Equations

Harris eqs

Cornerness

$$E(u,v)=\sum w(x,y)[I(x+u,y+v)-I(x,y)]^2$$

Taylor series 2D first order approx

$$f(x+u,y+v)\approx f(x,y)+(uf_x(x,y)+vf_y(x,y))$$

Harris Corner Derivation

$$\sum[I(x+u,y+v)-I(x,y)]^2$$

Use first order approx to get:

$$\sum u^2I_x^2+2uvI_xI_y+v^2I_y^2$$

Written in a matrix equation this is:

$$\begin{bmatrix}u & v\end{bmatrix}=(\sum\begin{bmatrix}I_x^2 & I_xI_y\\ I_xI_y & I_y^2\end{bmatrix})\begin{bmatrix}u\\ v\end{bmatrix}$$

The quadratic approx simplifies to:

$$E=(u,v)\cong[u,v]M\begin{bmatrix}u\\ v\end{bmatrix}$$

Where M is a 2x2 matrix computed from image derivatives:

$$M=\sum w(x,y)\begin{bmatrix}I_x^2 & I_xI_y\\ I_xI_y & I_y^2\end{bmatrix}$$

Harris corner response can be measure from the eigenvalues of M; If both λ_1 and λ_2 are great, it's a corner. If only one is, it's an edge. If both are smol, it's a flat area. Eigenvalues are however slow to compute, so the following function can be used instead:

$$R=detM-k(traceM)^2$$

because

$$detM=\lambda_1\lambda_2$$

$$traceM=\lambda_1+\lambda_2$$

k is an empirical constant

finding edges, scale space

Sobel operator, common approx of DoG

$$S_x=\begin{bmatrix}-1 & 0 & 1\\ -2 & 0 & 2\\ -1 & 0 & 1\end{bmatrix}$$

$$S_y=\begin{bmatrix}1 & 2 & 1\\ 0 & 0 & 0\\ -1 & -2 & -2\end{bmatrix}$$

Second derivative of Gaussian = Laplacian of Gaussian

Laplacian image:

$$f=L_{Kernel}*Image$$

Kernel:

$$L=\sigma^2(G_{xx}(x,y,\sigma)+G_{yy}(x,y,\sigma))$$

Where Gaussian:

$$G(x,y,\sigma)=\frac{1}{\sqrt{2\pi\sigma}}e^{-\frac{x^2+y^2}{2\sigma^2}}$$

DoG Pyramid:

$$\begin{aligned}D(x,y,\sigma)&=(G(x,y,k\sigma)-G(x,y,\sigma))*I(x,y)\\ &=L(x,y,k\sigma)-L(x,y,\sigma)\end{aligned}$$

Laplacian of Gaussian

$$LoG(x,y,t)=-\frac{1}{\pi t^2}(1-\frac{x^2+y^2}{2t})e^{-\frac{x^2+y^2}{2t}}$$

SIFT

The Hessian matrix can be defined as

$$H_{i,j}=\frac{\partial^2 f}{\partial x_i\partial x_j}$$

In SIFT, keypoints that do not satisfy the following equation are eliminated, because they are on the edge, and thus not as valuable as corner points.

$$\frac{trace(H)^2}{det(H)}<\frac{(r+1)^2}{r}$$

Nonlinear scale space

Nonlinear diffusion filtering

$$\frac{\partial L}{\partial t}=div(c(x,y,t)*\nabla L)$$

Where c is the conductivity function, which enables adaptation to the image structure. t is the scale parameter, larger values lead to simpler images.

AOS (Additive Operator Splitting) schemes, which are used for iteratively constructing nonlinear scale spaces

$$L^{i+1}=(I-\tau\sum_{t=1}^{m}A_l(L^i))^{-1}L^i$$

FED (Fast Explicit Diffusion) is a faster way to construct the scale space Given *a priori* $L^{i+1,0}=L^i$, a FED cycle is:

$$L^{i+1,j+1}=(I+\tau_jA(L^i))L^{i+1,j},j=0\cdots n-1,i=0\cdots M$$

The main idea is to perform M cycles of n explicit diffusion steps with varying step sizes τ_j:

$$\tau_j=\frac{\tau_{max}}{2cos^2(\pi\frac{2j+1}{4n+2}}$$

Loss equations for CNN based descriptors and pipelines

Hinge loss for L2 distance metric. $p_1\&p_2$ are patches, C is upper bound loss.

$$l(x_1,x_2)=\begin{cases}\|D(x_1)-D(x_2)\|_2, & p_1=p_2\\ max(0,C-\|D(x_1)-D(x_2)\|_2), & p_1\neq p_2\end{cases}$$

Overall loss function for end-to-end pipeline

$$\mathcal{L}(p_i)=\|g(p_i^1,f_w(p_i^1))-g(p_i^2,f_w(p_i^2))\|_2^2$$

$f_w(p_i^*)$ = orientation for patch p of ith pair
g(p,x) = patch descriptor for patch p, rotated to x degree
Training makes $f_w(p_i^*)$ converges to an orientation estimator pointing to the dominant direction of patch p.

TILDE losses

Loss 1: classification-like loss

$$\mathcal{L}_c(\omega) = \gamma_c \|\omega\|_2^2 + \frac{1}{K}\sum_{i=1}^{2} max(0, 1 - y_i F(x_i;\omega))^2$$

For the ith sample xi, induce F(xi;w) to be close to +1 for positive samples (yi = +1) and -1 for negative samples (yi = -1)
Loss 2: shape regularizer loss, to normalize the response for positive samples

$$\mathcal{L}_s(\omega) = \frac{\gamma_s}{K_p}\sum_{i|y_i=+1}\sum_n \|w_{n\eta_i(n)} * x_i(w_{n\eta_i(n)}^T x_i)h\|_2^2$$

$$h(x,y) = e^{\alpha(1-\frac{\sqrt{x^2+y^2}}{S})} - 1$$

Loss 3: temporal regularizer loss, to enforce repeatability of varying illumination conditions

$$\mathcal{L}_t(\omega) = \frac{\gamma_t}{K}\sum_{i=1}^{K}\sum_{j\in\mathcal{N}_i}(F(x_i;\omega) - F(x_j;\omega))^2$$

RANSAC

RANSAC iteration count

$$N = \frac{\log(1-p)}{\log(1-(1-\epsilon)^s)}$$

p = success rate, prob that at least one sample has no outliers
ϵ = outlier ration
s = sample size, line 2, circle 3
Error, or orthogonal distance between point and line

$$d_i = \frac{|ax_i - y_i + b|}{\sqrt{a^2+1^2}}$$

Circle equation

$$(x-a)^2 + (y-b)^2 = r^2$$

circle error

$$d_i = |\sqrt{(x_i-a)^2 + (y_i-b)^2} - r|$$

Algorithm evaluation criteria

Repeatability The geometric stability of detected interests points between different images of the given scene under varying conditions — Detect the same point independently in both images

Distinctiveness/informativeness If all descriptors lie close together, their information content is low. Spread out descriptors have more information content. — reliable matching of corresponding point

Localization accuracy How accurately an interest point can be located at a specific 2D location. — where exactly is the point

Recall and precision:

$$\text{recall} = \frac{|\{\text{relevant documents}\} \cap \{\text{retrieved documents}\}|}{|\{\text{relevant documents}\}|}$$

$$\text{precision} = \frac{|\{\text{relevant documents}\} \cap \{\text{retrieved documents}\}|}{|\{\text{retrieved documents}\}|}$$

Algorithm steps

Harris

1. Compute x and y derivatives of image
2. Compute products of derivatives at every pixel
3. Compute the sums of the products of derivatives at each pixel
4. Define at each pixel the Hessian matrix
5. Compute the response of the detector at each pixel
6. Threshold on value of R. Compute nonmax suppression.

SIFT

1. Search for potential points of interest by creation of a Difference of Gaussian (DoG) scale-space pyramid as image representation and filtering for extreme values.
2. Further filtering and reduction of the obtained points from 1. to select stable points with high contrast. To each remaining point, its position and size are assigned.
3. Orientation assignment to each point by finding a characteristic direction.
4. Feature vector calculation based on the characteristic direction from 3. to provide rotation invariance.
5. The whole process is stacked in a way that only a subset of elements from the beginning of a stage is passed onto the next stage.

SURF

1. Find image interest points — Use determinant of Hessian matrix
2. Find Major interest points in scale space — Non-maximum suppression on scaled interest point maps
3. Find feature "direction" – We want rotationally invariant features
4. Generate feature vectors

DAISY

Mostly differs in its sampling pattern, which consists of concentric circles.

Binary descriptors

The point of binary descriptors is to describe the feature with a series of pixel pairs around the keypoint. If we have 512 pairs, we usually get a 512 bit binary string. The values of the string are gotten by comparing the pairs; For example, if we compare intensity gradients, the value is 1 if the second pixel's gradient is lower, and 0 if it is higher, or something like that.

BRIEF Random sampling pairs

ORB Orientation calculation using moments, learned sampling pairs

BRISK Sampling pattern is concentric circles, with more points on outer rings. Orientation calc by comparing gradients of long pairs. Sampling pairs are only short pairs.

FREAK Overlapping concentric circles with more points on the inner rings. Orientation calc by comparing gradients of preselected 45 pairs. Learned sampling pairs.

Advanced methods

Non-linear scale space — Intensity Order Pooling — Using multiple scales – learned descriptors, DNN or tradmachine learning.