   = V.*F(:,4);
F(:,10)  = 2*V.*F(:,6) - V; F(:,9)   = U.*F(:,6);
%
%   Degree 4:
%
F(:,11) = 2*U.*F(:,7)   - F(:,4);
F(:,12) = V.*F(:,7);
F(:,13) = F(:,4).*F(:,6);
```

```
F(:,14) = U.*F(:,10);
F(:,15) = 2*V.*F(:,10) - F(:,6);
%
%   Degree 5:
%
F(:,16) = 2*U.*F(:,11)  - F(:,7);
F(:,17) = V.*F(:,11);
F(:,18) = F(:,6).*F(:,7);
F(:,19) = F(:,4).*F(:,10);
F(:,20) = U.*F(:,15);
F(:,21) = 2*V.*F(:,15) - F(:,10);
%
cx = F\X; cy = F\Y;
%
%   Since, for ray-casting purposes, we need
%   the opposite transformation, we go through
%   the motions once more:
%
FF = zeros(n,21);
%
%   Constant term:
%
FF(:,1)  = FF(:,1)+1;
%
%   Degree 1:
%
FF(:,2)  = X; FF(:,3) = Y;
%
%   Degree 2:
%
FF(:,4)  = 2*X.*X - 1;
FF(:,5)  = X.*Y;
FF(:,6)  = 2*Y.*Y -1;
%
%   Degree 3:
%
FF(:,7)  = 2*X.*FF(:,4) - X; FF(:,8)  = Y.*FF(:,4);
FF(:,10) = 2*Y.*FF(:,6) - Y; FF(:,9)  = X.*FF(:,6);
```

```
%
%   Degree 4:
%
FF(:,11) = 2*X.*FF(:,7)  - FF(:,4);
FF(:,12) = Y.*FF(:,7);
FF(:,13) = FF(:,4).*FF(:,6);
FF(:,14) = X.*FF(:,10);
FF(:,15) = 2*Y.*FF(:,10) - FF(:,6);
%
%   Degree 5:
%
FF(:,16) = 2*X.*FF(:,11)  - FF(:,7);
FF(:,17) = Y.*FF(:,11);
FF(:,18) = FF(:,6).*FF(:,7);
FF(:,19) = FF(:,4).*FF(:,10);
FF(:,20) = X.*FF(:,15);
FF(:,21) = 2*Y.*FF(:,15) - FF(:,10);
%
cu = FF\U; cv = FF\V;
```

Appendix 2: MatLab graphics

Figure 1 shows the original points on the slide (marked "+") overlaid with the resulting points in the (u,v)-coordinate space (marked "o"), both sets balanced and scaled. It is thus a comparison of apples and pears, but it illustrates the nature of the non-perspective part of the transformation.

Figure 2 shows the effect of the transformation $u = u(p,q)$, $v = v(p,q)$ as applied to a rectangular grid in the (p,q)-space. The resulting grid is displayed in the manner of figure 1. The image clearly shows the scaling and "rounding" caused by the anamorphism, as well as the tilt and assymetry caused by the off-centre position of the projector.

Figure 3 shows the effect of the transformation $p = p(u,v)$, $q = q(u,v)$ as applied to a rectangular grid in the (u,v)-space. If printed as a slide in the correct size (without MatLabs additional anisotropy) and projected onto the dome, a picture like this will thus appear as a set of latitudes and longitudes on the dome. Again, the effects of scaling, "rounding" and assymmetry are visible, the latter being the dominant.

Figure 4 shows a simple consistency test: a rectangular grid in (p,q) was mapped to (u,v) and the resulting image mapped back by the two transformations. The function displayed is the error in q. Clearly, almost all significant error is outside the area actually projected — in fact, a closer study revealed that within a 60mm by 60mm square contained in the parameter area, the highest *combined* error was less than 2%. This, of course, says nothing about the visual effects of these transformations, but it confirms that the method treats the actual information in the original data in a consistent manner.

Appendix 3: Some fotos (courtesy Sv. B. Sørensen)

(Images follow figures 1-4, see captions)

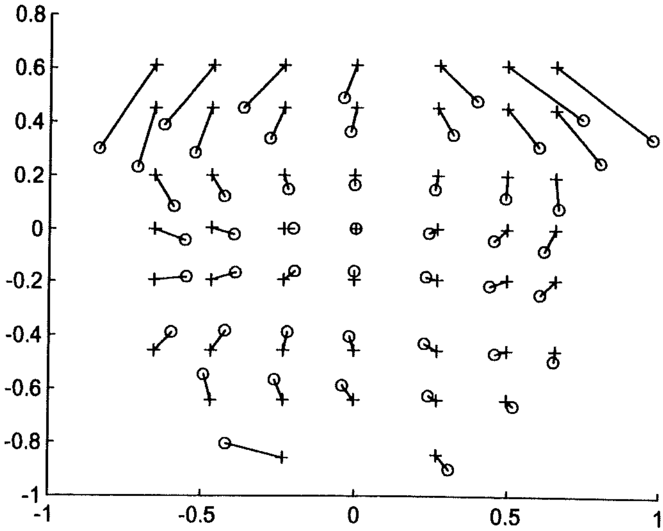

Figure 1: From image to parameters

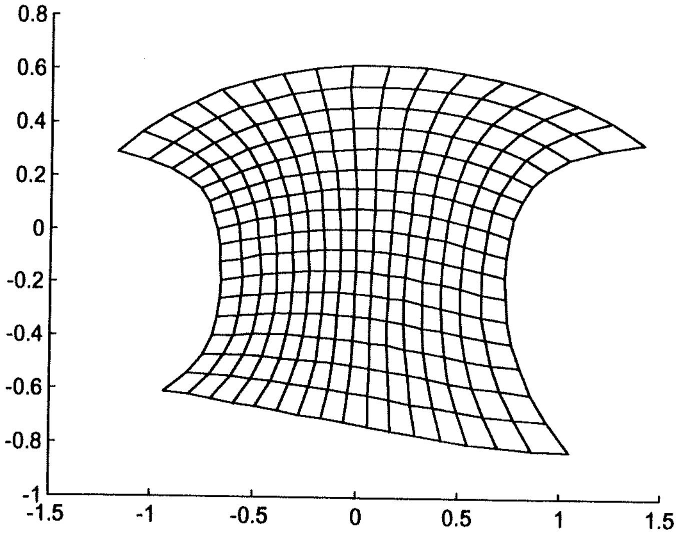

Figure 2: From image to parameters, a grid

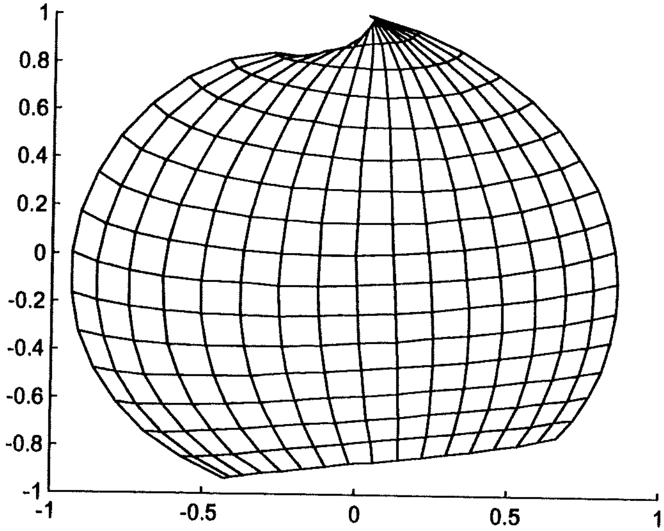

Figure 3: A pre-distorted latitude-longitude net

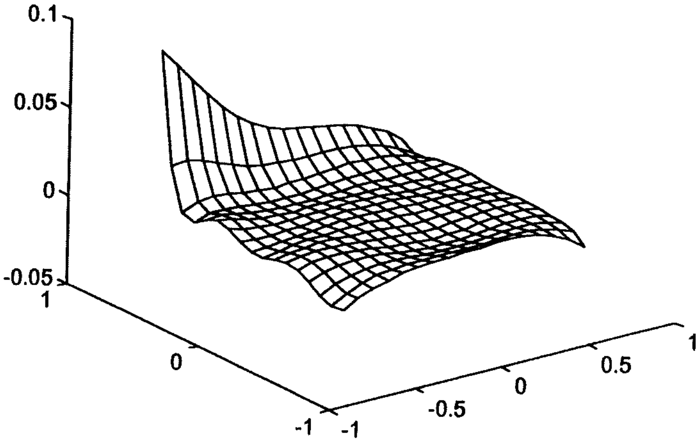

Figure 4: Consistency of the map and its inverse

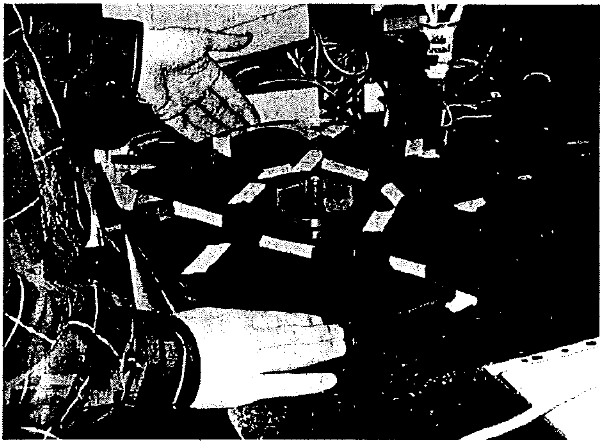

Figure 5: Mouting the test image

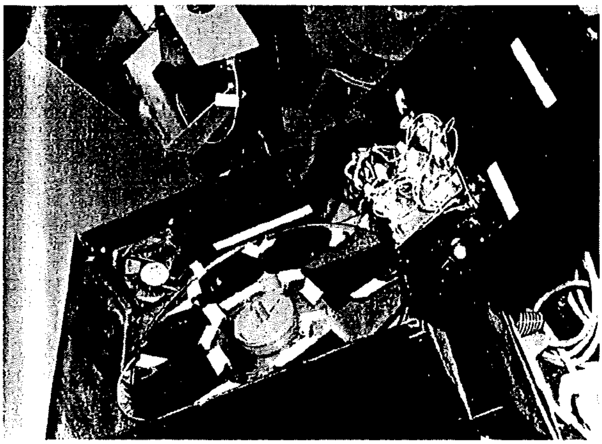

Figure 6: The projector, open, with the mounted image

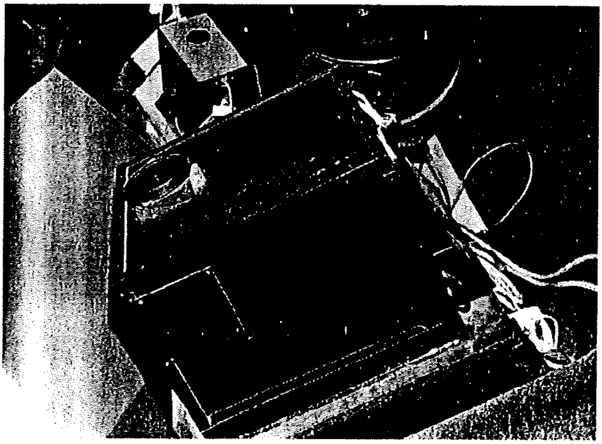

Figure 7: The projector, closed

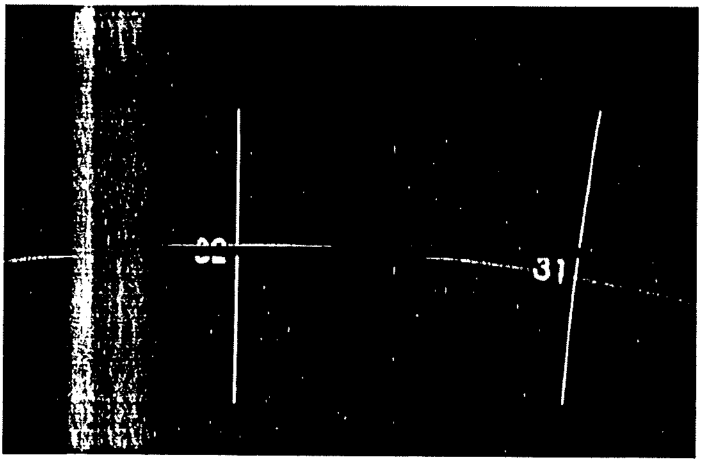

Figure 8: A small section of the projected test image as seen in the dome