Lecture 9: Translation invariance and generic features

P. Perona and M. Weber
California Institute of Technology

EE/CNS148 - Spring 2004

1 Translation invariance

Consider the vector of feature coordinates $\overline{x}$ and call μ and Σ its mean vector and covariance matrix. Suppose that there are F features and dF entries in $\overline{x}$, where d is the number of scalar coordinates per feature (i.e. $d = 2$ in the image plane).

Call x_1 the vector of d coordinates of the first feature. We may obtain a translation-invariant model if we use the first feature as the origin of our reference frame. Call $\overline{x}'$ the new coordinates where $x'_i = x_i - x_1$. One may calculate directly the mean and covariance matrix of the new model. The transformation to the local reference frame is a linear operation and can be expressed as

$$\overline{x}' = L\overline{x}.$$

Assuming that x_1 is occupying the first d dimensions of x, we use

$$L = I_{dF \times dF} - \begin{pmatrix} I_{d\times d} & & \vdots & \\ I_{d\times d} & \cdots & 0 & \cdots \\ \vdots & & \vdots & \end{pmatrix}$$

where $I_{k\times k}$ is the identity matrix of size k. A well known fact from statistics states that a linear transformation of a random variable that is distributed according to a Gaussian density with mean μ and covariance matrix Σ will again yield a Gaussian density with

$$\mu' = L\mu \qquad \text{and} \qquad \Sigma' = L\Sigma L^T.$$

You can prove this fact starting from the definition of expectation and covariance and making use of the fact that the expectation of a random variable is a linear operator, and therefore it commutes with L.

The above formula will allow us to choose any feature point as a reference point and evaluate the foreground probability density using the remaining points, using the proper matrix L. This flexibility is important since any given feature might be absent from a constellation to be evaluated, so that a different one needs to be selected.

2 Generic feature detectors

2.1 Lucas-Tomasi-Kanade feature detector

Good generic features are points, corners and, in general, highly textured points. At those locations the brightness gradient of the image is 'rich' i.e. within a small neighborhood of the point of interest the gradient takes numerous directions. One can transform this intuition into an algorithm that will detect points of interest.

Call $\nabla I = [I_x, I_y]^T$ the brightness gradient of the image. Consider the 'moments' matrix

$$M(x,y) \doteq \nabla I \nabla I^T = \begin{bmatrix} I_x^2 & I_x I_y \\ I_y I_x & I_y^2 \end{bmatrix}$$

This matrix has rank one since it is obtained by taking the product of a single vector by itself. If we add two such rank-one matrices obtained from linearly independent vectors then we obtain a rank-two matrix, while if the vectors are linearly dependent the matrix will have rank one. So if we sum M over a small image neighborhood W its rank will be two if the neighborhood contains a feature, while it will be one if it contains a flat region or a straight line:

$$A(x,y) \doteq \int_W M(x,y)\,dx\,dy \approx \sum_{(i,j)\in W} M_{i,j}$$

therefore the components of A, $a_{11}, \ldots, a_{22}$, are calculated as:

$$\begin{aligned} a_{11} &= \sum_{(i,j)} I_x^2(i,j) \\ a_{12} = a_{21} &= \sum_{(i,j)} I_x(i,j) I_y(i,j) \\ a_{22} &= \sum_{(i,j)} I_y^2(i,j) \end{aligned}$$

In order to measure the 'goodness' of a feature we just have to check the condition number of the matrix A, i.e. the ratio of its eigenvalues: $c = \lambda_2/\lambda_1$. Since the matrix is 2x2 its eigenvalues may be computed directly from the entries of the matrix in closed form:

$$\begin{aligned} \lambda_1 &= \frac{a_{11}+a_{22}}{2} + \sqrt{\frac{(a_{11}-a_{22})^2}{4} + a_{12}^2} \\ \lambda_2 &= \frac{a_{11}+a_{22}}{2} - \sqrt{\frac{(a_{11}-a_{22})^2}{4} + a_{12}^2} \\ c &= \frac{\lambda_2}{\lambda_1} \end{aligned}$$

Therefore one can construct a simple feature detector by calculating c for every location in the image and declaring a feature wherever c exceeds a suitable threshold and has a local maximum.

2.2 The Förstner interest operator

This operator is able to detect points in the image which correspond to corners (defined as points where two or more lines intersect) as well as center points of circular symmetric patterns, such as dots or circles.

Our objective is to find the point which has the closest (squared) distance to all lines in the image. We interpret the gradient vector $\nabla I_i = \nabla I(x_i, y_i)$, at a given point $p_i = (x_i, y_i)^T$, as an indication for a line passing through the point along the direction perpendicular to ∇I_i. Given any point, $\hat{p}$, its distance to such a line is given by

$$d_i(\hat{p}) = \frac{\nabla I_i^T}{||\nabla I_i||}(\hat{p} - p_i).$$

The absolute value of ∇I_i is a measure of the contrast of the line passing through the point p_i. In the extreme case, where the absolute value is zero, we might even say that there is no line at this location. Therefore, taking the absolute value of the gradient as a measure of confidence for the presence of a line, we use $||\nabla I(i)||$ as a weighting factor for the d_i. If we integrate d over a window, W, we obtain a function of p which is small if p is close to all lines in the image.

$$D(\hat{p}) = \int_W d^2(\hat{p})\, dxdy = \int_W \left(\frac{\nabla I^T}{||\nabla I||}(\hat{p} - p(x,y))\right)^2 ||\nabla I||^2\, dxdy = \int_W \left(\nabla I^T \hat{p} - \nabla I^T p(x,y)\right)^2\, dxdy$$

Ideally, if this function equals zero, $\hat{p}$ is located exactly at the intersection of all the lines in the image. On the other hand, minimizing D with respect to $\hat{p}$ will provide us with the closest point to a corner in the image.[1] In order to find the minimum of D we set its derivative with repsect to $\hat{p}$ to zero:

$$\frac{\partial}{\partial \hat{p}} D(\hat{p}) = 2 \int_W \nabla I \nabla I^T \hat{p} - \nabla I \nabla I^T p(x,y)\, dxdy = 0$$

With the notation from the previous section we define $\int_W M(x,y)\, dxdy = \overline{M}$ and obtain

$$\overline{M}\hat{p} = \int_W M(x,y) p(x,y)\, dxdy.$$

If $\overline{M}$ is an invertible matrix, we can write down the solution as

$$\hat{p} = \overline{M}^{-1} \int_W M(x,y) p(x,y)\, dxdy$$

What if we cannot invert the matrix $\overline{M}$? According to the previous section, this simply indicates that the region of the image we are considering does not contain a good feature, and therefore no corner, so we can simply reject it.

In essence, what one should do is to select interesting windows in the image using the Lucas-Tomasi-Kanade method, and then apply Förstner's method only inside those windows.

[1] Note that this point could even lie outside of W!

The Förstner operator may be easily extended to detecting the center of circles in the image. Observe that the center of a circle is at the intersection of all the radii of the same, and that the radii intersect the circle at 90^o angles. Our objective is now to find the point which has the closest (squared) distance to all lines in the image that intersect circles at 90^o. We interpret the vector $\nabla I_i^\perp = \nabla I(-y_i, x_i)$, orthogonal to the gradient, at a given point $p_i = (x_i, y_i)^T$, as an indication for a line passing through the point along the direction parallel to ∇I_i. Given any point, $\hat{p}$, its distance to such a 'radius' line is given by

$$d_i(\hat{p}) = \frac{\nabla I_i^\perp}{||\nabla I_i||}(\hat{p} - p_i).$$

The rest proceeds as for the corner detector.