Reconstruction - II

September 6, 2017

1 Brief recap

In the last lecture, we talked about the essential matrix. To recap, we found that given two images taken by a pair of cameras for which we know the intrinsic parameters $\mathbf{K}$, we can derive a constraint on pairs of corresponding pixels. First, let's write down the projection equations:

$$\vec{\mathbf{p}}_1 \sim \mathbf{K}_1[\mathbf{R}_1|\mathbf{t}_1]\vec{\mathbf{P}} \tag{1}$$

$$\vec{\mathbf{p}}_2 \sim \mathbf{K}_2[\mathbf{R}_2|\mathbf{t}_2]\vec{\mathbf{P}} \tag{2}$$

Now, if we know $\mathbf{K}_1$ and $\mathbf{K}_2$, we can simply set $\vec{\mathbf{p}}_1 \leftarrow \mathbf{K}_1^{-1}\vec{\mathbf{p}}_1$ and $\vec{\mathbf{p}}_2 \leftarrow \mathbf{K}_2^{-1}\vec{\mathbf{p}}_2$ and eliminate it out of the equation.

Since we don't have a coordinate system a priori, we can use the first camera's coordinate system as the coordinate system of choice. This leads to the equations

$$\vec{\mathbf{p}}_1 \sim [\mathbf{I}|\mathbf{0}]\vec{\mathbf{P}} \tag{3}$$

$$\vec{\mathbf{p}}_2 \sim [\mathbf{R}|\mathbf{t}]\vec{\mathbf{P}} \tag{4}$$

Next, we write $\vec{\mathbf{P}}$ in terms of the non-homogenous coordinates $\mathbf{P}$ as $\begin{pmatrix} \mathbf{P} \\ 1 \end{pmatrix}$, and replace equivalence with equality by adding a free scale parameter:

$$\lambda_1\vec{\mathbf{p}}_1 = \mathbf{P} \tag{5}$$

$$\lambda_2\vec{\mathbf{p}}_2 = \mathbf{R}\mathbf{P} + \mathbf{t} \tag{6}$$

Substituting the first equation into the second, we get:

$$\lambda_2\vec{\mathbf{p}}_2 = \lambda_1\mathbf{R}\vec{\mathbf{p}}_1 + \mathbf{t} \tag{7}$$

Taking a cross product with $\mathbf{t}$ and then taking a dot product with $\vec{\mathbf{p}}_2$ gives:

$$\lambda_2\vec{\mathbf{p}}_2 \cdot (\mathbf{t} \times \vec{\mathbf{p}}_2) = \lambda_1\vec{\mathbf{p}}_2 \cdot (\mathbf{t} \times \mathbf{R}\vec{\mathbf{p}}_1) + \vec{\mathbf{p}}_2 \cdot (\mathbf{t} \times \mathbf{t}) \tag{8}$$

The LHS and the last term on the RHS are 0, so we have:

$$\vec{\mathbf{p}}_2 \cdot (\mathbf{t} \times \mathbf{R}\vec{\mathbf{p}}_1) = 0 \tag{9}$$

$$\Rightarrow \vec{\mathbf{p}}_2^T \mathbf{t}_\times \mathbf{R}\vec{\mathbf{p}}_1 = 0 \tag{10}$$

$$\Rightarrow \vec{\mathbf{p}}_2^T \mathbf{E}\vec{\mathbf{p}}_1 = 0 \tag{11}$$

where the *Essential matrix* $\mathbf{E} = \mathbf{t}_\times \mathbf{R}$. We saw last time that from 8 correspondences, we can estimate $\mathbf{E}$ and thus both $\mathbf{t}_\times$ and $\mathbf{R}$. Thus, from 8 correspondences, we can calibrate both cameras w.r.t each other.

However, this constraint, called the epipolar constraint, is also a constraint on the correspondences between the two images. Recall that in the last lecture, we saw that to reconstruct the 3D location of a pixel, we need the location of the corresponding pixel in another image. The epipolar constraint tells us that this corresponding pixel cannot be located in an arbitrary location in the other image.

2 Consequences of the epipolar constraint

2.1 The corresponding pixel lies on a line

The epipolar constraint is that for corresponding pixels $\vec{\mathbf{p}}_1$ and $\vec{\mathbf{p}}_2$, $\vec{\mathbf{p}}_2^T \mathbf{E}\vec{\mathbf{p}}_1 = 0$. If we fix $\vec{\mathbf{p_2}}$, then we end up with an equation in $\vec{\mathbf{p}}_1$:

$$\mathbf{l}^T \vec{\mathbf{p}}_1 = 0 \tag{12}$$

where $\mathbf{l} = \mathbf{E}^T \vec{\mathbf{p}}_2$. If $\mathbf{l} = (l_x, l_y, l_z)^T$, and $\vec{\mathbf{p}}_1 = (x, y, 1)^T$, then this equation is $l_x x + l_y y + l_z = 0$, i.e., the equation of a line. This line is called the *epipolar line* in the first image corresponding to point $\vec{\mathbf{p}}_2$ in the second image.

Similarly, if we fix $\vec{\mathbf{p}}_1$, then we get an epipolar line in the second image:

$$\vec{\mathbf{p}}_1^T \mathbf{l} = 0 \tag{13}$$

where $\mathbf{l} = \mathbf{E}\vec{\mathbf{p}}_1$.

2.2 All epipolar lines intersect at a point

What are epipolar lines physically? As we have seen previously, pixel $\vec{\mathbf{p}}_1$ in the first image corresponds to a ray in the 3D world. This ray passes through the pinhole of the first camera, and the image pixel in question. When photographed by the second camera, this ray appears as a line. This line is the epipolar line: the true 3D point corresponding to this image pixel must lie somewhere along this ray, and so its image in the second camera must lie somewhere along this line.

Clearly, the rays corresponding to different pixels $\vec{\mathbf{p}}_i$ in the first image all pass through the first camera's pinhole. This means that the corresponding epipolar lines in the second image must pass through the *image* of the first

camera's pinhole in the second camera. Similarly, epipolar lines in the first image must pass through the *image* of the second camera's pinhole in the first camera. Let's verify this mathematically.

Let us first look at the image of the first camera's pinhole in the second camera. The first camera's pinhole is at the origin of our chosen coordinate system. The image of this in the second camera is therefore:

$$\lambda_2 \vec{\mathbf{c}}_2 = \mathbf{R} \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} + \mathbf{t} = \mathbf{t} \tag{14}$$

$$\Rightarrow \vec{\mathbf{c}}_2 \sim \mathbf{t} \tag{15}$$

Now consider any point $\vec{\mathbf{p}}_1$ in the first image. The corresponding epipolar line is given by $\mathbf{l} = \mathbf{E}\vec{\mathbf{p_1}}$. Then,

$$\vec{\mathbf{c}}_2^T \mathbf{l} \sim \vec{\mathbf{c}}_2^T \mathbf{E} \vec{\mathbf{p}}_1 \tag{16}$$

$$\sim \vec{\mathbf{c}}_2^T \mathbf{t}_\times \mathbf{R} \vec{\mathbf{p}}_1 \tag{17}$$

$$\sim \mathbf{t}^T \mathbf{t}_\times \mathbf{R} \vec{\mathbf{p}}_1 \tag{18}$$

$$\sim \mathbf{t} \cdot (\mathbf{t} \times \mathbf{R} \vec{\mathbf{p}}_1) \tag{19}$$

$$= 0 \tag{20}$$

Thus $\vec{\mathbf{c}}_2$ lies on $\mathbf{l}$ for all epipolar lines $\mathbf{l}$ in the second image.

Next let us look at the image of the second camera's pinhole in the first camera. To do that we need the location of the second camera's pinhole. In the second camera's coordinate system, this is at $\begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$. But we know that the second camera is related to the first camera by rotation $\mathbf{R}$ and translation $\mathbf{t}$. So we have that:

$$\begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} = \mathbf{R}\mathbf{C} + \mathbf{t} \tag{21}$$

$$\Rightarrow \mathbf{C} = -\mathbf{R}^T \mathbf{t} \tag{22}$$

The image location of this is given by $\vec{\mathbf{c}}_1 \sim \mathbf{C} = -\mathbf{R}^T \mathbf{t}$. Again, for any point $\vec{\mathbf{p}}_2$ in the second image, if $\mathbf{l} = \mathbf{E}^T \vec{\mathbf{p}}_2$ is the epipolar line, then:

$$\mathbf{l}^T \vec{\mathbf{c}}_1 = \vec{\mathbf{p}}_2 \mathbf{t}_\times \mathbf{R}(-\mathbf{R}^T \mathbf{t}) = -\vec{\mathbf{p}}_2 \mathbf{t}_\times \mathbf{t} = 0 \tag{23}$$

Thus all epipolar lines in the first image pass through $\mathbf{c}_1$, which is the epipole in the first image.

3 Special case 1: Pure translation along X

Consider the case when the two cameras are pointing in the same direction and are separated only along the X axis. In this case $\mathbf{R} = \mathbf{I}$, and $\mathbf{t} = \begin{pmatrix} t_x \\ 0 \\ 0 \end{pmatrix}$. Then the epipole $\vec{\mathbf{c}}_2 = \mathbf{t}$. Since the z coordinate is 0, this point is *at infinity* along the X axis. Thus all epipolar lines in the second image are horizontal and parallel to each other. Similarly, epipolar lines in the first image are horizontal and parallel to each other.

We can also say more about where points project in the two images. Consider the projections of a point $\mathbf{P} = (X, Y, Z)$. For the first camera:

$$\vec{\mathbf{p}}_1 \sim \mathbf{P} = \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} \tag{24}$$

$$\Rightarrow \mathbf{p}_1 = \begin{pmatrix} \frac{X}{Z} \\ \frac{Y}{Z} \end{pmatrix} \tag{25}$$

$$\tag{26}$$

where the second equation converts into non-homogenous coordinates. Similarly, for the second camera:

$$\vec{\mathbf{p}}_2 \sim \mathbf{P} + \mathbf{t} = \begin{pmatrix} X + t_x \\ Y \\ Z \end{pmatrix} \tag{27}$$

$$\Rightarrow \mathbf{p}_2 = \begin{pmatrix} \frac{X+t_x}{Z} \\ \frac{Y}{Z} \end{pmatrix} \tag{28}$$

$$\tag{29}$$

Thus, the two images have the same y coordinate, but the x coordinate differs by $\frac{t_x}{Z}$. This means two things:

1. Given a pixel in one image, to find its correspondence in the other image, we simply need to search *along the same row*. This makes searching for correspondence exceedingly simple.

2. Once the corresponding pixel in the other image is found, reconstructing the depth of this point is also very simple, amounting to simply inverting the difference in the X coordinate of the two pixels.

The difference between x coordinates is called the *disparity*, and clearly, it is *inversely* proportional to depth. Given the *baseline* t_x, knowing the correspondence between pixels gives their disparity and thus their depth. In this case, we can create a *disparity image* in which each pixel has as its value the corresponding disparity.

Because of the simplicity of the calculations in this case, many stereo rigs will use this setup of cameras pointing in the same direction but translated perpendicular to the viewing direction (eg. Kinect). Our eyes use a very similar setup.

Note that, denoting disparity by d,

$$d = \frac{t_x}{Z} \tag{30}$$

$$\Rightarrow \Delta d = -\frac{t_x}{Z^2}\Delta Z \tag{31}$$

$$\Rightarrow \Delta Z \propto Z^2 \Delta d \tag{32}$$

This means that the error in depth estimation increases as the *square* of the depth. Because of this, stereo-based depth estimation is most accurate for points close to the camera.

4 Special case 2: Pure rotation

Consider the case when the two cameras are at the same location, but are simply rotated. Thus $\mathbf{t} = 0$. Thus:

$$\lambda_2 \vec{\mathbf{p}}^2 = \mathbf{P}^2 = \mathbf{R}\mathbf{P}^1 = \lambda_1 \mathbf{R}\vec{\mathbf{p}}^1 \tag{33}$$

In this case, the two images are related by a fixed linear transformation of the homogenous coordinates. This has two important implications. First, in this case, the two images offer no additional information to locate the points in 3D. Second, it is possible to rotate a camera in place "virtually" without requiring any knowledge of the 3D structure.

4.1 Rectifying images

As we saw above, if two cameras are related by a pure rotation, then their corresponding images are related by a simple linear transformation of the coordinates. This means that we can produce one image from the other simply by copying the pixel at the appropriate location, without knowing what 3D point they correspond to.

Now consider the general case where two cameras are related to each other through an arbitrary rotation and translation. Suppose we know the rotation and translation, for example, from the essential matrix. Then, we can rotate both cameras in place so that they use the same coordinate system, but are just translated w.r.t each other. Concretely, define a new coordinate system with the new X axis along the translation vector between the two cameras, and with Y and Z axes being perpendicular. We can then rotate both cameras in place till their coordinate system matches this new set of axes; they are now simply translated along X. This produces the simple setup of two cameras parallel to each other translated along the X axis. This is usually the first step of many stereo algorithms and is called "stereo rectification".

5 Special case 3: Points on a plane

Suppose we have a point $\mathbf{P}$ on a plane given by equation $\mathbf{N}^T\mathbf{X} = d$. The corresponding image is $\vec{\mathbf{p}}_1$ in image 1. Then:

$$\lambda_1 \vec{\mathbf{p}}_1 = \mathbf{P} \tag{34}$$

Putting this into the equation of the plane:

$$\begin{aligned} \lambda_1 \mathbf{N}^T \vec{\mathbf{p}}_1 &= d \\ \Rightarrow \lambda_1 &= \frac{d}{\mathbf{N}^T \vec{\mathbf{p}}_1} \end{aligned} \tag{35}$$

We can now look at this point in the coordinate system in camera 2:

$$\begin{aligned} \lambda_2 \vec{\mathbf{p}}_2 &= \mathbf{R}\mathbf{P} + \mathbf{t} \\ &= \tfrac{d}{\mathbf{N}^T \vec{\mathbf{p}}_1} \mathbf{R}\vec{\mathbf{p}}_1 + \mathbf{t} && (36) \\ &= \tfrac{d}{\mathbf{N}^T \vec{\mathbf{p}}_1} (\mathbf{R}\vec{\mathbf{p}}_1 + \tfrac{\mathbf{t}}{d}\mathbf{N}^T \vec{\mathbf{p}}_1) && (37) \\ &= \tfrac{d}{\mathbf{N}^T \vec{\mathbf{p}}_1} (\mathbf{R} + \tfrac{\mathbf{t}}{d}\mathbf{N}^T) \vec{\mathbf{p}}_1 && (38) \\ &= \lambda \mathbf{H} \vec{\mathbf{p}}_1 && (39) \\ \Rightarrow \vec{\mathbf{p}}_2 &\sim \mathbf{H}\vec{\mathbf{p}}_1 && (40) \end{aligned}$$

where $\mathbf{H} = (\mathbf{R} + \frac{\mathbf{t}}{d}\mathbf{N}^T)$ in homogenous coordinates.

Thus, for points on a plane, there is a direct linear mapping from one image to another. Note that even though the camera motion is general, in this case we obtain a direct mapping because we know something about the world. This is much stronger than the epipolar constraint.

Given 4 corresponding points in the two images, we can solve for $\mathbf{H}$ (it has 9 parameters, but one extra degree of freedom). This is unlike the essential matrix, which requires 8 correspondences.

What happens if in estimating the essential matrix we take 8 correspondences from the same plane? Consider the matrix $\mathbf{E}(\mathbf{u}) = \mathbf{u}_\times \mathbf{H}$ for some arbitrary vector $\mathbf{u}$. Then, for corresponding points $\vec{\mathbf{p}}_1$ and $\vec{\mathbf{p}}_2$ from this plane:

$$\vec{\mathbf{p}}_2^T \mathbf{E}(\mathbf{u}) \vec{\mathbf{p}}_1 \sim \vec{\mathbf{p}}_2^T \mathbf{u}_\times \mathbf{H} \vec{\mathbf{p}}_1 \sim \vec{\mathbf{p}}_2^T \mathbf{u}_\times \vec{\mathbf{p}}_2 = 0 \tag{41}$$

Thus there is an entire family of matrices that satisfy the epipolar constraint. Therefore, if we are just given correspondences from the same plane, we cannot estimate the essential matrix.

Note that $\mathbf{H}$ contains the camera translation $\mathbf{t}$ and the distance of the plane from the origin / camera 1, d, only appear as a ratio. Thus, two pictures of a nearby plane taken close to each other are mathematically equivalent to two pictures of a far away plane taken far apart.

$\mathbf{H}$ is called a planar homography.

6 The uncalibrated case

Till now we have assumed that we know $\mathbf{K}_1$ and $\mathbf{K}_2$. What if we don't?

In this case, we can again produce an epipolar constraint. First, let's write down the projection equations:

$$\vec{\mathbf{p}}_1 \sim \mathbf{K}_1[\mathbf{R}_1|\mathbf{t}_1]\vec{\mathbf{P}} \tag{42}$$

$$\vec{\mathbf{p}}_2 \sim \mathbf{K}_2[\mathbf{R}_2|\mathbf{t}_2]\vec{\mathbf{P}} \tag{43}$$

We will first invert the transformations $\mathbf{K}_1$ and $\mathbf{K}_2$.

$$\mathbf{K}_1^{-1}\vec{\mathbf{p}}_1 \sim \mathbf{K}_1[\mathbf{R}_1|\mathbf{t}_1]\vec{\mathbf{P}} \tag{44}$$

$$\mathbf{K}_2^{-1}\vec{\mathbf{p}}_2 \sim \mathbf{K}_2[\mathbf{R}_2|\mathbf{t}_2]\vec{\mathbf{P}} \tag{45}$$

Since we don't have a coordinate system a priori, we can use the first camera's coordinate system as the coordinate system of choice. This leads to the equations

$$\mathbf{K}_1^{-1}\vec{\mathbf{p}}_1 \sim [\mathbf{I}|\mathbf{0}]\vec{\mathbf{P}} \tag{46}$$

$$\mathbf{K}_2^{-1}\vec{\mathbf{p}}_2 \sim [\mathbf{R}|\mathbf{t}]\vec{\mathbf{P}} \tag{47}$$

Next, we write $\vec{\mathbf{P}}$ in terms of the non-homogenous coordinates $\mathbf{P}$ as $\begin{pmatrix} \mathbf{P} \\ 1 \end{pmatrix}$, and replace equivalence with equality by adding a free scale parameter:

$$\lambda_1\mathbf{K}_1^{-1}\vec{\mathbf{p}}_1 = \mathbf{P} \tag{48}$$

$$\lambda_2\mathbf{K}_2^{-1}\vec{\mathbf{p}}_2 = \mathbf{R}\mathbf{P} + \mathbf{t} \tag{49}$$

Substituting the first equation into the second, we get:

$$\lambda_2\mathbf{K}_2^{-1}\vec{\mathbf{p}}_2 = \lambda_1\mathbf{R}\mathbf{K}_1^{-1}\vec{\mathbf{p}}_1 + \mathbf{t} \tag{50}$$

Taking a cross product with $\mathbf{t}$ and then taking a dot product with $\mathbf{K}_2^{-1}\vec{\mathbf{p}}_2$ gives:

$$\lambda_2(\mathbf{K}_2^{-1}\vec{\mathbf{p}}_2)\cdot(\mathbf{t}\times\mathbf{K}_2^{-1}\vec{\mathbf{p}}_2) = \lambda_1(\mathbf{K}_2^{-1}\vec{\mathbf{p}}_2)\cdot(\mathbf{t}\times\mathbf{R}\mathbf{K}_1^{-1}\vec{\mathbf{p}}_1) + (\mathbf{K}_2^{-1}\vec{\mathbf{p}}_2)\cdot(\mathbf{t}\times\mathbf{t}) \tag{51}$$

The LHS and the last term on the RHS are 0, so we have:

$$(\mathbf{K}_2^{-1}\vec{\mathbf{p}}_2)\cdot(\mathbf{t}\times\mathbf{R}\mathbf{K}_1^{-1}\vec{\mathbf{p}}_1) = 0 \tag{52}$$

$$\Rightarrow \vec{\mathbf{p}}_2^T\mathbf{K}_2^{-T}\mathbf{t}_\times\mathbf{R}\mathbf{K}_1^{-1}\vec{\mathbf{p}}_1 = 0 \tag{53}$$

$$\Rightarrow \vec{\mathbf{p}}_2^T\mathbf{F}\vec{\mathbf{p}}_1 = 0 \tag{54}$$

where the *Fundamental matrix* $\mathbf{F} = \mathbf{K}_2^{-T}\mathbf{t}_\times\mathbf{R}\mathbf{K}_1^{-1} = \mathbf{K}_2^{-T}\mathbf{E}\mathbf{K}_1^{-1}$.

Thus, there is still an epipolar constraint constraining correspondences. However, now, if we know the fundamental matrix, we can no longer break it down into $\mathbf{R}$ and $\mathbf{t}$, because $\mathbf{K}_1$ and $\mathbf{K}_2$ are in general arbitrary matrices.