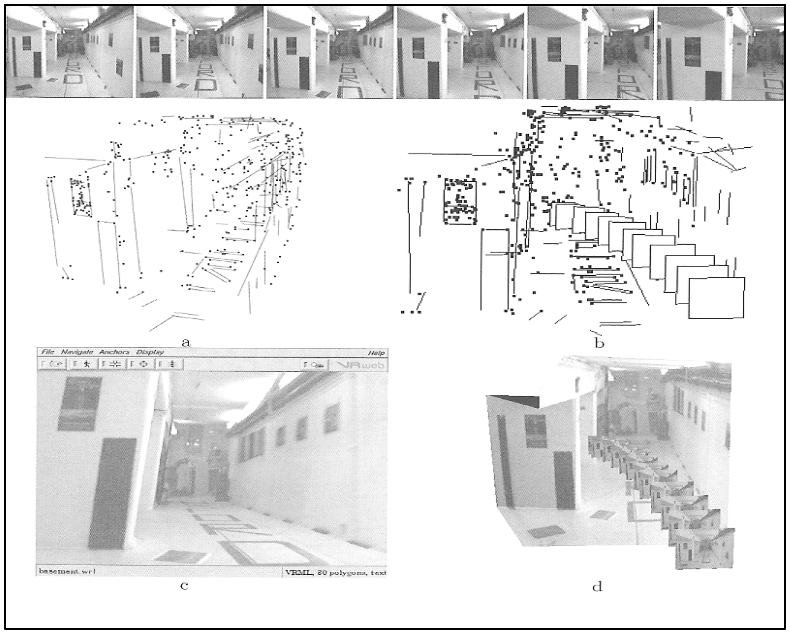
A Summary of Projective Geometry

In the last 10 years a unified approach to creating 3D models from multiple images has been developed by Beardsley[1],Hartley[4,5,9],Torr[1,6] and Zisserman[1,9]. In principle it allows point and line features that can be distinguished from more than one point of view in 2D images to be automatically converted into points in a 3D world coordinate model. If a sufficient number of vertices or points in one image can be matched with points in another image, information about the locations of the points as well as the relative locations, orientations and fields of view of the cameras can be identified. The cameras may or may not be identical.



An example of scene capture from [9]. Images used to create the model are in the top row. The middle left is a rendition of the 3D points located, and middle right shows image planes for each and lines recovered. The lower left is a computer generated reconstruction of the scene from a new point of view, and the lower right shows the camera path through the scene.

The following description of the reconstruction process has been accumulated from the sources listed above, and the complete mathematics and techniques for implementation are found in the book **Multiple View Geometry** [9]. Below is a summary of the steps in this approach. Terms in quotations in the steps below are those of the developers of the technique.

1. Detect at least 8 point features in 2 images obtained from distinct points of view.
2. Identify pairs of points, one in each image, that may represent one feature or location in the world.
3. Using groups of 8 point-pairs at a time, compute candidates for the “Fundamental Matrix F” which relates the images to one another.
4. Use the RANSAC[2] algorithm to find a most likely F, i.e. one that most closely agrees with the largest number of point pair candidates.
5. Repeat steps 1-4 using the second image and a 3rd image.
6. Compute the “Camera Projection Matrices” for images 1-3 from F12 and F23
7. Compute the “Trifocal Tensor” from the Camera Projection Matrices
8. Using the Trifocal Tensor, verify or reject candidate points found in each image.
9. Instantiate 3D model from detected points by finding intersection of lines formed by camera positions and detected 2D points

10. Where 2D image processing indicates a line exists between points, add that line to the model.
11. Continue processing subsequent images against 3D model, adding new model features, refining existing model features and camera internal parameter.

Detecting Points

The first step is to select 2 images taken from similar points of view, so that features in one appear in almost the same location in the other. Easch image is then searched for significant point features. For this task, the Harris corner detector[3] will be employed. The detector calculates a value denoting the amount of intensity gradient between a certain pixel and those around it with a convolutional mask. If the convolution output for a particular pixel is above a threshold, a strong positive correlation exists and the pixel is a corner or edge. Sorting may be employed after the image has been convolved to identify the most distinct corners.

Detecting Edges

Lines are essential to defining the connection of polygons. To detect lines or edges the Canny edge detector is used[7]. This is similar to the Harris corner detector. This approach has been augmented recently, and the detector can be made more accurate by an approach in [8]. The method involves marking edges with a gradient magnitude larger than its neighbors but in the wrong direction as a “minor edge”. These minor edges occur when a picture has a section with many intensity changes in a small area (the intersection of squares pattern on a chess board). The improved detector can sense lines right up to the intersection of the intensity change. Output data is significantly improved and thus the technique will be employed in the software.

Once points and corners in each image, or frame, have been identified putative correspondences are established. A feature in one image is said to correspond with a feature in another if both represent the same 3D point. For each location in one image where a feature was detected, an area within a circular region around that position in the second image is searched for a similar feature. If there is a match the point is tentatively valid. The description of world model reconstruction from 3 views is summarized from [2],[3],[4],[9].

The Fundamental Matrix

These point matches can then be used to calculate a 3x3 rank 2 Fundamental Matrix F relating the mapping of the points in image 1 to those in image 2. Define a point found in a 2D image by:

$x = \{x, y, w\}$, where x, y are position, with w making them homogeneous.

The term w represents the fact that these are real points and not ideal points on the plane at infinity (w may be set to 1). Then the Fundamental Matrix F is defined by

$x'^T Fx = 0$, where x, x' are points in images 1,2

Since the *x, x’* are homogeneous coordinates (x, y, 1), F may be computed up to a scale factor using 8 corresponding point pairs *x’*, *x*.. An important property of F is its singularity: F is of rank 2. However, solutions for F obtained from real data will not generally be singular. This can be corrected using singular value decomposition[9]. This matrix contains 3D information about the points which have been matched across the two images, much in the same way our two eyes give us perspective to detect depth. Since F is singular, x’ can only be computed from an x up to a scale factor, that is, a ray projected from camera location 1 through the image point x will map onto a line crossing the second image, and the (unknown) distance of **X** from camera location 1 will determine where on the line **x'** lies. This is shown graphically in Figure 1.

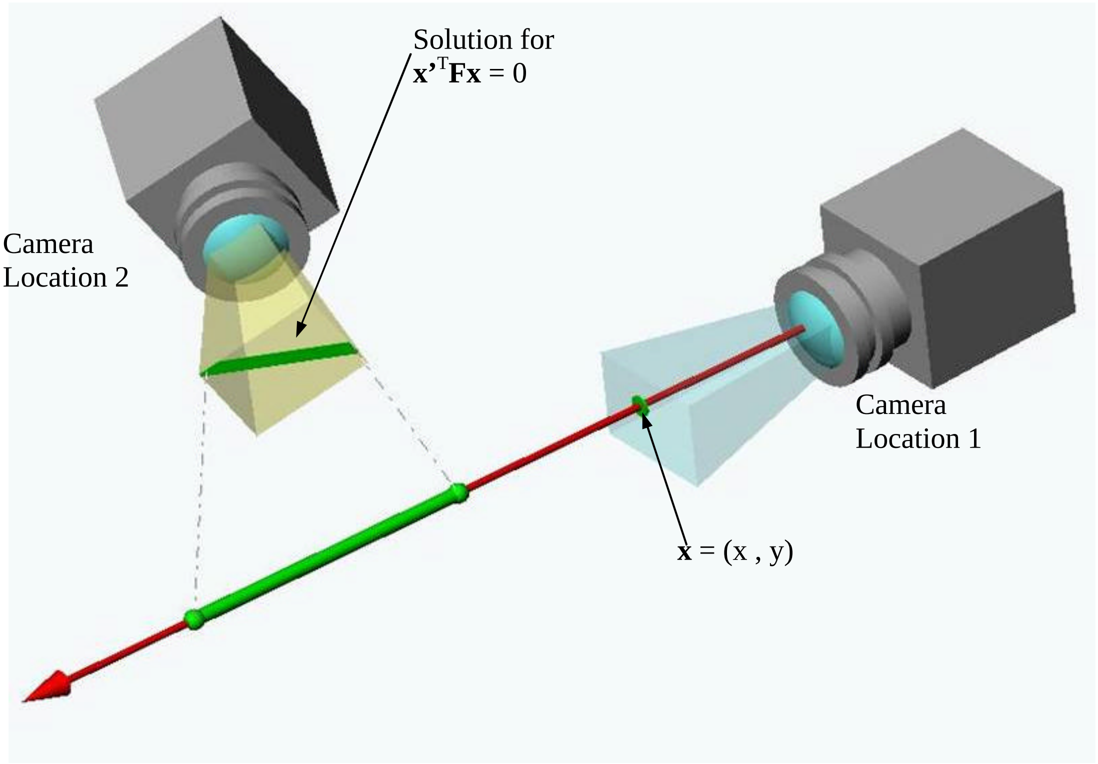


Figure 1.

Projection of a point x in image 1 into a line in image 2 with the Fundamental Matrix.

Verify The Fundamental Matrix F Using The RANSAC Algorithm

To verify point pairs and select the best possible F, the RANSAC method is employed. This iterative computational technique assumes that some matches are good and some not, and that it makes more sense to separate them than average them as would be done with a least squares fit. It compares candidate funda-mental matrices obtained for many point pairs to find a matrix that is supported by the greatest number of pairs. The RANSAC algorithm is outlined below:

1. Randomly select a sufficiently large group of sets of 8 point pairs.
2. Compute a fundamental matrix F(x, x') for set of point pairs.
3. Test which fundamental matrix has the most pairs supporting it, i.e. $| x'^T Fx | \le e$, for some small e.
4. Save this matrix and support number.
5. If a sufficient number of permutations have been tested, then go to 6. Else go back to 1 and select another group.
6. Select the matrix with the highest number of included points and mark these as the "inliers". Mark all others as "outliers".
7. Recompute F using only inliers and verify that all inliers are still inliers.

To compute just how many points represent a “sufficiently large group of sets of 8 point pairs”, the procedure uses the equation below to calculate the probability that a good sub-sample has been selected [1]:

$$Y = 1 - (1 - (1 - e)^{p})^{m} ,$$

where ε is the fraction of contaminated data, and p the number of points considered in each sample. The value for m, the number of sets that should be considered, must be chosen such that Y >= .95, for a particular p and ε. If ε is unknown, as is always the case in reality, then a putative estimate must be used, and updated as the process continues. This process narrows down the results to a group of points that result in a shared Fundamental Matrix for the two images and corresponding (motionless) features. Once this has been done for image 1 and 2, a third image should be selected, and the operation performed on images 2 and 3.

Compute the Camera Projection Matrices

Once a fundamental matrix is found, establishing the relationship between the first two images, the camera projection matrices P, P’ can be computed. This holds the information relating the camera position, orientation, xy aspect ratio and effective focal length for one image to the parameters for the camera that obtained the other image. The camera matrices allow the construction of 3D points from 2D features that correspond to each other.

The next step is to calculate the camera projection matrices P for each image. These 3x4 matrices represent the projection of the points X in 3D space to the points x found in the image, and will allow the calculation of their original 2D coordinates:

$$x = PX , \qquad \text{where} \qquad X = \{x, y, z, 1\} \text{ and } x = \{x, y, w\}$$

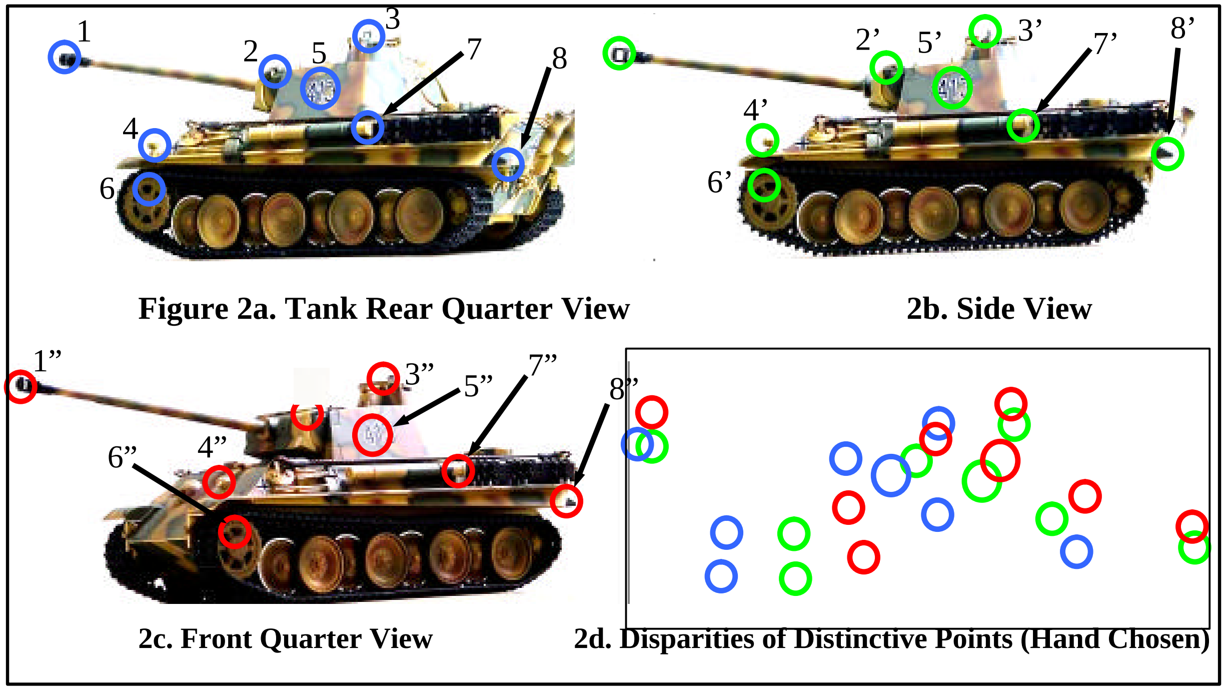


Figure 2a. Tank Rear Quarter View

2b. Side View

2c. Front Quarter View

2d. Disparities of Distinctive Points (Hand Chosen)

Figure 2. Three images and corresponding points in each image.

Since the projection matrices relate one camera (or frame) to another, the center of the first camera coordinate system can be chosen by setting P = [I|0] for the first camera[4], where I is the 3x3 identity matrix and 0 is a null 3 vector. Given this definition, the camera matrices for other images P´ and P´´ can be calculated by the method below[1,4]. Given the Fundamental Matrix **F** for two 2 images,

$$\text{define} \qquad P \equiv [I \mid 0] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \equiv \text{ camera center for first image}$$

Then $P' = [m' \mid e']$ where e' is obtained by solving

$F^T e' = 0$ (Null space of F), and m' is obtained by solving

$F = [e']_x m'$ where $[a]_x b = a \times b$, the outer product of a and b.

The process is repeated with images 2 and 3 to obtain P''.

Verify Points and Lines Found in each Image Across the Image Triplet

Once the camera projection matrices P´ and P´´ are found for images 2 and 3, the Trifocal Tensor (TFT) can be computed using these matrices. This is a 3x3x3 tensor that allows a point found in two images to be projected onto a third image. With this tensor, all the detected point pairs can be projected into the image from which they were not derived, and can be verified by the presence of a 2D point feature at that location in that image:

$$T_{ijk} = P'_{ji} P''_{k4} - P'_{j4} P''_{ki} \,.$$

This 3x3x3 homogeneous tensor can then be used to verify points and lines[1] across triplets of images, allowing us to add more points to the 3D model. Any point found in two images can be transferred to the third using the tensor as shown below. If a matching feature is found where it is supposed to be, then it is considered verified[1], and added to the model. The same tensor can be used to transfer points and lines[9]:

$$x''_l = x'_i \sum_{k=1}^{k=3} x_k T_{kjl} - x'_j \sum_{k=1}^{k=3} x_k T_{kil} \qquad \text{(Point Transform), and}$$

$$l_i = \sum_{j=1}^{j=3} \sum_{k=1}^{k=3} l'_j l''_k T_{ijk} \qquad \text{(Line Transform).}$$

Figure 3 shows the graphical representation of the point transform through TFT from **x** and **x'** onto a 3rd camera image plane point **x''**. The uncertainty in X results from measurement error: The rays from Camera 1 and Camera 2 will never exactly intersect.

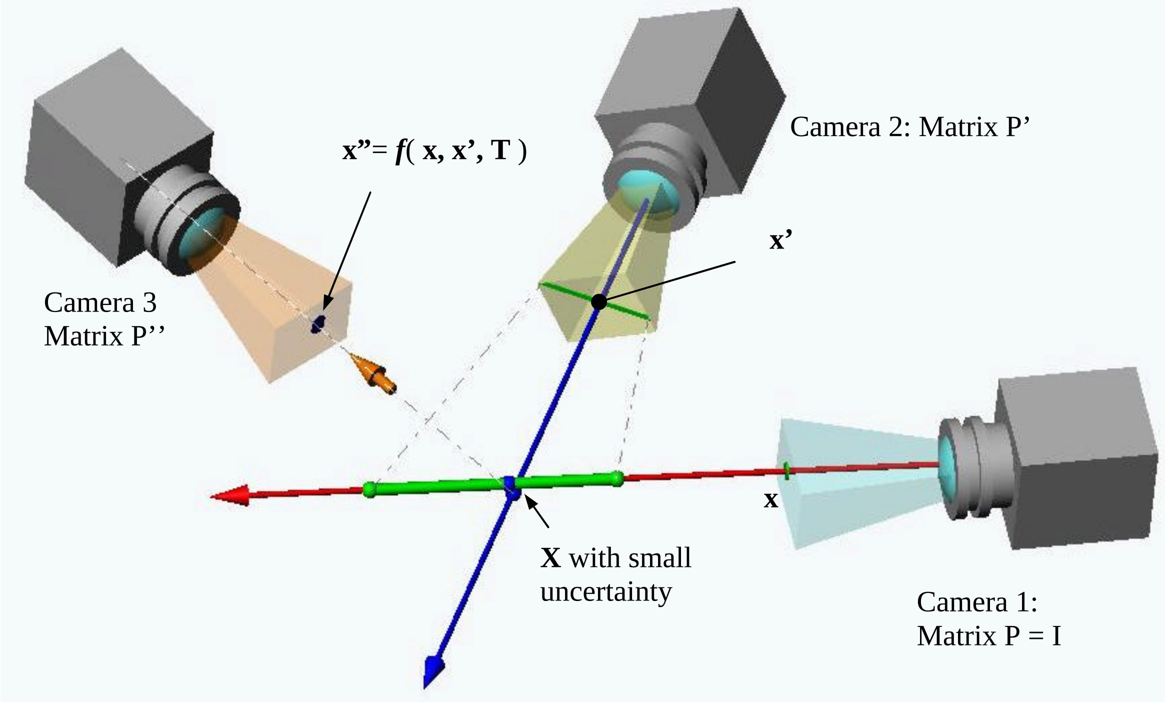


Figure 3. Determining x’’ in a third image from two images and camera parameters.

Instantiate the 3D Model

Thus, given the camera projection matrices, the 3D positions of the points in the images can be found. This creates an instantiation of a 3D model of points and lines from a triplet of images. Additional images may be added by combining them with 2 images from an earlier triplet. Of course with noisy data the camera matrices will not give 3D positions that precisely match up across different image triplets. This positional error requires a least squares (LMS) or other minimization scheme be applied to find the best position [5].

Using the model we have just instantiated, we can check all other lines and points as we detect them in the coming images. Given the fundamental matrices, the trifocal tensor for any 3 images can be used to verify the validity of features found in the additional images and the validity of the 3D model.

References

1. P. Beardsley, P.H.S. Torr and A. Zisserman “3D model acquisition from extended image sequences”. Technical report, Dept of Eng Science, University of Oxford, 1996
2. M.A. Fischler and R.C. Bolles. “Random sample consensus: a paradigm for model fitting with application to image analysis and automated cartography”. *Commun. Assoc. Comp. Mach*., vol. 24:381-95,1981.
3. C.G. Harris and M. Stephens. “A combined corner and edge detector”. In *Fourth Alvey Vision Conference*, pages 147-151, 1988.

4. R.I. Hartley. “Estimation of relative camera positions for uncalibrated cameras”. In *Proc. 2nd European Conference on Computer Vision*, pages 579-587. Springer Verlag 1992.
5. R.I. Hartley. And P. Sturm., (1994) “Triangulation”. In *AIUWS*, pages 957-966.
6. P.H.S. Torr. *Motion segmentation and outlier detection*. PhD thesis, Dept. of Engineering Science, University of Oxford, 1995.
7. J.F. Canny. “A computational approach to edge detection”. *IEEE Transactions Pattern Analysis and Machine Intelligence*, 8:769-798,1986.
8. L. Ding and A. Goshtasby, "On the Canny edge detector," *Pattern Recognition*, vol. 34, 2001, 721-725
9. R. Hartley and A. Zisserman, Multiple View Geometry, Cambridge Press, 2000.