Panorama Stitching
Supplementary Notes

Per-Erik Forssén, June 2008

Computer Vision Laboratory, Linköping University, Sweden

Last update by Maria Magnusson: November 23, 2012

Abstract

This document describes equations and relations needed for the panorama stitching assignment in the course TSBB09 Image Sensors. We define the camera calibration matrix and rotational homographies, and solve the Orthogonal Procrustes problem. We also describe the axis-angle parametrisation, points on the unit sphere, and spherical coordinates.

1 Image Plane and Image Grid

The *image plane* is the 3D plane where the image sensor is located, see figure 1. By the *image grid* we instead mean the grid of image pixels grabbed by the sensor. Whereas the image plane has physical dimensions, the dimensions of the image grid are measured in numbers of pixels. The *camera coordinate system* (CCS) is a 3D coordinate system centred in a point known as the *optical centre*, see figure 1. In the ideal pin-hole camera, the optical centre is the pin hole, and for the ideal thin lens camera, the optical centre is the centre of the lens. Conversions between CCS coordinates and image grid coordinates (hereafter image coordinates) are performed using the *camera calibration matrix*.

The camera calibration matrix $\mathbf{K}$ is a 3×3 matrix which maps a point $\mathbf{X} \in \mathbb{R}^3$ in the CCS to image coordinates according to:

$$\tilde{\mathbf{x}} \sim \mathbf{K}\mathbf{X}\,. \tag{1}$$

Here $\tilde{\mathbf{x}} = (sx, sy, s)^T \in \mathbb{P}^2$ is the homogeneous representation of an image point $\mathbf{x} = (x, y)^T \in \mathbb{R}^2$.

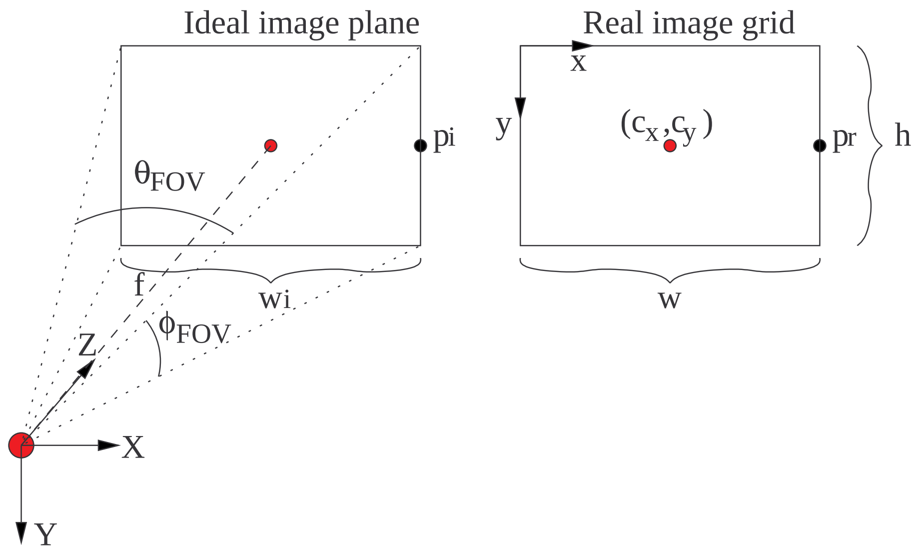

Figure 1: Left: Ideal image plane (where the sensor is located), the camera view-vector (or forward vector) is Z. Right: Real image grid (where the pixels live).

The camera calibration matrix is often parameterized as:

$$\mathbf{K} = \begin{pmatrix} K_{11} & K_{12} & K_{13} \\ K_{21} & K_{22} & K_{23} \\ K_{31} & K_{32} & K_{33} \end{pmatrix} = \begin{pmatrix} asf & \gamma & c_x \\ 0 & sf & c_y \\ 0 & 0 & 1 \end{pmatrix} . \tag{2}$$

Here f is the *focal length*, see figure 1, which is measured in e.g. mm, s is a scale factor that defines the relation between mm and pixels. Consequently, sf is the focal length measured in pixels. Moreover, γ is the *skew*, a is the *aspect ratio*, and $\mathbf{c} = (c_x, c_y)^T$ is the projection of the optical centre onto the image grid. Using the parametrization of $\mathbf{K}$ as in (2), (1) can be written:

$$\begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \sim \begin{pmatrix} xZ \\ yZ \\ Z \end{pmatrix} = \begin{pmatrix} K_{11} & K_{12} & c_x \\ 0 & K_{22} & c_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} . \tag{3}$$

The image width and height are denoted w and h, respectively. We can extract the offset of the optical centre from the geometrical centre in the image as

$$\text{horiz}_{\text{off}} = \tan^{-1}((w/2 - c_x)/K_{11}) \quad \text{and} \tag{4}$$

$$\text{vert}_{\text{off}} = \tan^{-1}((h/2 - c_y)/K_{22})\,. \tag{5}$$

We can also extract the horizontal and vertical fields of view as

$$\theta_{\text{FOV}} = 2\tan^{-1}(w/2/K_{11}) \quad \text{and} \tag{6}$$

$$\phi_{\text{FOV}} = 2\tan^{-1}(h/2/K_{22})\,. \tag{7}$$

Proof:
See figure 1. For simplicity, assume that $\text{horiz}_{\text{off}} = \text{vert}_{\text{off}} = 0$. Then the points $p_i = (w_i/2, 0, f)^T$ and $p_r = (c_x + w/2, c_y, 1)^T$. Therefore

$$Z\begin{pmatrix} c_x + w/2 \\ c_y \\ 1 \end{pmatrix} = \begin{pmatrix} K_{11} & K_{12} & c_x \\ 0 & K_{22} & c_y \\ 0 & 0 & 1 \end{pmatrix}\begin{pmatrix} w_i/2 \\ 0 \\ f \end{pmatrix}$$

The first row gives $Z(c_x + w/2) = K_{11}w_i/2 + c_x f$ and the third row gives $Z = f$. Therefore $f(c_x + w/2) = K_{11}w_i/2 + c_x f$, which gives $w_i = wf/K_{11}$.
Finally, $\theta_{\text{FOV}} = 2\arctan(w_i/(2f)) = 2\arctan(w/(2K_{11}))$.
Similarly, $\phi_{\text{FOV}} = 2\arctan(h/(2K_{22}))$.
Q.E.D.

2 Rotational Homographies

For two cameras that share the same optical centre, a 3D point, $\tilde{\mathbf{X}} \in \mathbb{P}^3$, is imaged as shown in figure 2. If we denote the two camera projection operators by $\mathbf{P}_1$ and $\mathbf{P}_2$, we get:

$$\tilde{\mathbf{x}}_1 \sim \mathbf{P}_1\tilde{\mathbf{X}} \quad \text{and} \quad \tilde{\mathbf{x}}_2 \sim \mathbf{P}_2\tilde{\mathbf{X}}\,. \tag{8}$$

In general, a camera projection operator is written as:

$$\mathbf{P} = \mathbf{K}\,[\mathbf{R}\;\mathbf{t}]\;, \tag{9}$$

where $\mathbf{K}$ is the camera calibration matrix, and $\mathbf{R}$, $\mathbf{t}$ specifies the location of the optical centre of the camera in some world coordinate system (WCS).

Since the two cameras in figure 2 share the same optical centre, we can simplify calculations by choosing the WCS origin as the optical centre. This gives us:

$$\tilde{\mathbf{x}}_1 \sim \mathbf{P}_1\tilde{\mathbf{X}} = \mathbf{K}\,[\mathbf{R}_1\;\mathbf{0}]\,\tilde{\mathbf{X}} = \mathbf{K}\mathbf{R}_1\mathbf{X} \quad \text{and} \quad \tilde{\mathbf{x}}_2 \sim \mathbf{K}\mathbf{R}_2\mathbf{X}\,. \tag{10}$$

Now, assume the existence of a homography $\mathbf{H}_{21}$ that maps points from image 2 to image 1 according to:

$$\tilde{\mathbf{x}}_1 \sim \mathbf{H}_{21}\tilde{\mathbf{x}}_2\,. \tag{11}$$

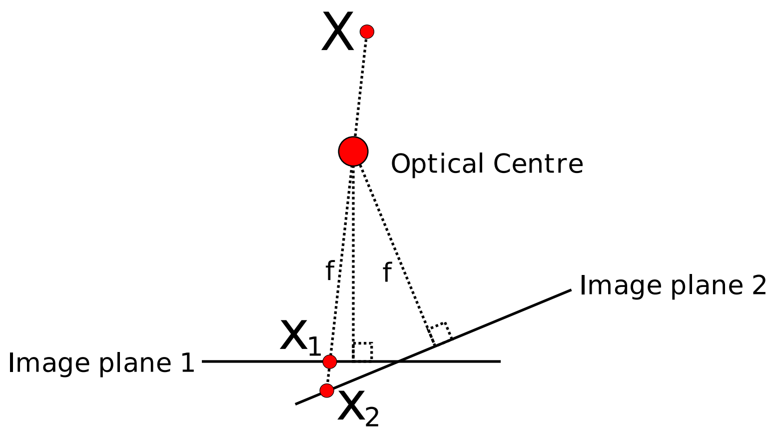

Figure 2: Rotation geometry.

If we insert the expressions from (10) we get

$$\mathbf{K}\mathbf{R}_1\mathbf{X} \sim \mathbf{H}_{21}\mathbf{K}\mathbf{R}_2\mathbf{X}\,, \tag{12}$$

which is satisfied when

$$\mathbf{H}_{21} = \mathbf{K}\mathbf{R}_1\mathbf{R}_2^T\mathbf{K}^{-1}\,. \tag{13}$$

Since the choice of the point $\mathbf{X}$ was arbitrary, we have now proven the existence of the homography $\mathbf{H}_{21}$. A homography induced by a pure rotation about the optical centre is known as a *rotational homography*, and the form derived in (13) will be needed in the panorama stitching assignment.

Finally, in order to clarify any misunderstandings, we write equation (11) in an expanded version,

$$\begin{pmatrix} x_1 \\ y_1 \\ 1 \end{pmatrix} \sim \begin{pmatrix} x_1 s \\ y_1 s \\ s \end{pmatrix} = \begin{pmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{pmatrix} \begin{pmatrix} x_2 \\ y_2 \\ 1 \end{pmatrix}\,, \tag{14}$$

where $\tilde{\mathbf{x}}_1 = (x_1, y_1, 1)$, $\tilde{\mathbf{x}}_2 = (x_2, y_2, 1)$ and

$$\mathbf{H}_{21} = \begin{pmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{pmatrix}\,. \tag{15}$$

3 The Orthogonal Procrustes problem

Consider two sets of 3D points $\mathbf{X} = \begin{pmatrix} \mathbf{X}_1 & \dots & \mathbf{X}_N \end{pmatrix}$ and $\mathbf{Y} = \begin{pmatrix} \mathbf{Y}_1 & \dots & \mathbf{Y}_N \end{pmatrix}$. The points are related with an orthonormal matrix $\mathbf{R}$, according to

$$\mathbf{X}_k = \mathbf{R}\mathbf{Y}_k + \epsilon_k\,, \tag{16}$$

where ϵ_k is an additive Gaussian noise term.

Finding $\mathbf{R}$ is called the *Orthogonal Procrustes problem* [1], and is normally formulated as:

$$\arg\min_{\mathbf{R}} ||\mathbf{X} - \mathbf{R}\mathbf{Y}||^2 \text{ subject to } \mathbf{R}\mathbf{R}^T = \mathbf{I}\,. \tag{17}$$

The solution to (17) is found using the singular value decomposition (SVD), $\mathbf{U}\mathbf{D}\mathbf{V}^T$, of $\mathbf{X}\mathbf{Y}^T$. The singular value decomposition gives that $\mathbf{U}$ and $\mathbf{V}$ are orthonormal matrices and $\mathbf{D}$ is a diagonal matrix. The matrix $\mathbf{R}$ that minimises (17) can now be found as $\mathbf{R} = \mathbf{U}\mathbf{V}^T$.

Proof: The trace (tr) of a matrix is the sum of its diagonal elements. We need to make use of the fact that $\text{tr}(\mathbf{A}\mathbf{B}) = \text{tr}(\mathbf{B}\mathbf{A})$, and $\text{tr}(\mathbf{A}) = \text{tr}(\mathbf{A}^T)$, when $\mathbf{A}$ is a square matrix. Now we can rewrite the term to minimise in (17) as:

$$\min_{\mathbf{R}} \text{tr}\left[(\mathbf{X} - \mathbf{R}\mathbf{Y})(\mathbf{X} - \mathbf{R}\mathbf{Y})^T\right] = \tag{18}$$

$$\min_{\mathbf{R}} \text{tr}(\mathbf{X}\mathbf{X}^T) - \text{tr}(\mathbf{Y}\mathbf{Y}^T) - 2\text{tr}(\mathbf{R}\mathbf{Y}\mathbf{X}^T)\,. \tag{19}$$

As the first two terms of (19) do not contain $\mathbf{R}$, this is equivalent to:

$$\max_{\mathbf{R}} \text{tr}(\mathbf{R}\mathbf{Y}\mathbf{X}^T)\,. \tag{20}$$

We now define the SVD of $\mathbf{X}\mathbf{Y}^T$ to be $\mathbf{X}\mathbf{Y}^T = \mathbf{U}\mathbf{D}\mathbf{V}^T$. Consequently $\mathbf{Y}\mathbf{X}^T = \mathbf{V}\mathbf{D}\mathbf{U}^T$ and we get:

$$\max_{\mathbf{R}} \text{tr}(\mathbf{R}\mathbf{V}\mathbf{D}\mathbf{U}^T) = \max_{\mathbf{R}} \text{tr}(\mathbf{U}^T\mathbf{R}\mathbf{V}\mathbf{D})\,. \tag{21}$$

As $\mathbf{R}$, $\mathbf{U}$, and $\mathbf{V}$ all are orthonormal matrices, the maximum trace is $\text{tr}(\mathbf{D})$, and is obtained for $\mathbf{R} = \mathbf{U}\mathbf{V}^T$, which is an orthonormal matrix. □

4 Axis-Angle Representation

An important result in 3D geometry is that any combination of 3D rotations may be represented as a single 3D rotation about a rotation axis $\hat{\mathbf{n}}$, with an angle $\alpha \in [0, \pi[$, see figure 3.

Note that any 3D point along $\hat{\mathbf{n}}$ will be unaffected by such a rotation. In other words, for a rotation matrix $\mathbf{R}$ there exists a vector $\hat{\mathbf{n}}$ such that

$$\hat{\mathbf{n}} = \mathbf{R}\hat{\mathbf{n}}\,. \tag{22}$$

We now observe that (22) is an eigenvalue equation, and thus $\hat{\mathbf{n}}$ is an eigenvector of $\mathbf{R}$ with eigenvalue 1. The other two eigenvectors, by definition

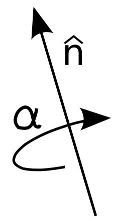

Figure 3: Illustration of axis-angle rotation parametrisation.

have to be orthogonal to $\hat{\mathbf{n}}$. It can be shown (using Rodrigues' rotation formula, cross-product matrices, Euler's formula, and matrix exponentials) that their eigenvalues are $e^{\mathbf{i}\alpha}$, and $e^{-\mathbf{i}\alpha}$ respectively. This is an important result, as it allows us to determine the rotation angle, and the rotation axis for any rotation matrix.

5 Points on the Unit Sphere

In (1) we saw that applying the camera calibration matrix to a 3D point in the CCS gave us image coordinates. Since $\mathbf{K}$ is a square matrix, we can also consider the inverse mapping:

$$\tilde{\mathbf{p}} \sim \mathbf{K}^{-1}\tilde{\mathbf{x}}\,. \tag{23}$$

The obtained vector $\mathbf{p}$ is a homogeneous quantity, and by a proper scaling it can be made equal to any of the points on the 3D line passing through the optical centre of the camera, and the actual 3D point $\mathbf{X}$ that generated the image point $\mathbf{x}$.

By choosing to normalise $\tilde{\mathbf{p}}$ to have an Euclidean norm of 1, i.e. $\mathbf{p} = \tilde{\mathbf{p}}/||\tilde{\mathbf{p}}||$, we can interpret it as a point on the unit sphere. This interpretation is important for the purpose of panorama generation, as we can then use the Procrustes algorithm to find rotations between two sets of points $\mathbf{p}$.

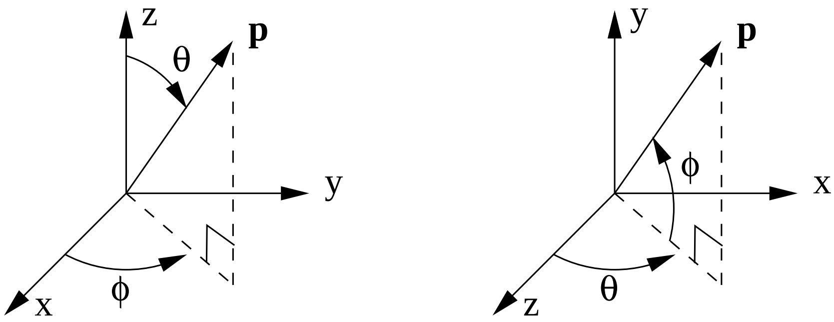

Figure 4: Illustration of spherical coordinate systems. Left: Standard form (24). Right: Longitude-latitude form (25).

6 Spherical Coordinates

Consider a point in 3D with the Cartesian coordinates $\mathbf{p} = \begin{pmatrix} x & y & z \end{pmatrix}^T$. If the point lies on a sphere, it satisfies the constraint $r^2 = x^2 + y^2 + z^2$ (where r is the radius of the sphere) and can be expressed in spherical coordinates.

By spherical coordinates one normally means the coordinate triplet (r, ϕ, θ), which expresses $\mathbf{p}$ as:

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} r\sin\theta\cos\phi \\ r\sin\theta\sin\phi \\ r\cos\theta \end{pmatrix} . \tag{24}$$

See figure 4, left. Note that when $\theta \in \{0, \pi\}$, the value of ϕ is arbitrary. This means that the mapping has a singularity at these two points. For the purpose of image resampling, it is thus convenient to instead use the longitude-latitude form of spherical coordinates, see figure 4, right. Here the (ϕ, θ, r) triplet expresses $\mathbf{p}$ as:

$$\mathbf{p} = \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} r\cos\phi\sin\theta \\ r\sin\phi \\ r\cos\phi\cos\theta \end{pmatrix} \tag{25}$$

Here ϕ is the latitude coordinate, and θ is the longitude coordinate. Note that this parametrisation instead places the two inevitable singularities at the north and south poles (i.e. directly above and below the camera). The longitude-latitude form of spherical coordinates is called so, as it defines a map projection sometimes used to represent the surface of the Earth on a flat paper, see figure 5.

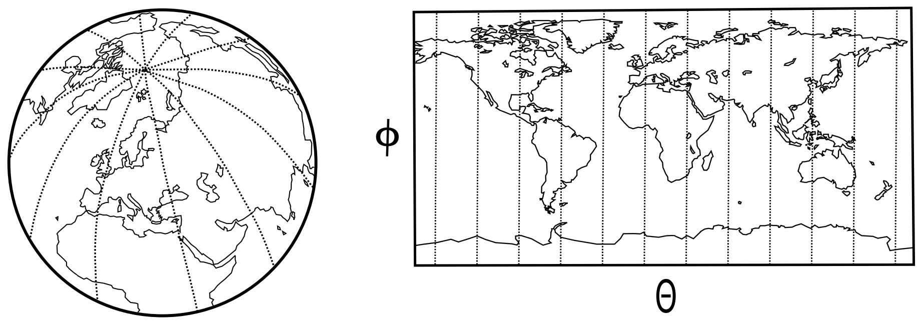

Figure 5: Illustration of spherical coordinates. Left: A world map painted on a sphere. Right: The same map in longitude-latitude space.

7 Image Resampling to Spherical Coordinates

In order to obtain pixel values in the spherical coordinate system, we first generate coordinates in a regular grid in (ϕ, θ) space, cf. figure 5, right.

For the purpose of panorama stitching, it is then convenient to transform these to Cartesian 3D coordinates using (25). This allows us to change coordinates according to the rotation between views, using matrix multiplication:

$$\mathbf{p}' = \mathbf{R}\mathbf{p}\,, \tag{26}$$

where $\mathbf{R}$ is related to the image-to-image homography according to (13). Finally, we can now project the points into the image, using the camera calibration matrix $\mathbf{K}$. To summarise, the full transformation from spherical coordinates $\mathbf{p}(\phi, \theta)$ to image coordinates $\mathbf{x} = (x, y)^T$ (where $\tilde{\mathbf{x}} = (sx, sy, s)^T$) is given by:

$$\tilde{\mathbf{x}} \sim \mathbf{K}\mathbf{R}\mathbf{p}(\phi, \theta)\,, \tag{27}$$

Note that we have omitted the r coordinate from our parametrisation of $\mathbf{p}$. Since $\tilde{\mathbf{x}}$ is obtained in homogeneous form, and r appears as a linear scaling of $\mathbf{p}$, the choice of r is arbitrary, and can be set to e.g. 1.

References

[1] G. H. Golub and C. F. Van Loan. *Matrix Computations*. Johns Hopkins University Press, Baltimore, Maryland, 1983.