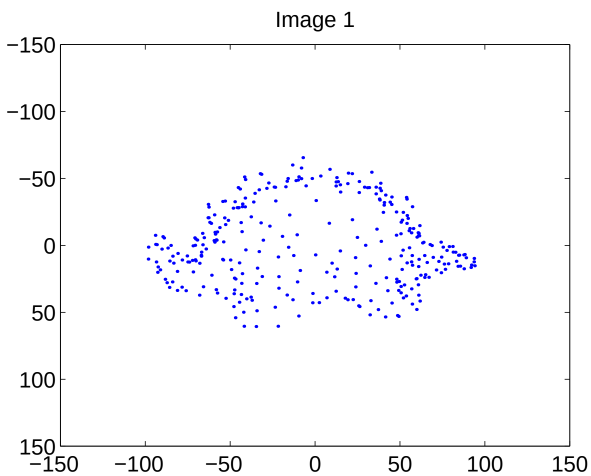
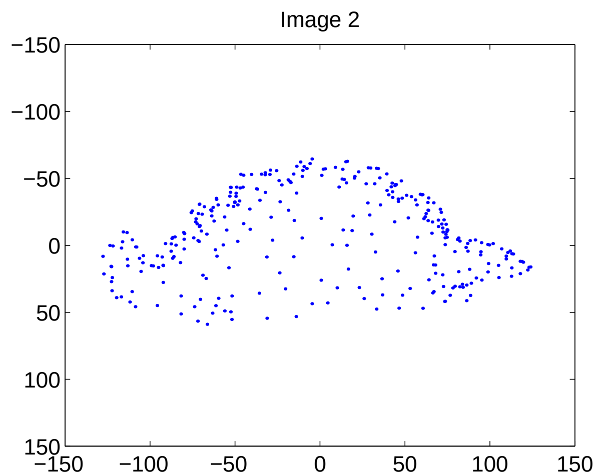
Multi-View Factorization Techniques

Suppose $\{\vec{x}_{j,n}\}_{j=1,n=1}^{J,N}$ is a set of *corresponding* image coordinates for N scene points in J images. That is, $\vec{x}_{j,n}$ denotes the location in the j^{th} image for the n^{th} 3D scene point, $\vec{X}_n$.





Such corresponding points may be obtained from local feature points, for example.

Problem: Estimate the 3D point positions, $\{\vec{X}_n\}_{n=1}^{N}$, along with the placement and calibration parameters for the J cameras.

Tutorials: tutorials/3dRecon/orthographic/orthoMassageDino.m, and 3dRecon/projective/projectiveMassageDino.m (both in utvis)

Perspective Projection

The image points, $\vec{p}_{j,n}$, and the 3D scene points, $\vec{P}_n$, are related by perspective projection,

$$\vec{p}_{j,n} = \frac{1}{z_{j,n}} M_j \vec{P}_n \ . \tag{1}$$

- $\vec{p}_{j,n} = (x_{j,n}, y_{j,n}, 1)^T$ is given in homogeneous pixel coordinates;
- $\vec{P}_n = (P_{n,1}, P_{n,2}, P_{n,3}, 1)^T$ is also in homogeneous coordinates;
- $M_j = M_{in,j}\, M_{ex,j}$ is the 3×4 camera matrix formed from the product of the intrinsic and extrinsic calibration matrices;
- $z_{j,n}$ is the projective depth, $z_{j,n} = \vec{e}_3^{\,T} M_j \vec{P}_n$, where $\vec{e}_3^{\,T} = (0, 0, 1)$ (i.e., $\vec{e}_3$ is the third standard unit vector).

For convenience we assume the intrinsic matrices have the form

$$M_{in,j} = \begin{pmatrix} f_j & 0 & 0 \\ 0 & f_j & 0 \\ 0 & 0 & 1 \end{pmatrix} . \tag{2}$$

The extrinsic calibration matrices are in general given by

$$M_{ex,j} = \left(R_j, \ \ - R_j \, \vec{d}_j \right) \ , \tag{3}$$

where R_j is the rotation from the world to the j^{th}-camera's coordinates, and $\vec{d}_j$ is the position, in world coordinates, of the nodal point for the j^{th} camera.

Bundle Adjustment

We wish to solve for the point positions $\mathcal{P} \equiv \{\vec{P}_n\}_{n=1}^N$ and the camera matrices $\mathcal{M} \equiv \{M_j\}_{j=1}^J$ by minimizing

$$\mathcal{O}(\mathcal{M}, \mathcal{P}) \equiv \sum_{j,n} \left|\left| \begin{pmatrix} x_{j,n} \\ y_{j,n} \end{pmatrix} - \frac{1}{\vec{e}_3^{\,T} M_j \vec{P}_n} \left(I_2,\ \vec{0}\right) M_j \vec{P}_n \right|\right|^2 , \quad (4)$$

where the camera matrices M_j must be of the form $M_{in,j} M_{ex,j}$, as above in (2) and (3).

This nonlinear LS optimization problem is called **bundle adjustment**. For Gaussian IID measurement noise it is a ML estimator.

In these notes we discuss two approximations to bundle adjustment:

1. Approximate perspective projection by scaled orthographic projection.
2. Rescale each term in the bundle adjustment objective function (4) and solve a bilinear problem.

Scaled-Orthographic Projection

Scaled-orthographic projection provides an approximation of perspective projection (1) for the case of narrow fields of view,

$$\max\{|x_{j,n}|, |y_{j,n}|\} \ll f_j \ ,$$

and relatively shallow depth variations; i.e.,

$$z_{j,n} \approx 1/s \ ,$$

for some constant scale factor s.

For scaled-orthographic projection, the image points and the scene points are related by

$$\left(I_2,\ \vec{0}\right)\vec{p}_{j,n} \ = \ s\left(I_2,\ \vec{0}\right)M_j\vec{P}_n. \tag{5}$$

Here $\vec{p}_{j,n}$, $\vec{P}_n$ and M_j are as above. This is **bilinear** in the scaled camera matrix, sM_j, and the 3D point, $\vec{P}_n$.

Differences from Mean Image Points

Let $\bar{\vec{p}}_j = \frac{1}{N}\sum_{n=1}^{N} \vec{p}_{j,n}$ be the average image position. Similarly, let the average 3D scene position be $\bar{\vec{P}} = \frac{1}{N}\sum_{n=1}^{N} \vec{P}_n$. Then, using (5), we can show

$$\vec{d}_{j,n} \;=\; \tilde{M}_j\, \vec{D}_n, \tag{6}$$

where

$$\begin{aligned}
\vec{d}_{j,n} &= \left(I_2,\ \vec{0}\right)\,(\vec{p}_{j,n} - \bar{\vec{p}}_j)\ , \\
\vec{D}_n &= \left(I_3,\ \vec{0}\right)\,(\vec{P}_n - \bar{\vec{P}})\ , \\
\tilde{M}_j &= s\left(I_2,\ \vec{0}\right) M_j \left(I_3,\ \vec{0}\right)^T\ .
\end{aligned}$$

Moreover, from the form of the internal and external camera calibration matrices, (2) and (3), it follows that the scaled-orthographic projection matrix $\tilde{M}_j$ has the form

$$\tilde{M}_j \;=\; s\begin{pmatrix} f_j & 0 & 0 \\ 0 & f_j & 0 \end{pmatrix} R_j \;=\; s f_j \left(I_2,\ \vec{0}\right) R_j, \tag{7}$$

where R_j is the rotation matrix for the j^{th} camera, as above.

Derivation: Difference from Mean

Let $\vec{\bar{p}}_j = \frac{1}{N}\sum_{n=1}^{N} \vec{p}_{j,n}$ be the average image point in the j^{th} image, and $\vec{\bar{P}} = \frac{1}{N}\sum_{n=1}^{N} \vec{P}_n$ be the average scene point.

Then, by equation (5), we have

$$\left(I_2,\ \vec{0}\right)\vec{\bar{p}}_j = s\left(I_2,\ \vec{0}\right)M_j\vec{\bar{P}}$$

Subtracting this from (5) we find

$$\left(I_2,\ \vec{0}\right)(\vec{p}_{j,n} - \vec{\bar{p}}_j) = s\left(I_2,\ \vec{0}\right)M_j(\vec{P}_n - \vec{\bar{P}})\ .$$

Note the 4^{th} component of $\vec{P}_n - \vec{\bar{P}}$ is equal to $1-1=0$. Therefore we can drop this 4^{th} component, and obtain

$$\vec{d}_{j,n} = \tilde{M}_j\vec{D}_n,$$

where

$$\begin{aligned}
\vec{d}_{j,n} &= \left(I_2,\ \vec{0}\right)(\vec{p}_{j,n} - \vec{\bar{p}}_j)\ , \\
\vec{D}_n &= (I_3,\ \vec{0})(\vec{P}_n - \vec{\bar{P}})\ , \\
\tilde{M}_j &= s\left(I_2,\ \vec{0}\right)M_j\left(I_3,\ \vec{0}\right)^T\ .
\end{aligned}$$

Which is what we set out to show.

Notice we can use the definitions of $M_{in,j}$ and $M_{ex,j}$ to simplify $\tilde{M}_j$ above. We find

$$\begin{aligned}
\tilde{M}_j &= s\left(I_2,\ \vec{0}\right)M_j\left(I_3,\ \vec{0}\right)^T, && (8)\\
&= s\left(I_2,\ \vec{0}\right)M_{in,j}M_{ex,j}\left(I_3,\ \vec{0}\right)^T, && (9)\\
&= s\begin{pmatrix} f_j & 0 & 0 \\ 0 & f_j & 0 \end{pmatrix}R_j && (10)
\end{aligned}$$

This gives equation (7) above.

Scaled-Orthographic Factorization

As defined above, $\vec{d}_{j,n} = \vec{x}_{j,n} - \bar{\vec{x}}_j$ are the centred 2D observations ($\vec{x}_{j,n}$ is the observed position of the j^{th} point in the n^{th} image, and $\bar{\vec{x}}_j$ is the average of these over all n).

Let $C = \left(\vec{d}_{j,n}\right)$ be the $2J \times N$ data matrix, where the j^{th} column is formed by stacking the $\vec{d}_{j,n}$ for all N frames:

$$C \; = \; \begin{pmatrix} \vec{d}_{1,1} & \vec{d}_{1,2} & \cdots & \vec{d}_{1,N} \\ \vec{d}_{2,1} & \vec{d}_{2,2} & \cdots & \vec{d}_{2,N} \\ \vdots & \vdots & \ddots & \vdots \\ \vec{d}_{J,1} & \vec{d}_{J,2} & \cdots & \vec{d}_{J,N} \end{pmatrix} \tag{11}$$

From equation (6), i.e., $\vec{d}_{j,n} = \tilde{M}_j \, \vec{D}_n$, we then have

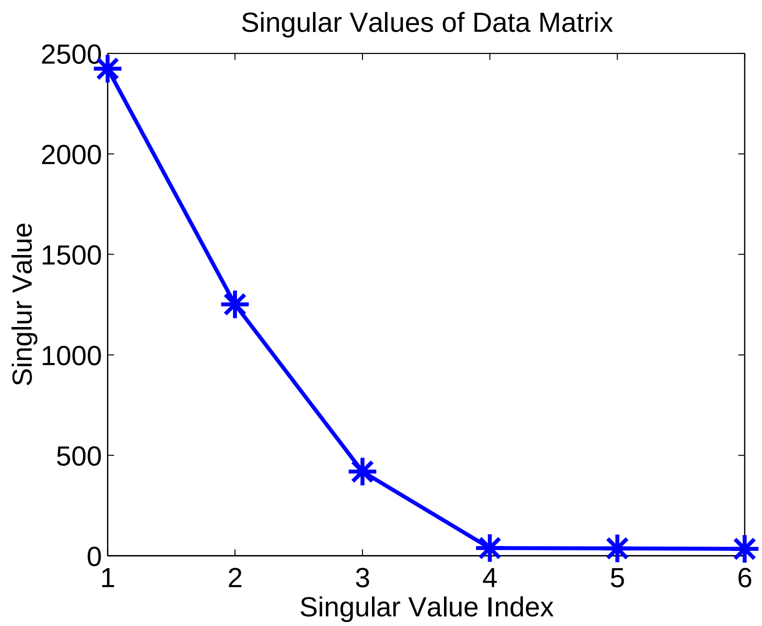
$$C \; = \; \begin{pmatrix} \tilde{M}_1 \\ \tilde{M}_2 \\ \vdots \\ \tilde{M}_J \end{pmatrix} \left(\vec{D}_1 \quad \vec{D}_2 \quad \cdots \quad \vec{D}_N \right) \; = \; MD \, . \tag{12}$$

That is, M is the $2J\times 3$ matrix formed by stacking the $\tilde{M}_j$ matrices, and D is the $3\times N$ shape matrix with columns $\vec{D}_n$.

This equation implies that **the data matrix has at most rank 3** (in the ideal case without considering noise).

Factorization via SVD

Performing an SVD on the data matrix C, for a case with $J = 3$ images, provides $C = W\Sigma V^T$ with the singular values shown below:



See the 3dRecon Matlab tutorial `orthoMassageDino.m` ($\sigma_n = 1$ pixel).

Affine Shape

What does the factorization $C = W\Sigma V^T$ tell us about the shape of the objects being imaged? For notational convenience, we assume that all but the first 3 singular values of Σ have been set to zero or, equivalently, Σ is 3×3, W is $2J \times 3$ and V^T is $3 \times N$.

We now have two rank 3 factorizations of C, namely MD and $W\Sigma V^T$. And there are multiple ways that we could specify M and D in terms of W, Σ and V. For example, we could let $M = W$ and $D = \Sigma V^T$.

In fact it is easy to see that the facotization is only unique up to a nonsingular 3×3 matrix A; for some nonsingular A we have

$$D = A^{-1}\Sigma V^T \tag{13}$$

$$M = WA \tag{14}$$

We therefore know the shape matrix, D, up to the 9 parameters in A, namely $AD = \Sigma V^T$. This is known as an **affine reconstruction** of the shape D.

Affine Shape (Cont.)

What can A do to a shape?

For example, consider a configuration of 3D points as specified by the $3 \times N$ matrix D, and suppose we have a nonsingular matrix A. What does the configuration of points AD look like?

Use SVD to decompose A into $U_a \Sigma_a V_a^T$. So $AD = U_a(\Sigma_a(V_a^T D))$ is obtained by rotating/reflecting D using V_a^T, then stretching/shrinking the result along the axes according to Σ_a, and finally rotating/reflecting this result using U_a. Affine shape preserves parallel lines and intersecting lines, but not angles and lengths.

The equivalence class of all configurations that can be obtained with transformations of this form is called **affine shape**.

See Tomasi and Kanade (1992) for the original factorization method.

Euclidean Reconstructions

We can determine many of the parameters in A from knowledge about the cameras.

In particular, as above, suppose the projection matrix $\tilde{M}_j$ satisfies

$$\tilde{M}_j \;=\; sf_j \left(I_2,\; \vec{0}\right) R_j \;,$$

for some value of sf_j. Then, based on $M = WA$ in (13), let W_j be the j^{th} 2×3 block in W. That is, W_j occupies the same rows of W as $\tilde{M}_j$ does in M, so $\tilde{M}_j = W_j A$. And since $R_j R_j^T = I_3$ we have

$$\tilde{M}_j\,\tilde{M}_j^T \;=\; s^2 f_j^2\, I_2 \;=\; W_j\, A\, A^T\, W_j^T \;. \tag{15}$$

Here, the only unknowns are the scale factor for the j^{th} image, sf_j, and the 3×3 symmetric positive definite matrix $Q = AA^T$.

For each j, (15) provides 2 linear homogeneous equations for the coefficients of Q. For $J \geq 3$ we have $2J \geq 6$ homogeneous linear equations which we can solve for Q, up to a scalar multiple r_q^2.

Then, since Q is symmetric positive definite, it can be factored as $Q = U_q \Lambda_q U_q^T$. The columns of U_q are the eigenvectors of Q, and the diagonal Λ_q are its eigenvalues. Assuming non-negative eigenvalues, we can write $A = \frac{1}{r_q} U_q \Lambda_q^{1/2} R_q^T$. Here, r_q represents the unknown scale factor in A, and R_q is an arbitrary orthogonal 3×3 matrix.

Euclidean Reconstruction (Cont.)

Therefore we have recovered $A = \frac{1}{r_q} K_q R_q^T$ where $K_q = U_q \Lambda_q^{1/2}$ is known. As a consequence we have recovered the shape matrix D_r and the camera matrix M_r as

$$D \;=\; r_q\, R_q\, D_r\,, \quad \text{for } \; D_r = K_q^{-1}\, \Sigma\, V^T\,, \tag{16}$$

$$M \;=\; \frac{1}{r_q} M_r\, R_q^T\,, \quad \text{for } \; M_r = W\, K_q\,. \tag{17}$$

This is called a **Euclidean reconstruction**. We have recovered the shape up to a 3D scale r_q, and a rotation/reflection R_q. Equivalently, this is referred to as **metric shape** recovery.

Ambiguities:

- The ambiguity of the rotation R_q reflects the fact that we cannot recover the true orientation of the world coordinate frame. The unknown rotation R_q affects the shape, via $D = R_q D_r$, and the camera matrices, via $M = M_r R_q^T$. That is, R_q rotates both the scene and the cameras together.
- The ambiguity in the scale r_q reflects the fact that we do not know the scale of the world coordinate frame. We could be imaging a tiny scene with large scale factors sf_j, and we could not tell from the images alone. (Think about making the movie Titanic.) Here r_q scales the shape via $D = r_q D_r$, and also the scale parameters f_j in the cameras, via $M = \frac{1}{r_q} M_r$.

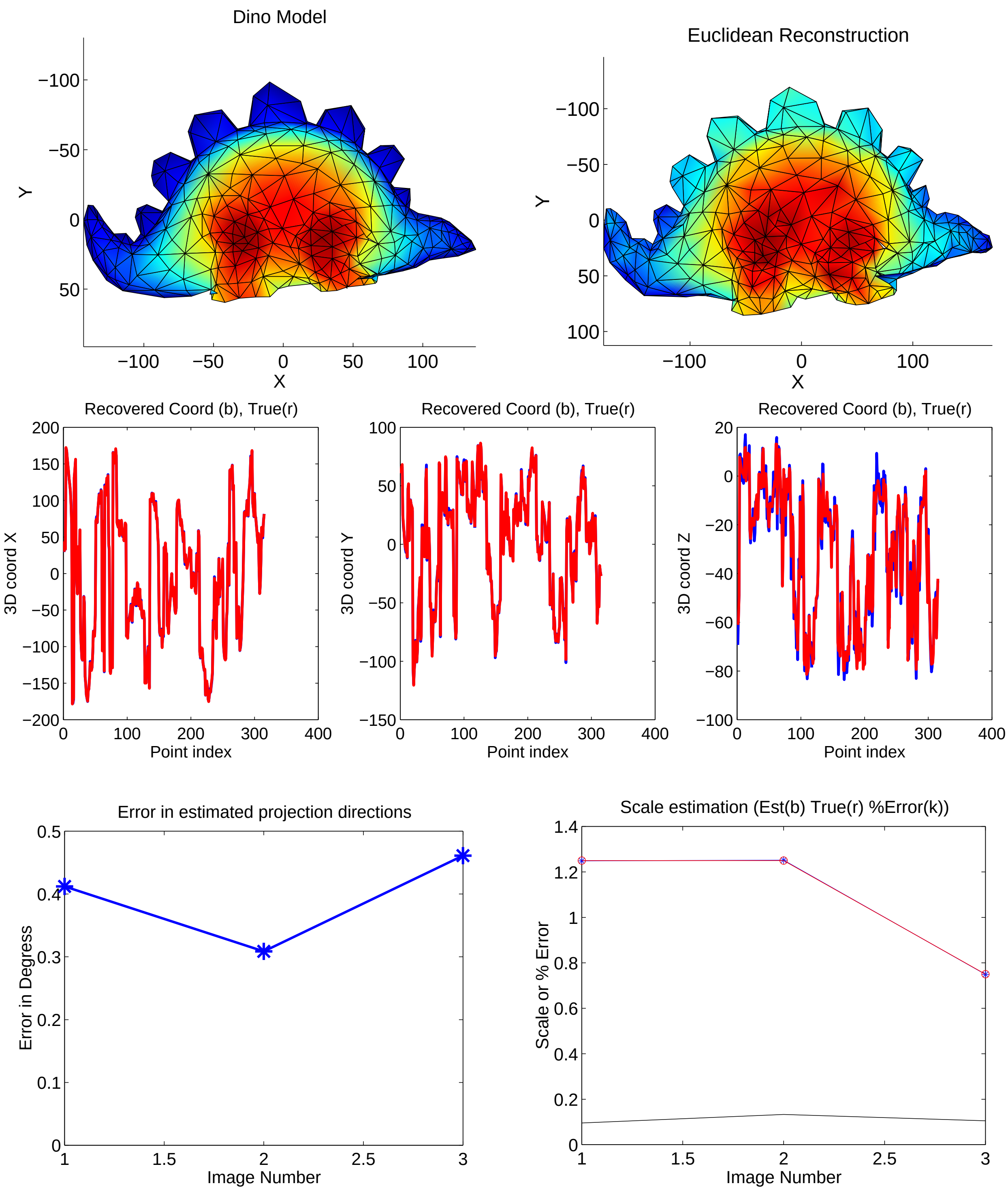
Remaining Ambiguities

The remaining ambiguity in R_q is the **Necker ambiguity**, that is, R_q could be a reflection (say $R_q = diag(1, 1, -1)$). Effectively, with orthographic projection we cannot tell the difference between a concave-in shape viewed from the left, and the reflected concave-out shape viewed from the right. Unlike the previous two ambiguities, this ambiguity does not persist (mathematically) when we switch to perspective projection.

For $J = 2$ orthographic views there is an additional ambiguity, known as the **bas-relief** ambiguity. For this ambiguity, there is an additional unknown parameter (in K_q above), which ties the overall depth variation of the shape to the amount of rotation between the two cameras. See `orthoMassageDino.m`.

Refs: See the classic paper by Koenderink and van Doorn (1991).

Dino Example, Orthographic Case



Introduction to Projective Reconstruction

Returning to perspective projection, recall the form of the original bundle adjustment objective function in (4):

$$\mathcal{O} \equiv \sum_{j,n} \left\| \begin{pmatrix} x_{j,n} \\ y_{j,n} \end{pmatrix} - \frac{1}{\vec{e}_3^{\,T} M_j \vec{P}_n} \left(I_2, \ \vec{0} \right) M_j \vec{P}_n \right\|^2 .$$

It might be tempting to modify this objective function by multiplying each term in the sum by the depths, $z_{j,n} = \vec{e}_3^{\,T} M_j \vec{P}_n$. In doing so, we obtain a *reweighted* objective function:

$$\mathcal{O}' = \sum_{j,n} \left\| z_{j,n} \begin{pmatrix} x_{j,n} \\ y_{j,n} \\ 1 \end{pmatrix} - M_j \, \vec{P}_n \right\|^2 . \qquad (18)$$

The unknowns in (18) are the depths $z_{j,n}$, the camera matrices M_j, and the 3D scene points $\vec{P}_n$, for $j = 1, \ldots, J$ and $n = 1, \ldots, N$.

The form of (18) suggests the following factorization approach.

Projective Factorization

Suppose we knew the projective depths $z_{j,n}$. We could then build a new data matrix C in terms of the 3D points $z_{j,n}\vec{p}_{j,n}$. Accordingly, we can form $C = [z_{j,n}\,\vec{p}_{j,n}]$, by stacking the depth-scaled image points just as we did in the orthographic case. Now C is $3J \times N$.

The perspective projection of the scene point $\vec{P}_n$ onto the j^{th} image plane (1) can be written as

$$z_{j,n}\,\vec{p}_{j,n} \;=\; M_j\,\vec{P}_n\;,$$

where M_j is now 3×4, and $\vec{P}_n$ is 4×1.

By stacking up the camera matrices, M_j, to form the $3J \times 4$ matrix M, and letting $P = (\vec{P}_1, \ldots, \vec{P}_N)$ be the $4\times N$ shape matrix, we obtain a factorization of the data matrix:

$$C \;=\; MP\;. \tag{19}$$

Of course, this assumes we knew the correct depths $z_{j,n}$ that we used to form C. And as above, from the form of (19) it is clear that C is, at most, rank 4 (ignoring noise).

Iterative Projective Factorization

Suppose we normalize $B = CL$ so the columns have unit length (using a diagonal matrix L). Then we factor B using SVD:

$$B = W\Sigma V^T , \tag{20}$$

where we set all but the first 4 singular values to zero. Equivalently, we have W is $3J \times 4$, Σ is 4×4, and V^T is $4 \times N$.

We can rewrite the n^{th} column of B as $Z_n \vec{z}_n$, where Z_n is a $3J \times J$ matrix obtained from the image points $\vec{p}_{j,n}$ and the n^{th} weight $L_{n,n}$. Here $\vec{z}_n = (z_{1,n}, \ldots, z_{J,n})^T$ comprises the projective depths for the n^{th} point in all J frames. We then update $\vec{z}_n$ to better match the current factorization. That is, we wish to minimize

$$||\, Z_n \vec{z}_n - W\Sigma V^T \vec{e}_n \,|| \tag{21}$$

wrt $\vec{z}_n$, subject to the constraint that the updated column of B still has unit length, i.e., $||Z_n \vec{z}_n|| = 1$. Here $\vec{e}_n$ is the n^{th} standard unit vector, $e_{n,i} = \delta_{n,i}$.

Once all the projective depths have been updated, we reform the data matrix C and the normalized data matrix B, and redo the factorization (20), etc. This process is iterated until convergence.

Details for Iterative Projective Reconstruction

Assume that, at the beginning of iteration i we have a current guess (or estimate) for the depths $z_{j,n}^{(i)}$. With these depths we formulate the data matrix:

$$C^{(i)} \;=\; [\, z_{j,n}^{(i)}\, \vec{p}_{j,n} \,] \;\equiv\; (\, \vec{c}_1^{\,(i)}, \ldots, \vec{c}_N^{\,(i)} \,) \tag{22}$$

Let $L^{(i)}$ be an $N \times N$ diagonal matrix with elements $L_{n,n}^{(i)} = 1/||\vec{c}_n^{\,(i)}||$. Then define the normalized data matrix

$$B^{(i)} \;=\; C^{(i)}\, L^{(i)} \tag{23}$$

Then, we form the best rank 4 approximation to $B^{(i)}$, denoted $\hat{B}^{(i)}$. That is, compute the SVD, $B^{(i)} = W \Sigma V^T$, and then define

$$\hat{B}^{(i)} = W \hat{\Sigma} V^T \;, \tag{24}$$

where $\hat{\Sigma}$ is simply Σ with all but the first 4 singular values set to zero.

Now, for the update of depths of the n^{th} point (at all J frames), we define $Z_n^{(i)}$ to be the matrix that satisfies

$$Z_n^{(i)}\, \vec{z}_n^{\,(i)} \;=\; \vec{c}_n^{\,(i)} \;, \tag{25}$$

where $\vec{z}_n^{\,(i)} = (\, z_{1,n}^{(i)}, \ldots, z_{J,n}^{(i)} \,)$ comprises the current depth estimates for the n^{th} scene point in all J frames. That is, $Z_n^{(i)}$ is the $3J \times J$ matrix given by

$$Z_n^{(i)} \;=\; L_{n,n}^{(i)} \begin{pmatrix} \vec{p}_{1,n} & \vec{0} & \cdots & \vec{0} \\ \vec{0} & \vec{p}_{2,n} & \cdots & \vec{0} \\ \vdots & \vdots & \ddots & \vdots \\ \vec{0} & \vec{0} & \cdots & \vec{p}_{J,n} \end{pmatrix} . \tag{26}$$

The only change in $Z_n^{(i)}$ at each iteration is the normalization constant.

Then, to find the depth update for the n^{th} point (for J frames), we solve for $\vec{z}_n^{\,(i+1)}$ which minimizes

$$||Z_n^{(i)}\, \vec{z}_n^{\,(i+1)} \;-\; \vec{\hat{b}}_n^{(i)}\, || \;, \tag{27}$$

subject to the constraint $||Z_n^{(i)}\, \vec{z}_n^{\,(i+1)}|| = 1$, where $\vec{\hat{b}}_n^{(i)}$ is the n^{th} column of $\hat{B}^{(i)}$. In the tutorial code `projectiveMassageDino.m`, this update of $\vec{z}_n$ is done with one step along the gradient direction for the constrained optimization problem. Given the new depths, we begin the next iteration with the formation of the data matrix as above, but now using depths $z_{j,n}^{(i+1)}$.

The convergence of this algorithm is discussed in the paper.

Projective Reconstruction

Upon convergence we have a projective factorization $B = W\Sigma V^T$. As in the orthographic case, this factorization is only unique up to a nonsingular matrix H. In this case, H is a 4×4, 3D homography matrix. In particular, we have the factorization, $C = BL^{-1} = MP$ with

$$P = H^{-1}\,\Sigma\,V^T\,L^{-1} \tag{28}$$

$$M = W\,H\,. \tag{29}$$

Since the shape matrix P is known up to a 3D homography H, this is called a **projective reconstruction**.

This projective reconstruction can be "upgraded" to a metric reconstruction by using information about the camera matrices M_j to constrain the 3D homography matrix H. In order to understand this, we must first introduce the absolute dual quadric from projective geometry.

Absolute Dual Quadric (Canonical Coords)

The equation of a plane in 3D is

$$\vec{m}^{\,T}\,\vec{P} = 0\ ,$$

where $\vec{P} = (X, Y, Z, 1)^T$ is a 3D point written in homogeneous coordinates.

Imagine two planes, with coefficient vectors $\vec{m}_1$ and $\vec{m}_2$. Then the angle between these two planes is θ where

$$\cos(\theta) \;=\; \frac{\vec{m}_1^T \hat{Q}_\infty \vec{m}_2}{\sqrt{(\vec{m}_1^T \hat{Q}_\infty \vec{m}_1)\,(\vec{m}_2^T \hat{Q}_\infty \vec{m}_2)}}\ . \tag{30}$$

Here $\hat{Q}_\infty$ is the absolute dual quadric in canonical coordinates,

$$\hat{Q}_\infty = \begin{pmatrix} I_3 & \vec{0} \\ \vec{0}^T & 0 \end{pmatrix}\ . \tag{31}$$

Absolute Dual Quadric (General Coords)

Suppose H is any nonsingular 3D homography matrix, and consider the projective coordinates $\vec{P}' = H\vec{P}$. The planes $\vec{m}_k \cdot \vec{P} = 0$ can be expressed in these new coordinates as

$$\vec{m}_k' \cdot \vec{P}' \;=\; 0, \quad \text{where } \vec{m}_k' \;=\; H^{-T}\vec{m}_k \,.$$

We can measure the same angle between these two planes using the absolute dual quadric in general projective coordinates, namely

$$Q_\infty = H\hat{Q}_\infty H^T. \tag{32}$$

In fact, it follows that

$$\cos(\theta) \;=\; \frac{(\vec{m}_1')^T Q_\infty \vec{m}_2'}{\sqrt{((\vec{m}_1')^T Q_\infty \vec{m}_1')\,((\vec{m}_2')^T Q_\infty \vec{m}_2')}} \,.$$

The general idea behind upgrading a projective reconstruction to a metric one is to use the absolute dual quadric to express known properties of the camera coordinates, such as the fact that the planes perpendicular to the X, Y, and Z axes are mutually perpendicular (i.e. $\cos(\theta) = 0$).

Upgrading to a Metric Reconstruction

In particular, from (2) and (3) it follows that

$$M_j \, \hat{Q}_\infty \, M_j^T \;=\; \begin{pmatrix} f_j^2 & 0 & 0 \\ 0 & f_j^2 & 0 \\ 0 & 0 & 1 \end{pmatrix} . \tag{33}$$

This is the analogue of equation (15) in the orthographic case. From equation (28) we also have $M = WH$, where W is known from the projective factorization. So

$$M_j \, \hat{Q}_\infty \, M_j^T \;=\; W_j \, H \, \hat{Q}_\infty \, H^T \, W_j^T \;=\; W_j \, Q_\infty \, W_j^T \; . \tag{34}$$

We see that, for the j^{th} camera, (34) and (33) provide 5 linear equations for Q_∞ (6 linear equations if f_j is known). So we have (at least) $5J$ linear constraints on Q_∞.

Since we know Q_∞ is a symmetric 4×4 matrix there are only 10 degrees of freedom to determine, and $J = 2$ frames are enough. (Note we also know Q_∞ has rank 3 and has non-negative eigenvalues.)

Solving for H

Given Q_∞ (which is symmetric, positive semi-definite, rank 3) we can compute its eigenvalue decompositon

$$Q_\infty \;=\; U_q \, \Lambda_q \, U_q^T \;, \quad \Lambda_q \;=\; \mathrm{diag}[\lambda_1, \lambda_2, \lambda_3, 0] \;,$$

with $\lambda_i > 0$. It then follows from (32) that

$$H \;=\; U_q \, \mathrm{diag}[\lambda_1^{1/2}, \lambda_2^{1/2}, \lambda_3^{1/2}, 1] \, A \;, \tag{35}$$

where A is the matrix for a general 3D similarity transform

$$A \;=\; \begin{pmatrix} R & \vec{d} \\ \vec{0}^T & s \end{pmatrix} \;.$$

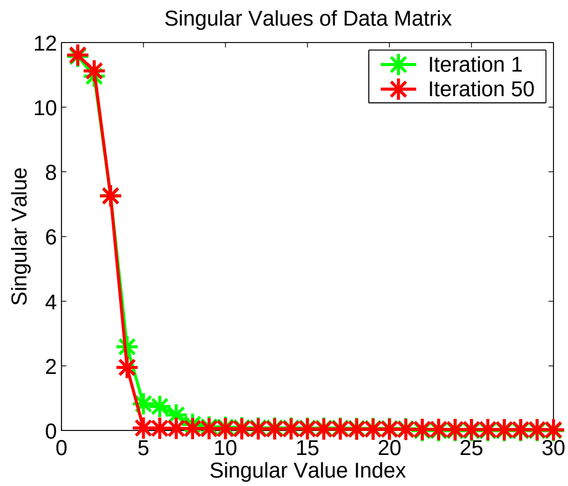
Here R is a unitary matrix. The reason A remains unknown is that the absolute dual quadric $\hat{Q}_\infty$ is invariant to similarity transformations

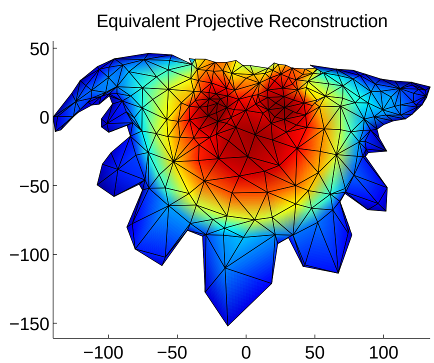
$$A \, \hat{Q}_\infty \, A^T \;=\; \hat{Q}_\infty \;.$$

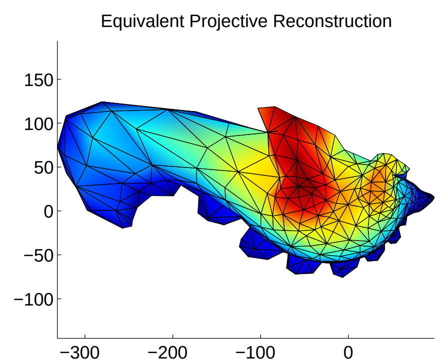
This is easy to verify from the forms of A and $\hat{Q}_\infty$.

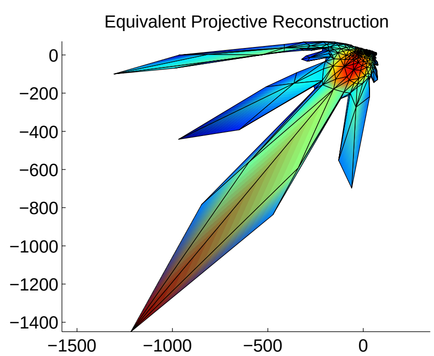
Finally, equations (2, 3, 28) can be used to remove the reflection ambiguity. The only remaining ambiguities are the overall orientation, origin and scale of the world coordinate frame.

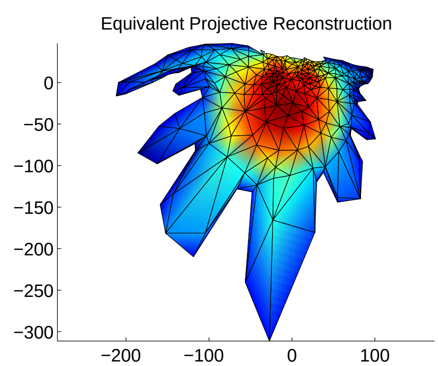
Dino Example, Projective Case

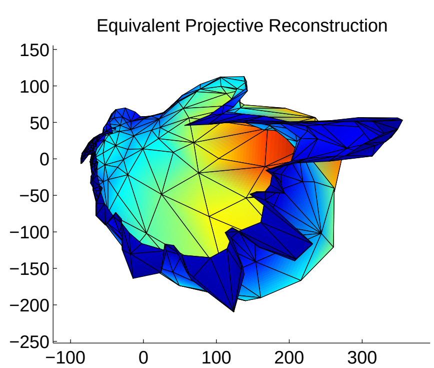


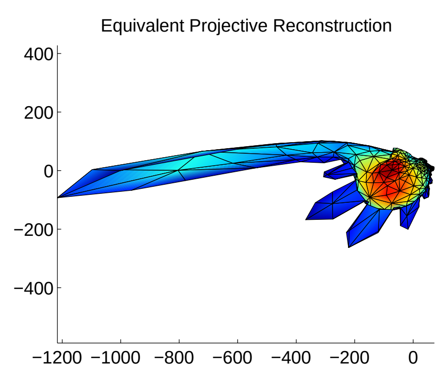




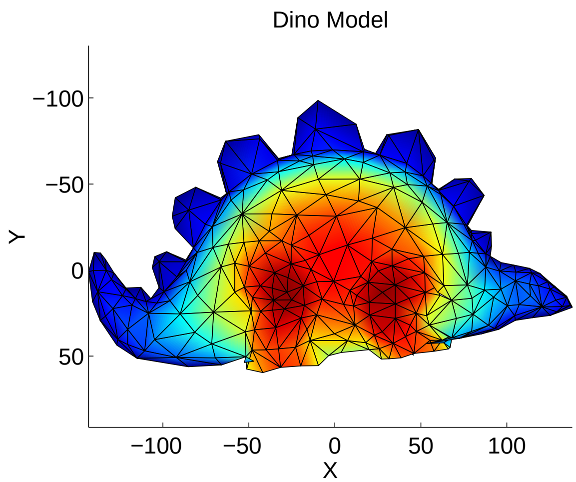


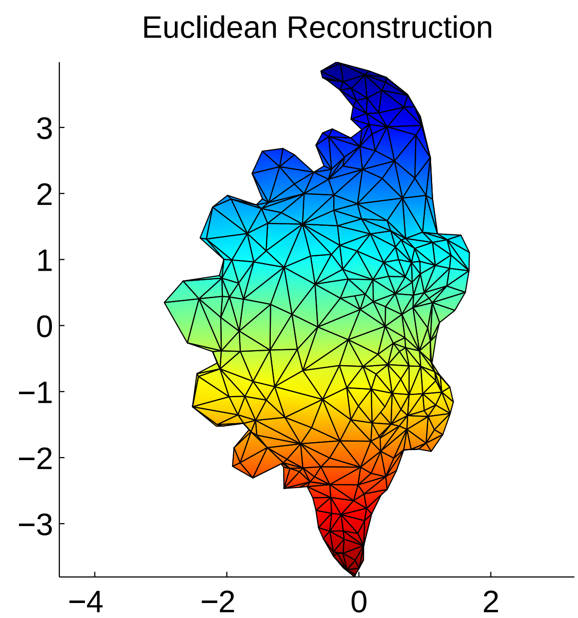


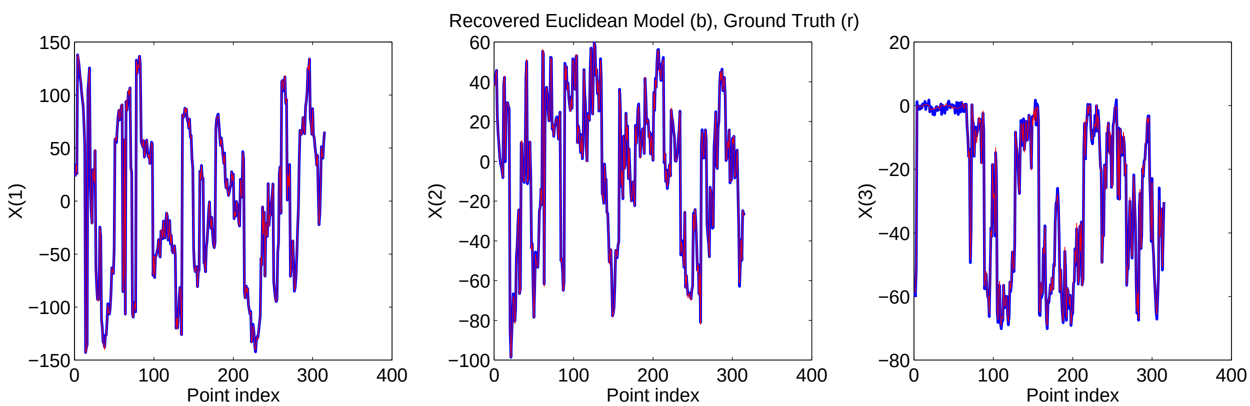




Dino Example, Projective Case







Structure from Motion

The use of the theoretical rank for a set of observations provides a key insight into the structure from motion problems (see Jepson and Heeger, 1991).

Consider a camera travelling through a stationary environment. Then the scene appears to move with translational velocity $\vec{T}$ and angular velocity $\vec{\Omega}$. In the camera's coordinates, the motion of any scene point $\vec{X}$ is

$$\frac{d\vec{X}}{dt} = \vec{T} + \vec{\Omega} \times \vec{X}.$$

Suppose we observe the motion field $\vec{u}(\vec{x}_k)$ at K image points, $\{\vec{x}_k\}_{k=1}^{K}$, in this camera's image. Let $\vec{X}(\vec{x}_k)$ be the 3D scene point associated with the k^{th} image point $\vec{x}_k$. Then it can be shown that $\vec{U}^{\,T} \equiv (\vec{u}_1^T, \ldots, \vec{u}_K^T)$ satisfies

$$\vec{U} = C(\vec{T}) \begin{pmatrix} \vec{z} \\ \vec{\Omega} \end{pmatrix} = A(\vec{T})\vec{z} + B\vec{\Omega}. \tag{36}$$

Here $\vec{z}$ is a K-vector, with elements $z_k = 1/||\vec{X}(\vec{x}_k)||$, $A(\vec{T})$ is a $2K \times K$ matrix that depends linearly on $\vec{T}$, and B is a $2K \times 3$ matrix that depends only on the image points $\vec{x}_k$.

Notice, for $\vec{T} = \vec{0}$ we have $A(\vec{T}) = 0$, and (36) states that the flow field $\vec{U}$ must be in the rank 3 subspace formed by the range of the matrix B. Similarly, for nonzero $\vec{T}$, equation (36) states that the $2K$-dimensional flow field $\vec{U}$ must be in the $K + 3$-dimensional subspace formed by the range of $C(\vec{T})$.

This range condition can be used to identify $\vec{T}$ (up to a speed ambiguity, i.e., $||\vec{T}||$ remains unknown) and $\vec{\Omega}$ given the motion field $\vec{U}$. Moreover, given $\vec{T}/||\vec{T}||$ and $\vec{\Omega}$, equation (36) can be used to solve for the inverse depths $\vec{z}$ (up to an overall scale ambiguity).

Further Readings

R Hartley and A Zisserman, Multiple View Geometry in Computer Vision, Cambridge Univ. Press, 2000.

A Jepson and D Heeger, A fast subspace algorithm for recovering rigid motion, Proc. IEEE Workshop on Visual Motion, Princeton, 1991, pp. 124-131.

J Koenderink and A van Doorn, Affine structure from motion, Journal of the Optical Society of America, 8(2), 1991, pp. 377-385.

S Mahamud, M Hebert, Y Omori and J Ponce, Provably-convergent iterative methods for projective structure from motion, CVPR 2001.

M Polleyfeys, R Koch and L van Gool, Self-calibration and metric reconstruction in spite of varying and unknown intrinsic camera parameters, IJCV 1999.

C Tomasi and T Kanade, Shape and motion from image streams under orthography: A factorization approach, IJCV, Vol. 9 (No. 2), 1992, pp.137-154.