Lecture 4: Autocorrelation, Triangulation and Homography Estimation

1 The Autocorrelation function

If we want to investigate how the structure of an image I varies we can look at the following energy function

$$E_{AC}(\tilde{u}, t) = \iint_{u \in \mathbb{R}^2} w(u - \tilde{u})(I(u + t) - I(u))^2 du, \tag{1}$$

where $w(u)$ are weights that are typically zero outside some small window. It is usually chosen as a Gaussian function with some width σ. This function describes how much the image changes at the points $\tilde{u} = (x, y)$ by translating the image by a vector t, taking the difference, and intergrate over some neighbourhood around $\tilde{u}$. We can further investigate this function by linearizing the image around the point $\tilde{u}$,

$$I(u + t) \approx I(u) + \nabla I(u)^T t \Leftrightarrow I(u + t) - I(u) \approx \nabla I(u)^T t, \tag{2}$$

so that

$$E_{AC}(\tilde{u}, t) \approx \iint_{u \in \mathbb{R}^2} w(u - \tilde{u})(\nabla I(u)^T t)^2 du = \iint_{u \in \mathbb{R}^2} w(u - \tilde{u}) t^T \nabla I(u) \nabla I(u)^T t du. \tag{3}$$

Here

$$\nabla I(u) \nabla I(u)^T = \begin{bmatrix} \frac{\partial I}{\partial x}^2 & \frac{\partial I}{\partial x}\frac{\partial I}{\partial y} \\ \frac{\partial I}{\partial x}\frac{\partial I}{\partial y} & \frac{\partial I}{\partial y}^2 \end{bmatrix}. \tag{4}$$

The integration doesn't depend on t so we can write our energy as

$$E_{AC}(\tilde{u}, t) \approx t^T \begin{bmatrix} \iint_{u \in \mathbb{R}^2} w(u - \tilde{u}) \frac{\partial I}{\partial x}^2 du & \iint_{u \in \mathbb{R}^2} w(u - \tilde{u}) \frac{\partial I}{\partial x}\frac{\partial I}{\partial y} du \\ \iint_{u \in \mathbb{R}^2} w(u - \tilde{u}) \frac{\partial I}{\partial x}\frac{\partial I}{\partial y} du & \iint_{u \in \mathbb{R}^2} w(u - \tilde{u}) \frac{\partial I}{\partial y}^2 du \end{bmatrix} t. \tag{5}$$

We call

$$A(\tilde{u}) = \begin{bmatrix} \iint_{u \in \mathbb{R}^2} w(u - \tilde{u}) \frac{\partial I}{\partial x}^2 du & \iint_{u \in \mathbb{R}^2} w(u - \tilde{u}) \frac{\partial I}{\partial x}\frac{\partial I}{\partial y} du \\ \iint_{u \in \mathbb{R}^2} w(u - \tilde{u}) \frac{\partial I}{\partial x}\frac{\partial I}{\partial y} du & \iint_{u \in \mathbb{R}^2} w(u - \tilde{u}) \frac{\partial I}{\partial y}^2 du \end{bmatrix}, \tag{6}$$

The autocorrelation function. It is also known as the structure tensor or the orientation tensor. It is a symmetric positive semidefinite matrix, and hence its Eigenvalues are real and non-negative. The Eigenvalues and Eigenvectors of this matrix tells a lot about the local intensity structure around the point $\tilde{u}$. At smooth areas A will have two small Eigenvalues, at edgelike structures it will have one large and one small Eigenvalue, and at cornerlike structures it will have two large Eigenvalues. The Eigenvectors of A will be directed along the dominant directions of the local structure. Many feature detectors are based on the functions on A for instance the classic Harris corner detector. The Harris corners are chosen finding the local maxima of

$$\det(A(\tilde{u})) - \alpha \text{trace}(A(\tilde{u}))^2 = \lambda_0 \lambda_1 - \alpha(\lambda_0 + \lambda_1)^2, \tag{7}$$

where λ_j are the Eigenvalues of $A(\tilde{u})$. In Figure 1 the result of applying the Harris corner on an image is shown.

Figure 1: The result of the Harris corner detector. To the right only the strongest local maxima within some fixed radius.

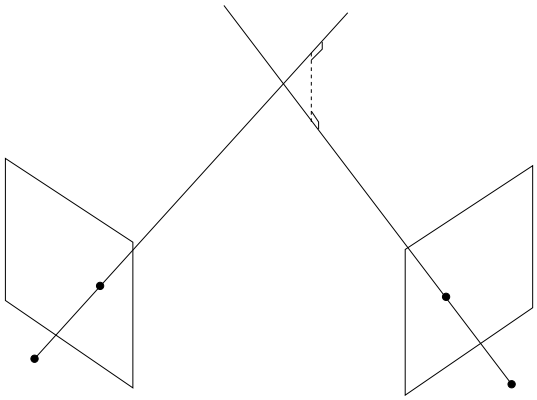

Figure 2: Geometrically triangulation is finding the 3D point closest to the image rays of the corresponding image points.

2 Triangulation

We will in this section describe a method for finding the position of a 3D point given that the projections of this point in a number of images is known, as well as the camera matrices. This problem is known as triangulation. If the camera matrix is known, then a 3D point projected in the corresponding image must lie on the 3D line that intersects the image point and the focal point of the camera. Hence triangulation can be seen geometrically as finding the intersection of a number of 3D lines, see Figure 2. It is clear that we at least need two lines, and so at least projections of the 3D point in two images. We can decribe our problem as

$$\lambda_i \mathbf{x}_i = P_i \mathbf{X}, i = 1 \dots n. \tag{8}$$

For n image points we have $3n$ equations and $n + 3$ unknowns, so $3n \geq n + 3$ or $n \geq \frac{3}{2}$. So this is consistent with our geometric argument. The problem is linear in the unknown λ_i and X, so we can find the least squares solution to this problem by formulating it as

$$Mv = 0, \tag{9}$$

with

$$M = \begin{bmatrix} P_1 & -\mathbf{x}_1 & 0 & \cdots & 0 \\ P_2 & 0 & -\mathbf{x}_2 & \cdots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ P_n & 0 & 0 & \cdots & -\mathbf{x}_n \end{bmatrix}, \tag{10}$$

and

$$v^T = \begin{bmatrix} X^T & \lambda_1 & \lambda_2 & \cdots & \lambda_n \end{bmatrix}. \tag{11}$$

As before we can find the solution using the singular value decomposition of M.

3 Homography Estimation

In homography estimation we want to find a projective transformation from $\mathbb{P}^k$ to $\mathbb{P}^k$, i.e. a homography. Usually $k = 2$ or $k = 3$. We will describe the problem for $k = 2$ but the procedure is exactly the same for any dimension. So given two sets of points u_i and v_i that are related by a homography H we want to solve

$$\lambda_i \mathbf{u}_i = H\mathbf{v}_i, i = 1 \dots n. \tag{12}$$

We write

$$H = \begin{bmatrix} h_1 \\ h_2 \\ h_3 \end{bmatrix}, \tag{13}$$

and

$$\mathbf{u}_i^T = [x_i \ y_i \ 1]. \tag{14}$$

Here H is a 3×3 matrix so h_1, h_2 and h_3 are 1×3 matrices. We can now write our problem as

$$Mv = 0, \tag{15}$$

with

$$M = \begin{bmatrix} \mathbf{v}_1^T & 0 & 0 & -x_1 & 0 & \cdots & 0 \\ 0 & \mathbf{v}_1^T & 0 & -y_1 & 0 & \cdots & 0 \\ 0 & 0 & \mathbf{v}_1^T & -1 & 0 & \cdots & 0 \\ \mathbf{v}_2^T & 0 & 0 & 0 & -x_2 & \cdots & 0 \\ 0 & \mathbf{v}_2^T & 0 & 0 & -y_2 & \cdots & 0 \\ 0 & 0 & \mathbf{v}_2^T & 0 & -1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & & \vdots \\ \mathbf{v}_n^T & 0 & 0 & 0 & 0 & \cdots & -x_n \\ 0 & \mathbf{v}_n^T & 0 & 0 & 0 & \cdots & -y_n \\ 0 & 0 & \mathbf{v}_n^T & 0 & 0 & \cdots & -1 \end{bmatrix}, \tag{16}$$

and

$$v^T = \begin{bmatrix} h_1 & h_2 & h_3 & \lambda_1 & \lambda_2 & \cdots & \lambda_n \end{bmatrix}. \tag{17}$$

And this can again be solved in a least squares sense using the singular value decomposition of M. For n points we have $3n$ equations and $8 + n$ unknown (remember that H is only determined up to scale), so we need at least 4 point correspondences to estimate a two-dimensional projective transformation. For higher dimensions we need a larger number of correspondences, but the procedure to estimate the homography will be the same.

3.1 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. So we assume that we have taken a number of images from the same location, and only rotated the camera between the images. 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 taken the images from 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 cameras and two corresponding image points we have the following system

$$\lambda_1 \mathbf{x}_1 = [R_1 \ 0] \begin{bmatrix} \mathbf{X} \\ 1 \end{bmatrix}, \tag{18}$$

$$\lambda_2 \mathbf{x}_2 = [R_2 \ 0] \begin{bmatrix} \mathbf{X} \\ 1 \end{bmatrix}. \tag{19}$$

This gives us

$$\left\{ \begin{array}{l} \lambda_1 \mathbf{x}_1 = R_1 \mathbf{X} \\ \lambda_2 \mathbf{x}_2 = R_2 \mathbf{X} \end{array} \right. \Leftrightarrow \left\{ \begin{array}{l} \mathbf{X} = \lambda_1 R_1^T \mathbf{x}_1 \\ \lambda_2 \mathbf{x}_2 = R_2 \mathbf{X} \end{array} \right. \Leftrightarrow \left\{ \begin{array}{c} \mathbf{X} = \lambda_1 R_1^T \mathbf{x}_1 \\ \lambda_2 \mathbf{x}_2 = \lambda_1 R_2 R_1^T \mathbf{x}_1 \end{array} \right. . \tag{20}$$

This means that we can write the last equation as

$$\lambda \mathbf{x}_2 = H \mathbf{x}_1, \tag{21}$$

with $\lambda = \frac{\lambda_2}{\lambda_1}$ and $H = R_2 R_1^T$. We know see that we can transfer points from the first image plane to the second by the use of a homography. We know from the previous discussion that we can estimate this homgraphy from at least four point correspondences. Once we have estimated the homgraphy, all points in an image can be transferred using this homography. If we have uncalibrated cameras the only difference is that the homography will be of the following form

$$H = K_2 R_2 R_1^T K_1^{-1}. \tag{22}$$