Lecture 4: Triangulation and Homography Estimation

1 Triangulation

The problem of finding an unknown 3D-point position $\mathbf{X}$ from measured projections $\mathbf{x}_i$, $i = 1, ...n$ into known cameras P_i is called **triangulation**. We have previously seen that if $\mathbf{X}$ is to project to the measurements then the camera equations

$$\lambda_i \mathbf{x}_i = P_i \mathbf{X}_i, \quad i = 1, ..., n. \tag{1}$$

must hold. Therefore we need to find $\mathbf{X}$ and λ_i that solve these equations. For simplicity we will assume that $\mathbf{X}$ is a regular point, not at infinity, with homogeneous coordinates $\mathbf{X} = \begin{bmatrix} X \\ 1 \end{bmatrix}$, where $X = \begin{pmatrix} x \\ y \\ z \end{pmatrix}$ are the regular Cartesian 3D-coordinates. Each projection gives us 3 equations and for n projections we therefore have $3n$ equations. Each projection also introduces one new unknown, the depth λ_i, and therefore we have $3 + n$ unknowns. To solve the problem we need

$$3n \geq 3 + n \Leftrightarrow n \geq \frac{3}{2}. \tag{2}$$

Thus two projections are enough.

If $P_i = \begin{bmatrix} A_i & t_i \end{bmatrix}$, where A_i is 3×3 and invertible the camera equations can be written

$$\lambda_i \mathbf{x}_i = A_i X + t_i \Leftrightarrow X = -A_i^{-1} t_i + \lambda_i A_i^{-1} \mathbf{x}_i. \tag{3}$$

Note that $C_i = -A_i^{-1} t_i$ are the 3D-coordinates of the camera center of P_i (see Lecture 1). The geometric interpretation of the above expression is that X should be on a line going though C_i with directional vector $A_i^{-1}\mathbf{x}_i$. When we vary λ_i we get different 3D points on the line and all of them project to $\mathbf{x}_i$. To find the correct X we need to determine the intersection of the lines coming from each camera as illustrated in Figure 1.

Exercise 1. Find the 3D-point $\mathbf{X}$ that projects to $\mathbf{x}_1 = \begin{pmatrix} 1/2 \\ 1/2 \\ 1 \end{pmatrix}$ in $P_1 = \begin{bmatrix} I & 0 \end{bmatrix}$ and $\mathbf{x}_2 = \begin{pmatrix} 0 \\ 1/2 \\ 1 \end{pmatrix}$ in $P_2 = \begin{bmatrix} I & \begin{pmatrix} -1 \\ 0 \\ 0 \end{pmatrix} \end{bmatrix}$.

In some special cases we cannot determine a unique intersection between the lines of (3). If the camera centers C_i, $i = 1, ..., n$ and the 3D point X are all located on one line in 3D, the expressions (3) become identical and all points on the 3D-line fulfill them. This is called a **degenerate configuration** and is illustrated to the right in Figure 1. While it is very unlikely that all camera centers and the 3D point should be exactly on the same line in any real case, degenerate configurations are important since problem instances that are close to degeneracy often exhibit numerics and become very sensitive to noise. Hence in practical cases when the camera centers and the 3D point are close to being on a line we can expect that we will get poor accuracy when we try to recover it.

1.1 Noisy Recovery using DLT

In practical cases our measurements will not be exact but will be affected by noise, and therefore we cannot expect the viewing rays to have an exact intersection point, as illustrated in Figure 2. Therefore we need to solve the

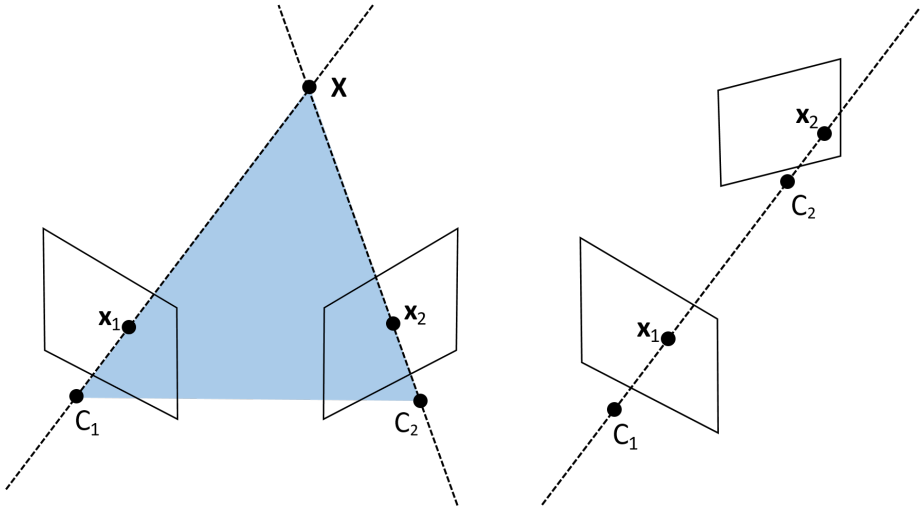

Figure 1: Illustration of triangulation. Left: The sought point $\mathbf{X}$ is in the intersection of the two viewing lines. Right: If the camera moves so that the two viewing lines intersect for all points the problem becomes degenerate and we cannot determine $\mathbf{X}$.

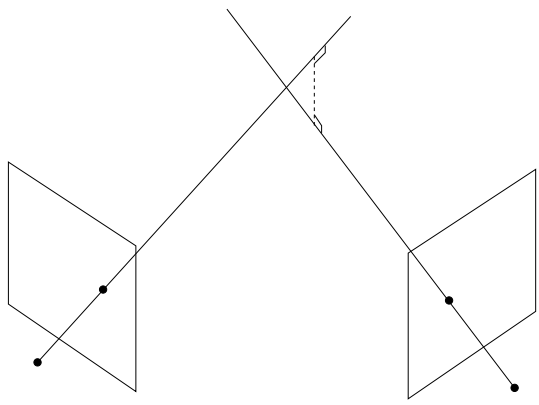

Figure 2: In case of noisy measurements we cannot expect the viewing rays to intersect in a point. In such cases we need to solve the camera equations approximately.

camera equations approximately in some sense. One way of doing this is to use the DLT method from Lecture 3. The problem is linear in the unknowns λ_i and X, so we can find the homogeneous least squares solution by re-formulating the problem as

$$Mv = 0, \tag{4}$$

with

$$M = \begin{bmatrix} P_1 & -\mathbf{x}_1 & 0 & \cdots & 0 \\ P_2 & 0 & -\mathbf{x}_2 & \cdots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ P_n & 0 & 0 & \cdots & -\mathbf{x}_n \end{bmatrix}, \tag{5}$$

and

$$v^T = \begin{bmatrix} X^T & \lambda_1 & \lambda_2 & \cdots & \lambda_n \end{bmatrix}. \tag{6}$$

As in Lecture 3 we solve

$$\min_{\|v\|^2=1} \|Mv\|^2 \tag{7}$$

by computing the singular value decomposition $M = USV^T$ of M and let v be the last column of V.

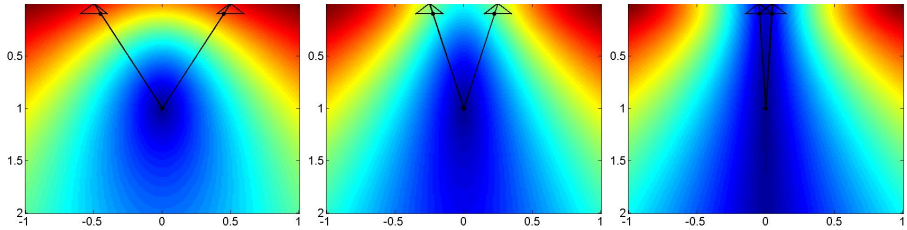

Figure 3: Visualization of the DLT objective for 2-dimensional cameras. When the cameras approach the degenerate case (by moving towards each other) the objective is roughly constant along the viewing direction.

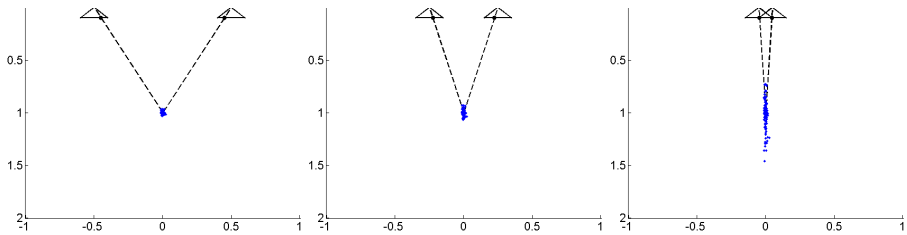

Figure 4: Estimation of a 2D point from 1D images using the DLT objective. Close to degenerate cases the depth of the point becomes uncertain.

Figure 3 shows three examples of the DLT objective for triangulation. We use 1D-cameras for visualization purposes. Here a camera is a 2×3 matrix which takes a point in the plane $\mathbb{P}^2$ and projects it onto the camera line, essentially providing a direction towards the point. For a given $2D$ point $\mathbf{X} = \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$ we compute

$$f(\mathbf{X}) = \min_{\lambda_1^2+\lambda_2^2+\|\mathbf{X}\|^2\gamma^2=1} \left\| M \begin{bmatrix} \gamma\mathbf{X} \\ \lambda_1 \\ \lambda_2 \end{bmatrix} \right\|^2, \tag{8}$$

and plot the result as a function of $\mathbf{X}$. We also plot the cameras and the projection of the optimal 2D point in the same figure. When the cameras are close to each other the level curves of the objective function become thin and elongated. As a result the location of the triangulated point will be more difficult to determine when noise is added to the problem. In the three images in Figure 4 perform the same triangulation experiment 100 times with noise added to the image projections. The noise is the same for all the three images only the locations of the cameras differ. The estimated 2D-points are the blue dots. When the cameras are sufficiently far apart all estimations end up close to the true 2D-point. In contrast when the cameras are close to the degenerate case, the result becomes very sensitive to noise and the exact distance from the cameras are difficult to determine accurately.

2 Homography Estimation

Projective transformations (or homographies) were introduced in Lecture 2. In practical computer vision problems they occur in a number of common settings. Figure 5 shows a plane induced homography. If a set of points from a 3D-plane is projected into two images then there is a projective transformation $\mathbb{P}^2 \to \mathbb{P}^2$ between the two images (see Assignment 1 for a proof). Another example is when two images are captured by cameras that have the same position, only differing by orientation and possibly calibration. This is useful for building panoramas (see Section 2.3). When we are solving uncalibrated reconstruction problems all possible reconstructions are related by a projective transformation $\mathbb{P}^3 \to \mathbb{P}^3$ (see Lecture 3).

In homography estimation we want to find a projective transformation from $\mathbb{P}^k$ to $\mathbb{P}^k$. We will describe the problem for $k = 2$ but the procedure is exactly the same for any dimension. Given two sets of points $\mathbf{y}_i$ and $\mathbf{x}_i$ that are

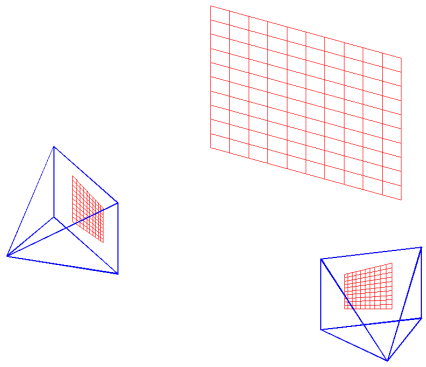

Figure 5: Example of a plane induced homography.

related by a homography H we want to solve

$$\lambda_i \mathbf{y}_i = H\mathbf{x}_i, \quad i = 1 \dots n. \tag{9}$$

The matrix H has 9 entries, but since its scale is arbitrary we can assume that one of them is fixed. For n points we therefore have $3n$ equations and $8 + n$ unknowns and the problem can therefore be solved if

$$3n \geq 8 + n \Leftrightarrow n \geq 4. \tag{10}$$

Note that in contrast to the triangulation problem, where we obtained an overdetermined system when using two point projections, the problem can be solved exactly if $n = 4$. Before we compute the solution we make one simplification. We assume that

$$\mathbf{x}_1 = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, \ \mathbf{x}_2 = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}, \ \mathbf{x}_3 = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \text{ and } \mathbf{x}_4 = \begin{pmatrix} a \\ b \\ c \end{pmatrix}, \tag{11}$$

where a, b, c are some known numbers. Note that $\mathbf{x}_1$,$\mathbf{x}_2$ are points at infinity. Although we cannot (directly) observe such points in a real image, this assumption simplifies the following derivations. We treat the general case at the end of this section through a change of variables.

Under the assumption (11) we have

$$H\mathbf{x}_i = \begin{cases} h_i & i = 1, 2, 3 \\ ah_1 + bh_2 + ch_3 & i = 4 \end{cases}, \tag{12}$$

where h_i is column i from H. Therefore H must be of the form

$$H = \begin{bmatrix} \lambda_1 \mathbf{y}_1 & \lambda_2 \mathbf{y}_2 & \lambda_3 \mathbf{y}_3 \end{bmatrix}. \tag{13}$$

To determine the unknowns λ_i, $i = 1, 2, 3$ we consider the second case of (12) which inserted in (9) gives

$$\lambda_4 \mathbf{y}_4 = H\mathbf{x}_4 = \lambda_1 a\mathbf{y}_1 + \lambda_2 b\mathbf{y}_2 + \lambda_3 c\mathbf{y}_3 = \begin{bmatrix} a\mathbf{y}_1 & b\mathbf{y}_2 & c\mathbf{y}_3 \end{bmatrix} \begin{pmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \end{pmatrix}. \tag{14}$$

Note that

$$\begin{bmatrix} a\mathbf{y}_1 & b\mathbf{y}_2 & c\mathbf{y}_3 \end{bmatrix} = \underbrace{\begin{bmatrix} \mathbf{y}_1 & \mathbf{y}_2 & \mathbf{y}_3 \end{bmatrix}}_{:=Y_{1:3}} \underbrace{\begin{pmatrix} a & 0 & 0 \\ 0 & b & 0 \\ 0 & 0 & c \end{pmatrix}}_{:=D_{\mathbf{x}_4}}. \tag{15}$$

Since the scale is arbitrary we can assume that $\lambda_4 = 1$. The homography can therefore be uniquely determined when the matrix $Y_{1:3}D_{\mathbf{x}_4}$ is invertible, and is given by the formula

$$H = Y_{1:3}D_{\lambda_{1:3}} \tag{16}$$

$$\lambda_{1:3} = D_{\mathbf{x}_4}^{-1}Y_{1:3}^{-1}\mathbf{y}_4, \tag{17}$$

where

$$\lambda_{1:3} = \begin{pmatrix}\lambda_1\\ \lambda_2\\ \lambda_3\end{pmatrix} \text{ and } D_{\lambda_{1:3}} = \begin{pmatrix}\lambda_1 & 0 & 0\\ 0 & \lambda_2 & 0\\ 0 & 0 & \lambda_3\end{pmatrix}. \tag{18}$$

In the next section we will study under what conditions the inverses above exist and what happens when they don't.

We conclude this section by showing how to estimate H with four generally placed points. This can be achieved through a change of coordinates. Let $X_{1:3}$ be the 3×3 matrix with columns $\mathbf{x}_1$, $\mathbf{x}_2$ and $\mathbf{x}_3$. If $X_{1:3}$ is invertible then (9) can then be written

$$\lambda_i\mathbf{y}_i = H\mathbf{x}_i = \underbrace{HX_{1:3}}_{:=\tilde{H}}\underbrace{X_{1:3}^{-1}\mathbf{x}_i}_{:=\tilde{\mathbf{x}}_i}. \tag{19}$$

Since $X_{1:3}^{-1}X_{1:3} = I$ and $\mathbf{x}_i$, $i = 1, 2, 3$ are the columns of $X_{1:3}$ it is now easy to see that in the new coordinates $\tilde{\mathbf{x}}_i$ are of the desired form (11). We can therefore instead solve for $\tilde{H}$, as described above, and compute the solution to (9) using $H = \tilde{H}X_{1:3}^{-1}$. We will return to the issue of $X_{1:3}$ being invertible in Section 2.1.

2.1 Uniqueness and Degeneracy

In deriving the formulas for H we have made the assumptions that the inverses $D_{\mathbf{x}_4}^{-1}$, $Y_{1:3}^{-1}$ exist. In addition to make the coordinate change which ensures that (11) holds we need that $X_{1:3}^{-1}$ exists. To see when this holds we recall from Linear Algebra that a matrix is invertible if and only if its columns are linearly independent. In $\mathbb{R}^3$ the three vectors are linearly dependent if and only if they lie in a plane (containing the origin). That is, there is a vector l such that $l^T\mathbf{x}_i = 0$ for $i = 1, 2, 3$. In $\mathbb{P}^3$ the interpretation of these equations is that the points represented by $\mathbf{x}_i$ all lie on the same line l (see Lecture 2). Therefore the inverse $X_{1:3}^{-1}$ exists as long as the three points $\mathbf{x}_1$, $\mathbf{x}_2$, $\mathbf{x}_3$ are not collinear. Since a homography always maps lines to lines (see Lecture 2) $Y_{1:3}^{-1}$ exists precisely when $X_{1:3}^{-1}$ exist.

In addition we have to ensure that the inverse of $D_{\mathbf{x}_4}$ exists. It is clear that this is true when $a \neq 0$, $b \neq 0$ and $c \neq 0$. Now suppose that for example $c = 0$. Then it is clear that $\mathbf{x}_1$,$\mathbf{x}_2$ and $\mathbf{x}_4$ are linearly dependent (as vectors in $\mathbb{R}^3$). Similar arguments for a and b show that the inverses of $X_{1:3}$, $Y_{1:3}$ and $D_{\mathbf{x}_1}$ all exist if and only if no combination of three points from $\mathbf{x}_1$, $\mathbf{x}_2$, $\mathbf{x}_3$, $\mathbf{x}_4$ are collinear in $\mathbb{P}^2$.

In general a set of points in $\mathbb{P}^n$ are called **projectively independent** if they have homogeneous coordinates that are linearly independent as vectors in $\mathbb{R}^{n+1}$. (Note that since the homogeneous representatives of a point can only differ with a non-zero scaling it does not matter which representatives we choose to test projective independence.) A set of $n + 2$ points in $\mathbb{P}^n$ is called a **projective basis** if no subset of $n + 1$ points is projectively dependent. A projective transformation $\mathbb{P}^n \to \mathbb{P}^n$ is uniquely determined by the mapping of the $n + 2$ points of a projective basis. In the case of $n = 2$ we have already seen that we need four points where no three are on a line. As a second example consider a projective mapping $\mathbb{P}^3 \to \mathbb{P}^3$. If we know how five points are mapped and these form a projective basis we can uniquely determine the mapping. In this case the basis assumption amounts to no subset of four points lie on a plane in $\mathbb{P}^3$.

We now consider what happens in the case of $\mathbb{P}^2$ when there is a subset of 3 points that are on a line. For simplicity let us assume that the first 3 points are not on a line so that $X_{1:3}$ and $Y_{1:3}$ exist. In this case one of the constants a,b or c will be 0. If for example $a = 0$ then (15) reduces to

$$\lambda_4\mathbf{y}_4 = \lambda_2 b\mathbf{y}_2 + \lambda_3 c\mathbf{y}_3. \tag{20}$$

It is clear from this expression that λ_1 cannot be determined in this case. However the equations can still be solved since $a = 0$ corresponds to $\mathbf{x}_2$,$\mathbf{x}_3$,$\mathbf{x}_4$ (and therefore $\mathbf{y}_2$,$\mathbf{y}_3$,$\mathbf{y}_4$) being linearly dependent. From Linear Algebra we then know that there are numbers γ_2, γ_3 and γ_4 not all of them zero such that

$$\gamma_2\mathbf{y}_2 + \gamma_3\mathbf{y}_3 + \gamma_4\mathbf{y}_4 = 0. \tag{21}$$

Furthermore, we must have $\gamma_4 \neq 0$ since otherwise $\gamma_2 \mathbf{y}_2 = -\gamma_3 \mathbf{y}_3$ which means that $\mathbf{y}_2$ and $\mathbf{y}_3$ are homogeneous representatives of the same point in $\mathbb{P}^2$. Therefore selecting $\lambda_2 = -\frac{\gamma_2}{b}$, $\lambda_3 = -\frac{\gamma_3}{c}$, $\lambda_4 = \gamma_4$ and λ_1 arbitrarily gives a family of solutions, all mapping the 4 points correctly. Figure 6 shows an example of a degenerate case where three of the four black points are on a line. Both the estimated transformations map the black points to themselves but differ elsewhere. The first homography maps the blue grid to the green one while the second maps the blue grid to the red one.

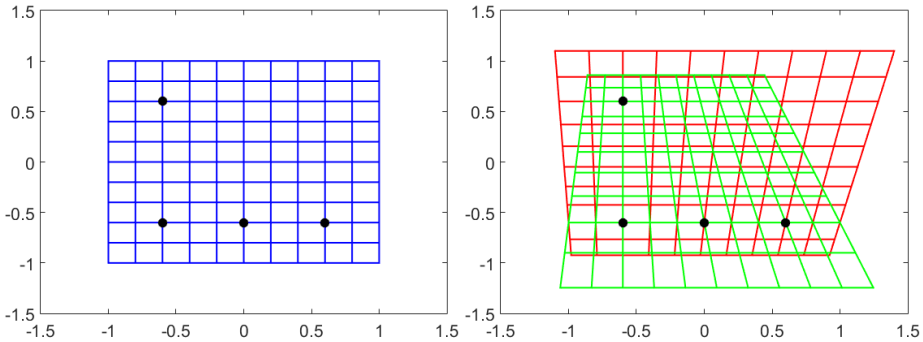

Figure 6: Two examples of homographies that map the four black points to themselves but differ for other points. The blue grid (left) is mapped to the green and red grids (right) by the two transformations respectively.

2.2 Noisy Recovery using DLT

As in the case of triangulation we often have overdetermined systems with noisy measurements. In such cases we can again apply the DLT approach. Suppose that we want to solve (9) with $n \geq 4$. Let h^i be row i of H, that is,

$$H = \begin{bmatrix} h^1 \\ h^2 \\ h^3 \end{bmatrix}, \tag{22}$$

and

$$\mathbf{y}_i = \begin{pmatrix} u_i \\ v_i \\ w_i \end{pmatrix}. \tag{23}$$

Here H is a 3×3 matrix so h^1, h^2 and h^3 are 1×3 matrices. By stacking all our unknowns in a vector

$$v^T = \begin{bmatrix} h_1 & h_2 & h_3 & \lambda_1 & \lambda_2 & \cdots & \lambda_n \end{bmatrix}. \tag{24}$$

we can write our problem as

$$Mv = 0, \tag{25}$$

where

$$M = \begin{bmatrix} \mathbf{x}_1^T & 0 & 0 & -u_1 & 0 & \cdots & 0 \\ 0 & \mathbf{x}_1^T & 0 & -v_1 & 0 & \cdots & 0 \\ 0 & 0 & \mathbf{x}_1^T & -w_1 & 0 & \cdots & 0 \\ \mathbf{x}_2^T & 0 & 0 & 0 & -u_2 & \cdots & 0 \\ 0 & \mathbf{x}_2^T & 0 & 0 & -v_2 & \cdots & 0 \\ 0 & 0 & \mathbf{x}_2^T & 0 & -w_2 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & & \vdots \\ \mathbf{x}_n^T & 0 & 0 & 0 & 0 & \cdots & -u_n \\ 0 & \mathbf{x}_n^T & 0 & 0 & 0 & \cdots & -v_n \\ 0 & 0 & \mathbf{x}_n^T & 0 & 0 & \cdots & -w_n \end{bmatrix}. \tag{26}$$

To account for noise we again solve the solve the homogeneous least squares problem

$$\min_{\|v\|^2=1} \|Mv\|^2 \tag{27}$$

using the singular value decomposition of M as described in Lecture 3.

2.3 Panoramic stitching

As an example of homography estimation we will show how we can stitch together a number of images taken from the same position (camera center). For ease of notation we will assume that we have calibrated cameras and that the image coordinates are normalized using the inverse of the calibration matrices. Since we have captured the images in the same point and we are free to choose a global coordinate system, we will assume that the camera center is at the origin. This means that for two corresponding image points $\mathbf{x}_i$ and $\mathbf{y}_i$ being projections of the 3D point $\mathbf{X}_i = \begin{bmatrix} X_i \\ 1 \end{bmatrix}$ we have the following system of equations

$$\gamma_i \mathbf{x}_i = [R_1 \ 0] \begin{bmatrix} X_i \\ 1 \end{bmatrix}, \tag{28}$$

$$\eta_i \mathbf{y}_i = [R_2 \ 0] \begin{bmatrix} X_i \\ 1 \end{bmatrix}, \tag{29}$$

where γ_i and η_i are unknown (non-zero) scalars. This gives us

$$\begin{cases} \gamma_i \mathbf{x}_i = R_1 X_i \\ \eta_i \mathbf{y}_i = R_2 X_i \end{cases} \Leftrightarrow \begin{cases} X_i = \gamma_i R_1^T \mathbf{x}_i \\ \eta_i \mathbf{y}_i = R_2 X_i \end{cases} \Leftrightarrow \begin{cases} X_i = \gamma_i R_1^T \mathbf{x}_i \\ \eta_i \mathbf{y}_i = \gamma_i R_2 R_1^T \mathbf{x}_i \end{cases}. \tag{30}$$

This means that we can write the last equation as

$$\lambda_i \mathbf{y}_i = H \mathbf{x}_i, \tag{31}$$

with $\lambda_i = \frac{\eta_i}{\gamma_i}$ and $H = R_2 R_1^T$. This shows that we can transfer points from the first image plane to the second by the use of a homography. We know from the previous sections that we can estimate this transformation from at least four point correspondences. Once we have estimated the homography, all other points in an image can be transferred using it.

The unnormalized points $\tilde{\mathbf{x}}_i$ and $\tilde{\mathbf{y}}_i$ are related to the normalized ones by $\mathbf{x}_i = K_1^{-1}\tilde{\mathbf{x}}_i$ and $\mathbf{y}_i = K_2^{-1}\tilde{\mathbf{y}}_i$ respectively. Therefore we have

$$\lambda_i K_2^{-1}\tilde{\mathbf{y}}_i = R_2 R_1^T K_1^{-1}\tilde{\mathbf{x}}_i \Leftrightarrow \lambda_i \tilde{\mathbf{y}}_i = K_2 R_2 R_1^T K_1^{-1}\tilde{\mathbf{x}}_i, \tag{32}$$

which means that the unnormalized points are mapped by a homography of the following form

$$H = K_2 R_2 R_1^T K_1^{-1}. \tag{33}$$

Figures 7-9 shows how we can use homographies to build panoramas from multiple images. In this case H_{21}, which transforms points in image 2 to Image 1, is estimated from green matches and H_{32}, which transforms points in image 3 to points in Image 2, is estimated from red matches. Figure 8 shows the effects of transforming Image 2 and 3 using the homographies H_{21} and $H_{31} = H_{21}H_{32}$. These transformations show where the pixels would have been placed had images 2 and 3 been captured with camera 1. When these transformations have been applied we can extend Image 1 with pixels from the other images.

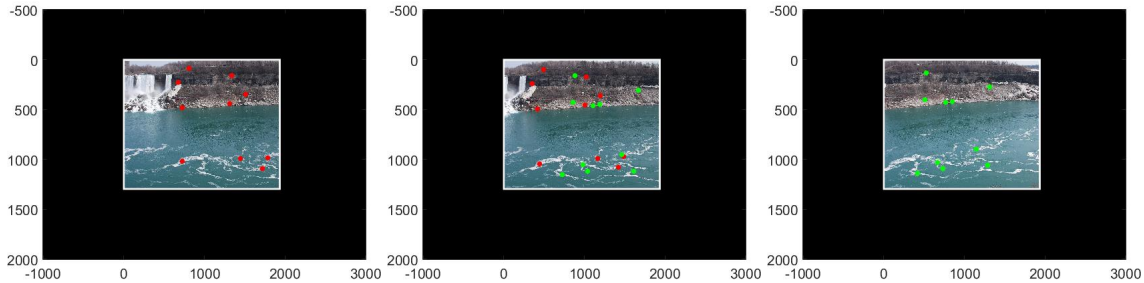

Figure 7: Original images with matches between image 1 and 2 (red) and between 2 and 3 (green). The matches can be used to compute two homographies, H_{21} which maps pixels in image 2 to image 1 and H_{32} which maps image 3 to image 2.

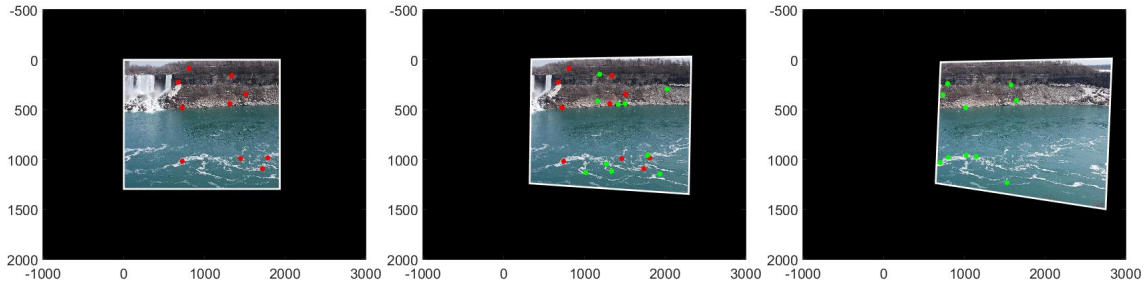

Figure 8: The homographies H_{21} and $H_{21}H_{32}$ can be used to transform the pixel of images 2 and 3 respectively, to the coordinate system of the first camera.

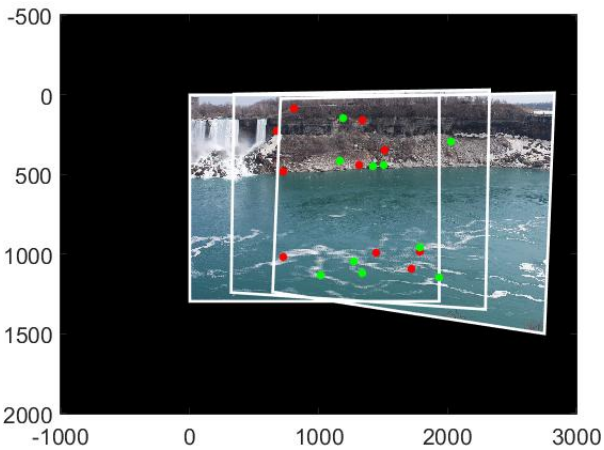

Figure 9: When images 2 and 3 has been transformed we can add the images together.