

Recovering Arches in Facade using Ray - Plane intersections in 3-D

G. D. Borshukov and P. Debevec

Department of Electrical Engineering and Computer Science,
University of California, Berkeley, CA 94720

1 Assumptions

1. Focal length f and image plane center (u_0, v_0) in pixels are known.
2. Camera locations $\mathbf{R}^C, \mathbf{T}^C$ in the world coordinate system are previously reconstructed by the minimization algorithm.
3. The arch is initially created as a box which parent and relation are specified. Then, its width, depth, and height are reconstructed by the minimization algorithm.
4. We know image points like (x_i, y_i) that lie on the arch contour in the image plane.

2 Derivation

First the image measurement (x_i, y_i) is converted into camera coordinates $\mathbf{p}_i = [x \ y \ -1]^T$ where

$$\begin{aligned} x &= (x_i - u_0) \frac{1}{f} \\ y &= (y_i - v_0) \frac{1}{f} \end{aligned} \quad (1)$$

Now $\mu \mathbf{p}_i$ is a ray from the camera's COP passing through (x_i, y_i) . This ray intersects the face of the arch box where the arch begins at point $\mu_0 \mathbf{p}_i$.

To find this intersection, i.e. the value of μ_0 , we use a point \mathbf{P}_c on the face (the middle of the bottom edge) with world coordinates \mathbf{p}_c^W and camera coordinates $\mathbf{p}_c^C = \mathbf{R}^C(\mathbf{p}_c^W - \mathbf{T}^C)$, and the face normal \mathbf{n} . The point $\mu_0 \mathbf{p}_i$ lies on the face, therefore, its distance from the face:

$$[\mu_0 \mathbf{p}_i - \mathbf{p}_c^C]^T (\mathbf{R}^C \mathbf{n}) = 0 \quad (2)$$

which gives:

$$\mu_0 = \frac{[\mathbf{p}_c^C]^T (\mathbf{R}^C \mathbf{n})}{\mathbf{p}_i^T (\mathbf{R}^C \mathbf{n})} \quad (3)$$

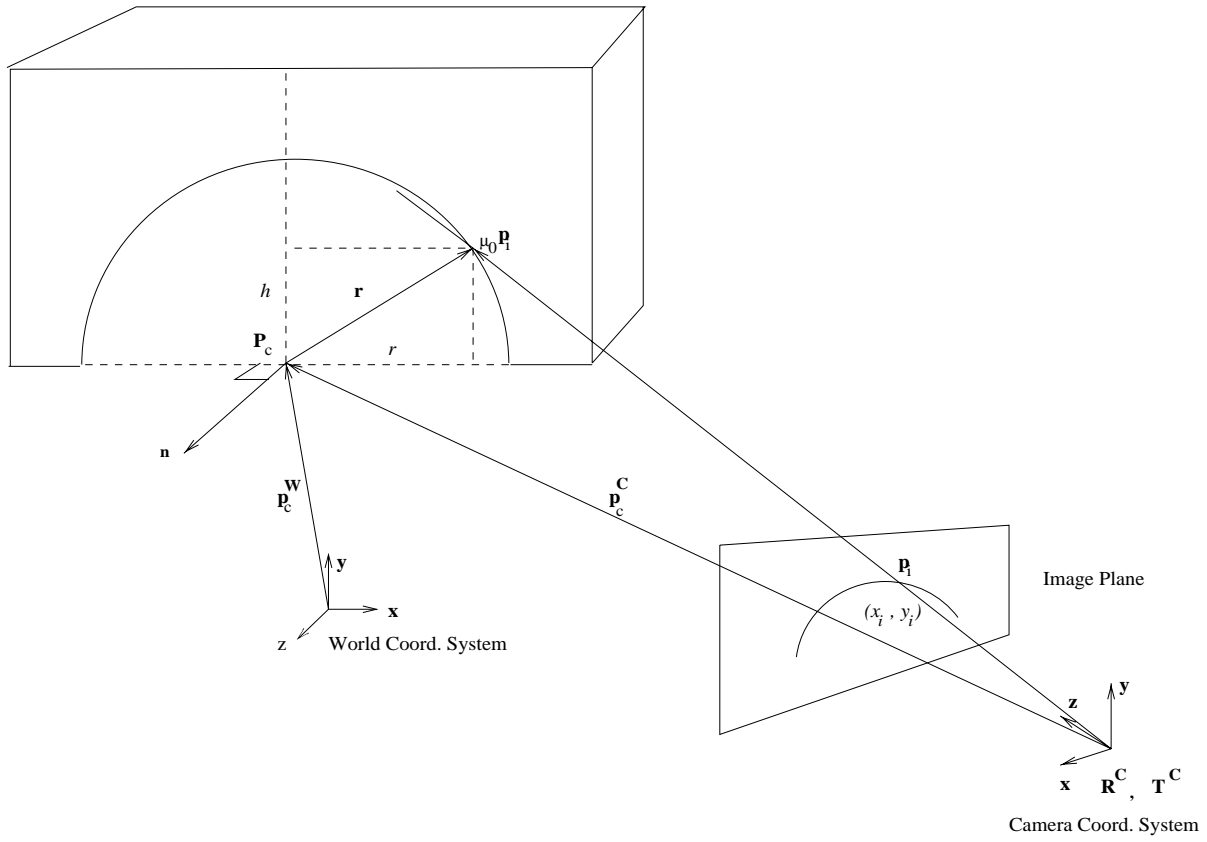


Figure 1: Geometry and notation

We need to rotate the vector $(\mu_0 \mathbf{p}_i - \mathbf{p}_c^C)$ back into world coordinates to obtain the desired vector \mathbf{r} :

$$\mathbf{r} = [\mathbf{R}^C]^{-1}(\mu_0 \mathbf{p}_i - \mathbf{p}_c^C) \quad (4)$$

The algorithm uses the projections r and h of this vector onto the bottom edge and the middle axis of the face to automatically generate the arch surface.

1.2.3 Results

Fig. 1.3 shows the results of reconstructing a 3-D model of the Arc de Triomphe using the new arch recovery tools.

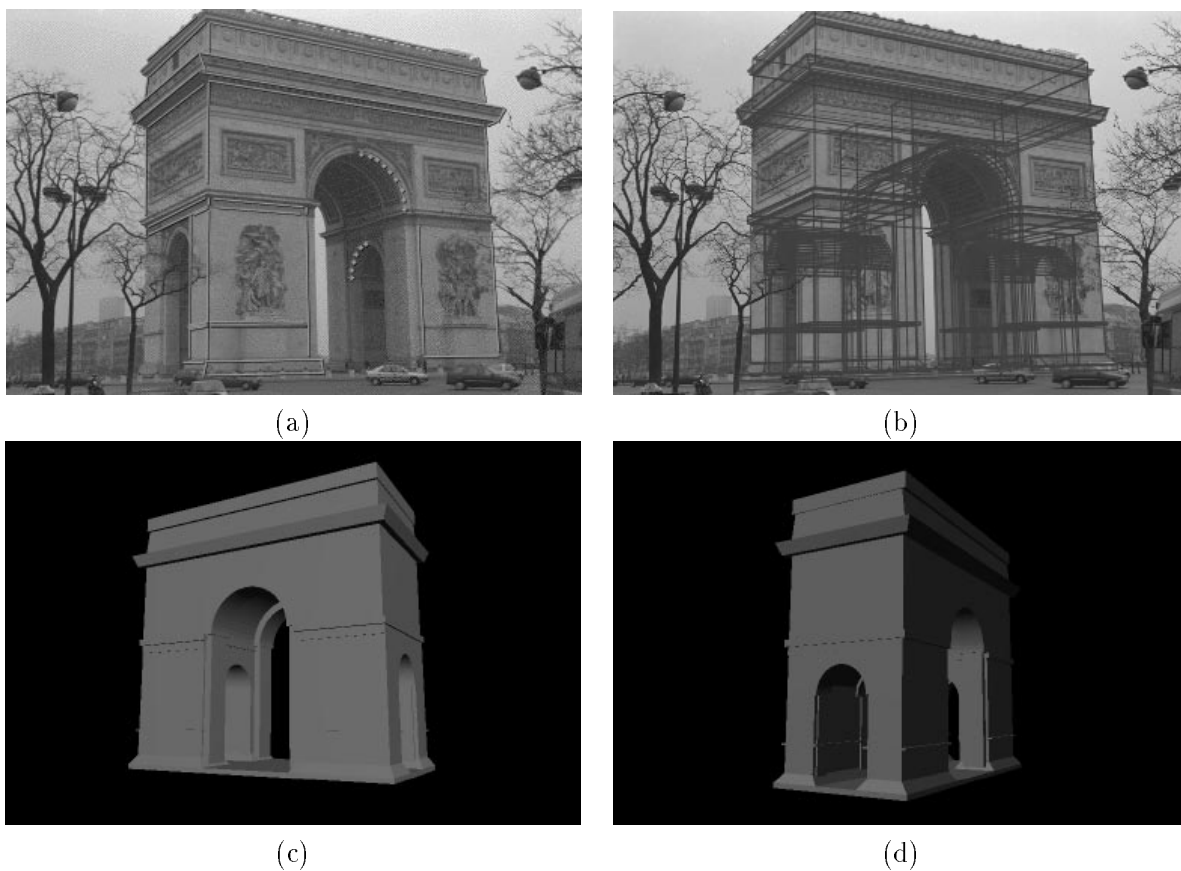


Figure 1.3: Model of the Arc de Triomphe demonstrating the new arch recovery capabilities of Façade. (a) One of three photographs used to reconstruct the Arc de Triomphe, with marked features indicated. (b) Reconstructed model edges projected into the original photograph. (c) Recovered model of the Arc de Triomphe. (d) Another view of the recovered 3-D model.

Recovering the Radius and Offset of a Cross-Section in SORs using Minimum Distance between Two Rays in 3-D

G. D. Borshukov and P. Debevec

Department of Electrical Engineering and Computer Science,
University of California, Berkeley, CA 94720

1 Assumptions

1. Focal length f and image plane center (u_0, v_0) in pixels are known.
2. Camera locations $\mathbf{R}^C, \mathbf{T}^C$ in the world coordinate system are previously reconstructed by the minimization algorithm. The camera coordinate system is aligned with the world coordinate system. From now on all vectors will be in world coordinates.
3. The SOR central axis is known, i.e. we know a point on the axis \mathbf{p}_b (usually the base point) and the axis direction \mathbf{m} (usually $[0 \ 1 \ 0]^T$).
4. We know image points like (x_i, y_i) that lie on the occluding contour in the image plane.

2 Derivation

First the image measurement (x_i, y_i) is converted into camera coordinates $\mathbf{p}_i = [x \ y \ -1]^T$ where

$$\begin{aligned} x &= (x_i - u_0) \frac{1}{f} \\ y &= (y_i - v_0) \frac{1}{f} \end{aligned} \quad (1)$$

We want find the minimum distance between the rays $\mathbf{p}_b + \lambda \mathbf{m}$ and $\mu \mathbf{p}_i$. Exploiting the fact that the minimum distance vector

$$\mathbf{d}_0 = (\mu_0 \mathbf{p}_i - \mathbf{p}_b - \lambda_0 \mathbf{m}) \quad (2)$$

must be perpendicular to each ray, conveniently, our task boils down to solving the following simultaneous equations with respect to λ_0 and μ_0 .

$$\mu_0 \mathbf{p}_i^T \mathbf{d}_0 = 0 \quad (3)$$

$$\lambda_0 \mathbf{m}^T \mathbf{d}_0 = 0 \quad (4)$$

Excluding the trivial solutions $\lambda_0 = 0$ and $\mu_0 = 0$ and substituting (1) into (2) and (3) we get

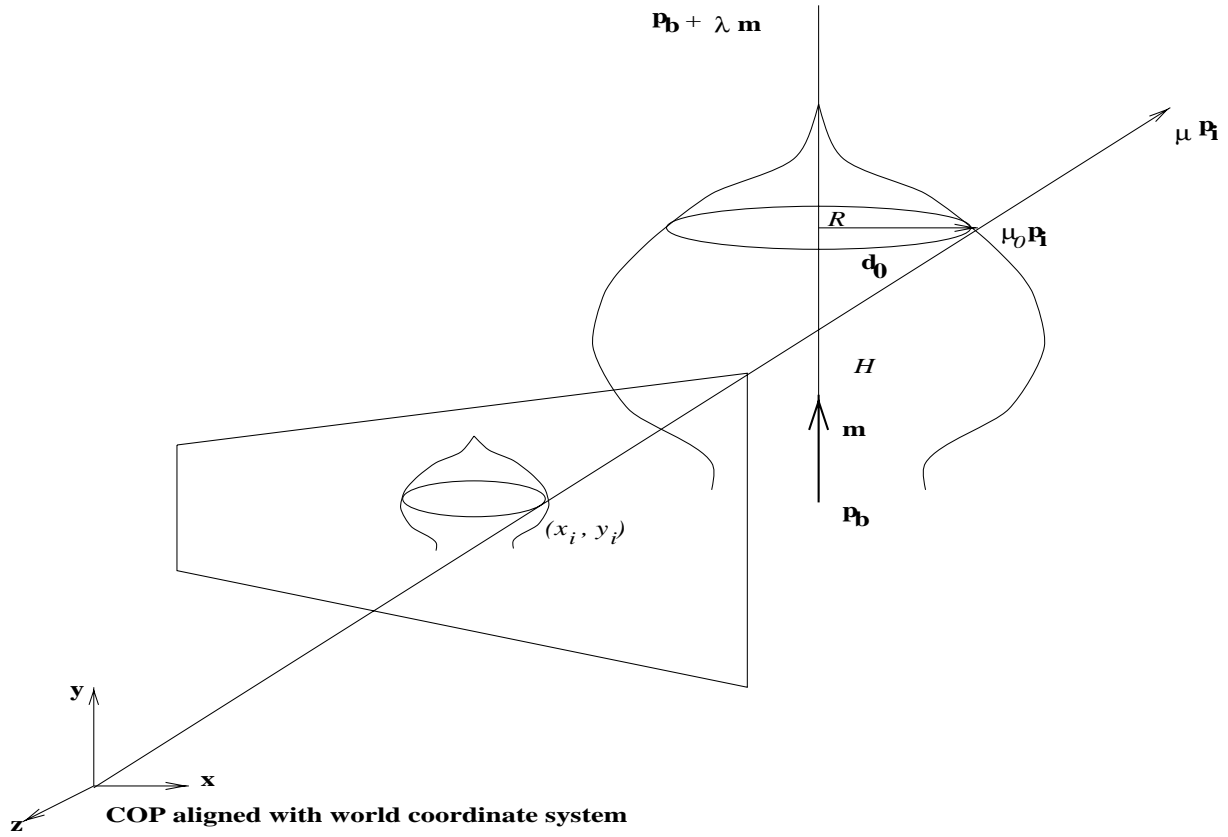


Figure 1: Geometry and notation

$$A\mu_0 - C\lambda_0 = B \quad (5)$$

$$C\mu_0 - E\lambda_0 = D \quad (6)$$

where

$$A = \mathbf{p}_i^T \mathbf{p}_i \quad (7)$$

$$B = \mathbf{p}_i^T \mathbf{p}_b \quad (8)$$

$$C = \mathbf{p}_i^T \mathbf{m} \quad (9)$$

$$D = \mathbf{m}^T \mathbf{p}_b \quad (10)$$

$$E = \mathbf{m}^T \mathbf{m} \quad (11)$$

Further using (4) we obtain

$$\mu_0 = \frac{B + C\lambda_0}{A} \quad (12)$$

which could be substituted in (5) to yield

$$\lambda_0 = \frac{BC - AD}{AE - C^2} \quad (13)$$

Now knowing λ_0 and μ_0 , the radius R of a circular cross section offset by $H = \lambda_0 \mathbf{m}$ from \mathbf{p}_b (usually on the base plane) can be expressed by

$$R = \sqrt{\mathbf{d}_0^T \mathbf{d}_0} \quad (14)$$

The algorithm uses the quantities H and R to automatically generate the surface of revolution.

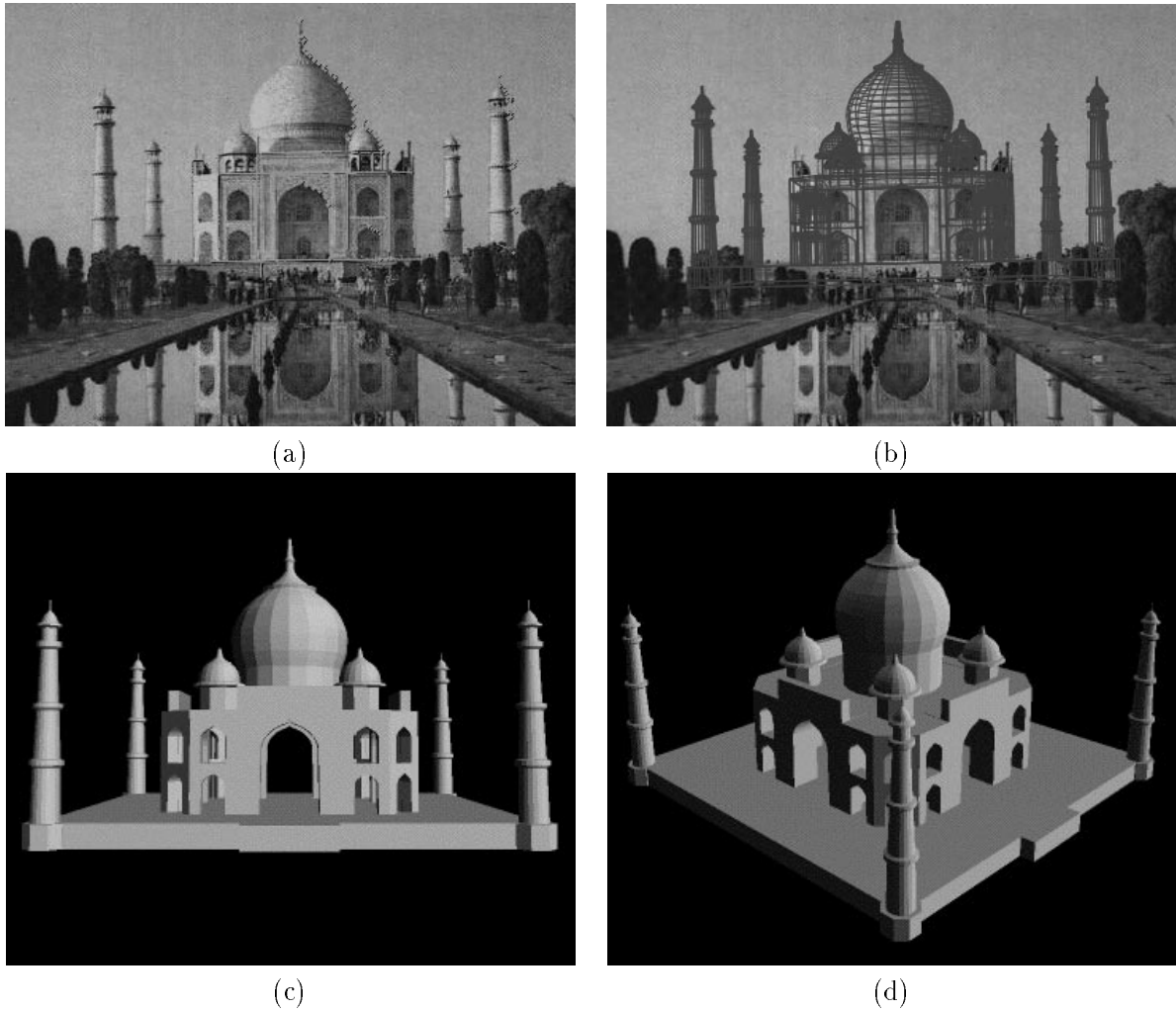


Figure 1.5: Model of the majestic Taj Mahal created with the new surface of revolution and arch reconstruction tools. **(a)** A single low-resolution photograph of the Taj Mahal obtained from the Internet, with marked features shown. **(b)** Reconstructed model edges projected onto the original photograph. **(c)** 3-D model of the Taj Mahal, complete with domes and minarets, recovered from the single photograph in less than an hour of modeling time. **(d)** Another view of the recovered 3-D model.