ITMO

Feature and Object Descriptions

Computer Vision

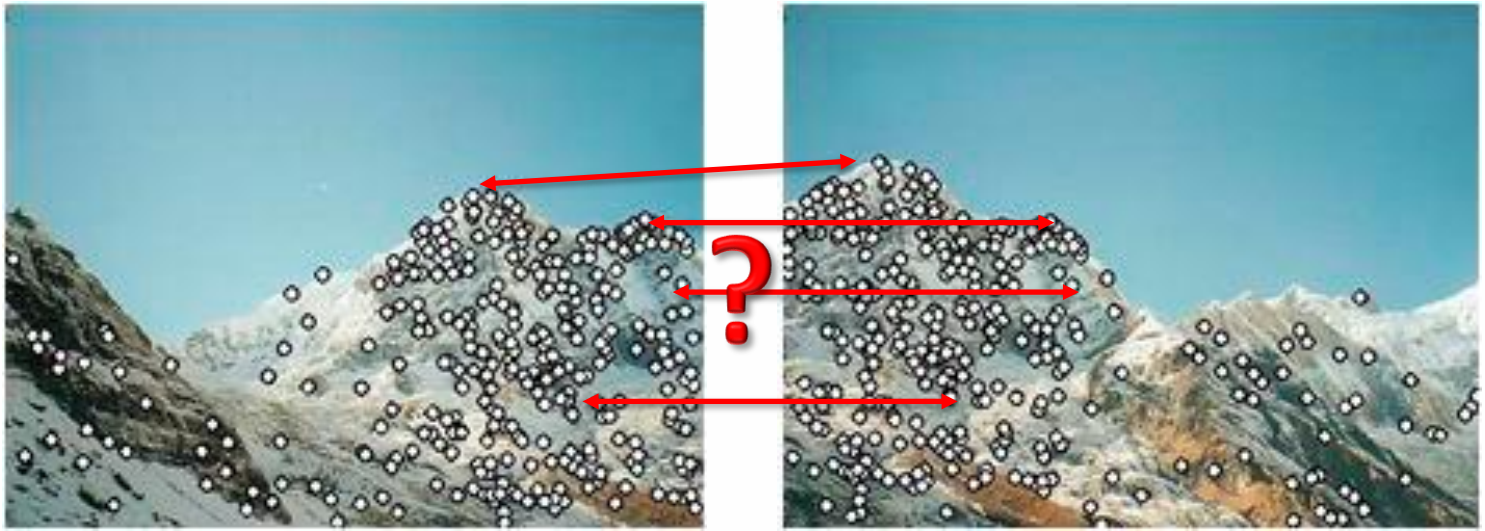
Outline

- **Feature and Object Descriptions**
  - Descriptors for contours
  - Descriptors for areas
  - Descriptors for points
  - Points matching
  - Model matching

Feature and Object Descriptions

Matching problem

How to match features on different images?



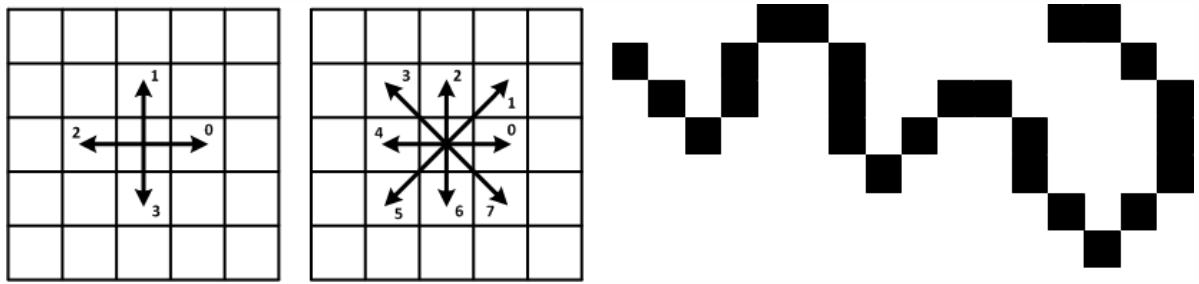
We have to describe features to be able to compare them.

Description of the contours

The result of detectors and segmentation algorithms is a set of special points for which it is necessary to construct a mathematical description.

Contour (chain) codes

Starting from the first point, the contour is traversed clockwise, with each subsequent point being encoded with a number **from 0 to 7**, depending on its location.



Curve coding example: 77121076711076771122334.

Description of the contours

ITMO

Contour (chain) codes

Disadvantages:

- dependence on the starting point of encoding;
- do not have the property of invariance to rotation;
- instability to noise, local changes in the contour can lead to different encoding results.

Description of the contours

Piecewise polynomial approximation

Search for a curve passing near a given set of contour points.

The curve is divided by separate nodes into segments, while the approximating function on each of the segments looks like:

$$f(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_n x^n,$$

where a_i – are the coefficients of the polynomial to be determined on each segment.

Description of the contours

ITMO

Piecewise linear approximation

For each pair of nodes, it is necessary to determine two coefficients a_0 and a_1, the total number of coefficients to be determined is $2(n+1)$, where n is the total number of nodes.

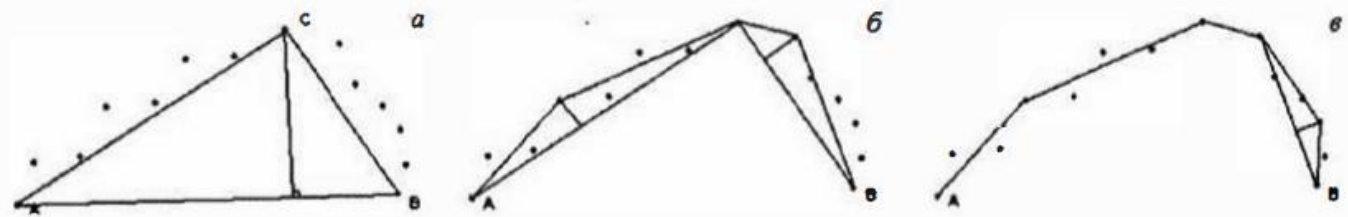
For piecewise linear approximation, an iterative algorithm for the selection of endpoints can be used:

1. The end points of the contour A and B are connected by a straight line.
2. Distances to line AB are calculated for the remaining points.
3. The point that has the greatest deviation from line AB is taken as an additional node.
4. The curve is replaced by two segments AC and CB.

The procedure continues until the maximum value of the points deviation is less than the specified threshold. The accuracy of the piecewise linear approximation approximation is determined by the threshold value.

Description of the contours

Piecewise linear approximation



Iterative selection of end points: on the left - the first stage;
in the center - the second stage; on the right is the third stage.

Disadvantages:

- the approximating function is not smooth (the first derivatives are discontinuous at the grid nodes);
- dependence of the approximation results on the initial experimental data.

Description of the contours

ITMO

Spline fit

- In practice, cubic splines are often used for approximation.
- Cubic splines give a high approximation accuracy and smoothness of the function.
- If the function being approximated has strong inflections, then in some cases the cubic spline gives outliers.
- The spline of the first degree in this situation does not allow outliers, but it is difficult to ensure the required accuracy of the approximation.
- Significant difficulties arise in the case of approximation of functions with large values of curvature.
- The use of both cubic and first degree splines is associated with a large number of interpolation nodes.

Description of the contours

Rational splines

They combine the properties of first-degree and cubic splines, allow approximating functions with large curvature values and with breakpoints.

A rational spline is a function $S_R(x)$, which on each segment $[x_i, x_{i+1}]$ has the form:

$$S_R(x) = a_i t + b_i(1-t) + \frac{c_i t^3}{1 + p_i(1-t)} + \frac{d_i(1-t)^3}{1 + q_i t}$$

where $t = \frac{x - x_i}{x_{i+1} - x_i}$, p_i, q_i – are given numbers, and $0 < p_i, q_i < \infty$.

The parameters p_i, q_i define the properties of rational splines:

1. If p_i, q_i are close to zero, then the rational spline becomes cubic;
2. if the parameters p_i, q_i are large enough, then the estimates of the spline error are comparable to a first-degree spline.

In most cases, it is customary to assume $p_i = q_i$.

Description of the contours

Natural curve representation

- The natural presentation of the curve implies the absence of connection points and branches on the contours.
- The contour is represented as a one-dimensional function of an attribute on the length of the arc.
- The length of the arc l_j of the discrete contour at the point $P(j) = (x_j, y_j)$ can be approximated as follows:

$$l_j = \sum_{i=1}^{j-1} \sqrt{(x_i - x_{i+1})^2 + (y_i - y_{i+1})^2}$$

- The contour representation is often used as a function of curvature $K(l)$, calculated by the formula:

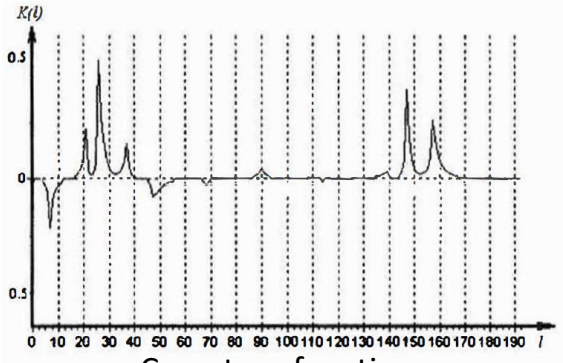
$$K(l) = K\big(x(l), y(l)\big) = \frac{f'_x f''_y - f''_x f'_y}{\sqrt{\left(f'^2_x + f'^2_y\right)^3}},$$

where f'_x, f'_y– the first derivatives with respect to *x* and *y* respectively;

f''_x, f''_y – the second derivatives with respect to *x* and *y*.

Description of the contours

Natural curve representation



Curvature function

Advantages of the curvature function:

- shift and rotation invariance.

Disadvantages:

- lack of invariance to scale;
- rectilinear (straight) contours cannot be represented as a function of curvature;
- the need to approximate curves for accurate calculation of derivatives at a point.

Description of the contours

Natural curve representation

- An analogue of curvature is the amount of contour inflection at a point.
- To obtain the inflection value, no curve approximation is required, but a discrete representation of the curve in the form of a sequence of pixel coordinates of the contour points is used.

To calculate the value of the inflection at the point $P(i)$ you must:

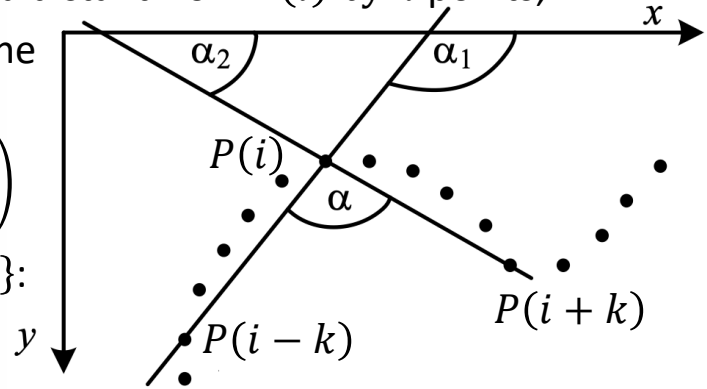
- choose two points of the sequence $P(i-k)$ and $P(i+k)$, equidistant from $P(i)$ by k points;
- determine the slope to the left $K\{L\}$ and right $K\{R\}$ from the point $P(i)$:

$$K\{L\} = \alpha_1 = arctg\left(\frac{y_i - y_{i-k}}{x_i - x_{i-k}}\right), K\{R\} = \alpha_2 = arctg\left(\frac{y_{i+k} - y_i}{x_{i+k} - x_i}\right)$$

- calculate the difference between the tilt angles $K\{L\}$ and $K\{R\}$:

$$K' = \alpha = K\{L\} - K\{R\}$$

where K' – the inflection value at the point.

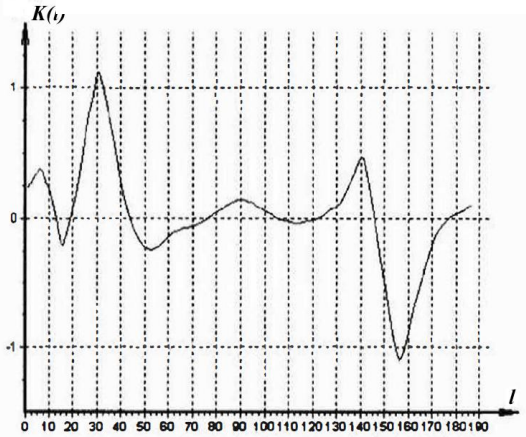


Calculating the inflection at a point

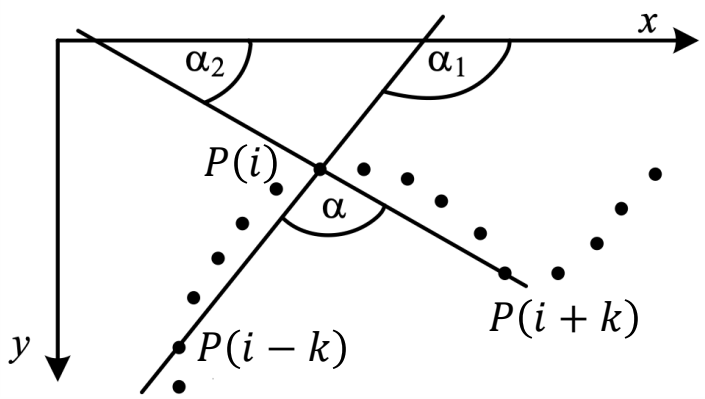
Description of the contours

Natural curve representation

If the contour does not contain branch (connection) points, then it can be represented as a one-dimensional inflection function $K'(l)$.



Inflection function



Calculating the inflection at a point

Description of the contours

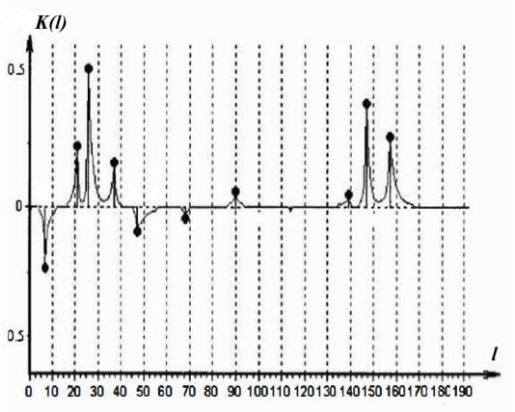
Singular points of contours

- The number and positions of contour singular points (points of maximum inflection, local extrema of the curvature function, end points, branch points) can be used as characteristic features.
- First of all, try to select corner points on the contour, because endpoints and branch points are not reliable enough and are highly susceptible to noise.
- A reliable way to identify special points is to search for the extreme values of any attribute of the contour, for example, the extrema of the curvature function, for the search of which it is necessary:
    1. Perform piecewise polynomial approximation of the contour;
    2. Construct a curvature function;
    3. Find all local extrema of curvature.

Description of the contours

Singular points of contours

Piecewise polynomial approximation of the curve allows you to more accurately calculate the values of the first two derivatives in directions at points, and, consequently, the value of the curvature itself.

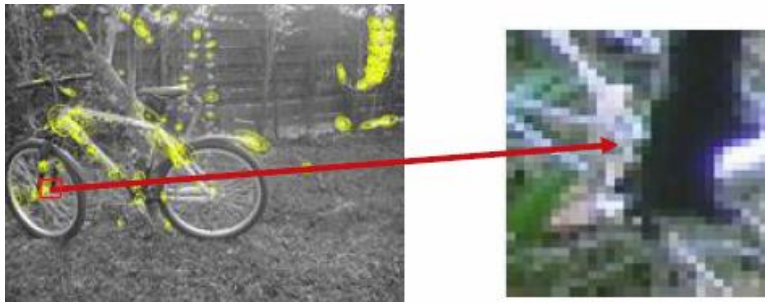


Local extrema of the curvature function

Description of areas

Description of selected areas

- Descriptors are vector-signs of a point's neighborhood.
- Features are built on the basis of information about the intensity, color and texture of special points.
- It is necessary to describe each point of interest with a certain set of parameters.



Characteristic point in the image

Description of areas

Typical feature set for areas (images)

1. **Topological features**:
   - **the number of disconnected components** - the number of separate objects in the image;
   - **the number of holes -** are there any holes inside the object;
   - **Euler's number** (Euler's characteristic) - the number of objects minus the number of holes).
2. **Geometric features** (characterize the shape of the image):
   - **the area of the image** S is calculated as the number of nonzero elements of the image;
   - **the position of the center of gravity** of the image, calculated in terms of static moments:

$$x_c = \frac{\int_\Omega B(x,y)xdx}{\iint_\Omega B(x,y)dxdy}, y_c = \frac{\int_\Omega B(x,y)ydy}{\iint_\Omega B(x,y)dxdy},$$

where Ω – the image in the Cartesian coordinate system (x, y);

$B(x, y)$ – the value of the intensity function at the point (x, y).

Description of areas

Typical feature set for areas

- position of the center of gravity of the area of a binary image:

$$x_c = \frac{\sum_\Omega x}{S}, y_c = \frac{\sum_\Omega y}{S}$$

- for a grayscale image:

$$x_c = \frac{\sum_\Omega x\, B(x,y)}{\sum_\Omega B(x,y)}, y_c = \frac{\sum_\Omega y\, B(x,y)}{\sum_\Omega B(x,y)}$$

- the perimeter of the area is equal to the sum of the modules of the elementary vectors of the contour connecting two neighboring elements (by 8-connectivity);

$$P = \sum_{k=1}^{N1} |P_1| + \sqrt{2} \sum_{k=N1+1}^{N} |P_2|$$

where P_1 and P_2 - are elementary vectors oriented along the grid and at an angle of 45°, respectively.

- the ratio of the square of the perimeter to the area of the image;

Description of areas

Typical feature set for areas

- **format** feature – the ratio of the sides of the circumscribed rectangle

To calculate the value of the **F** (format) feature, a scattering matrix is built from the contour points of the image:

$$E = \begin{pmatrix} S_{20} & S_{11} \\ S_{11} & S_{02} \end{pmatrix},$$

$$S_{pq} = \sum_{(x,y)\in D_\Omega} (x - x_c)^p (y - y_c)^q$$

and the eigenvalues of the scattering matrix are considered:

$$\lambda_i = \frac{s_{20} + s_{02}}{2} \pm \sqrt{\frac{(s_{20} + s_{02})^2}{4} + s_{11}^2}$$

Eigenvalues $\lambda_{1,2} > 0$ and $\lambda_{1,2} = 0$ in cases when the image is a straight line. The format is calculated using the formula ($\lambda_1 \geq \lambda_2$):

$$F = \frac{\lambda_1}{\lambda_2}$$

Description of areas

Typical feature set for areas

- **compactness** is calculated by the formula:

$$Z = \frac{S}{S_u}$$

where S – the area of the image, S_u – is the area of the circumscribed rectangle oriented as an equivalent ellipse.

- To determine **the orientation**, the eigenvectors of the scattering matrix are found:

$$\begin{pmatrix} S_{20} - \lambda_1 & S_{11} \\ S_{11} & S_{02} - \lambda_2 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = 0$$

Description of areas

Typical feature set for areas

- The magnitude of the projection of the contour point of the image (x, y) onto one of the eigenvectors (for example, (x_1, y_1), corresponding to the eigenvalue λ_1) is determined by the formula:

$$R = \sqrt{x^2 + y^2}\sin\left(\operatorname{arctg}\frac{y}{x} - \operatorname{arctg}\frac{y_1}{x_1}\right)$$

Substituting the values of the eigenvectors, we get:

$$R_1 = \left(y - \frac{\lambda_1 - S_{20}}{S_{11}}x\right)\frac{1}{\sqrt{x_1^2 + y_1^2}}, R_2 = \left(y - \frac{\lambda_2 - S_{02}}{S_{11}}x\right)\frac{1}{\sqrt{x_2^2 + y_2^2}},$$

where R_1 and R_2 - are the sides of the circumscribed rectangle oriented along the eigenvectors (the projection of the image onto the eigenvectors).

Description of areas

ITMO

Typical feature set for areas

- the perimeter and area of the circumscribed minimum area rectangle;
- the ratio of the area of the circumscribed rectangle to the area of the image;
- the ratio of the square of the perimeter of the circumscribed rectangle to its area;
- the format of the circumscribed rectangle;

$$F_1 = \frac{T_1}{T_2}$$

where T_1 and T_2 – are the sides of the circumscribed rectangle.

- the relative width and height of the image.

$$P_3 = \frac{P}{T_1}, P_4 = \frac{P}{T_2}$$

Description of areas

Typical feature set for areas

3. Moments

$$m_{\alpha\beta} = \iint_{\Omega} B(x,y) x^{\alpha} y^{\beta} \, dxdy$$

For a discrete case:

$$m_{pq} = \sum_{(x,y)\in\Omega} x^{p} y^{q} \, B(x,y)$$

- moments invariant to displacement: $\mu_{pq} = \sum_{(x,y)\in\Omega} (x - x_C)^{p} (y - y_C)^{q} \, B(x,y)$
- scale-invariant moments: $\eta_{pq} = \frac{\mu_{pq}}{\sum_{i+j=p+q} |\mu_{ij}|}$
- moments invariant to rotation: $M_1 = \eta_{02} + \eta_{20}$, etc.

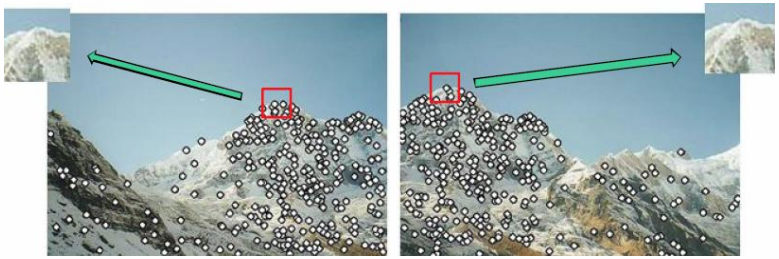
4. Textural features

Description of special points

Simple Neighborhood Approach

The simplest approach to identifying features:

- Take square neighborhoods, with sides parallel to the rows and columns of the image.
- Pixel intensities will be featured.
- When comparing points on images, we will compare their neighborhoods as images pixel by pixel.
- Such a neighborhood is invariant only to image shift.



Pixel-by-pixel comparison of the neighborhoods of singular points

Description of special points

Intensity invariance

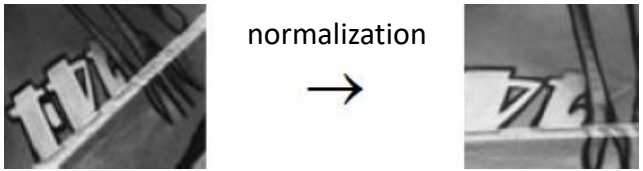
To achieve invariance to intensity, it is necessary to normalize the image intensities histogram as follows:

$$I' = \frac{I - \mu}{\sigma}$$

where I'– the normalized intensities,

μ – average value,

σ – variance.



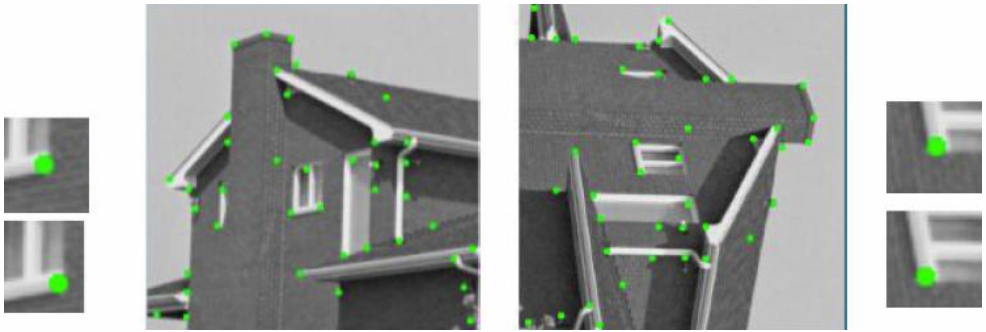
An example of normalizing the intensities of a neighborhood area

Description of special points

Simple Neighborhood Approach

Disadvantages:

- The point detector is rotation invariant, but the neighborhood is not;
- Small shifts, i.e., errors in finding a point make pixel-by-pixel comparison impossible.



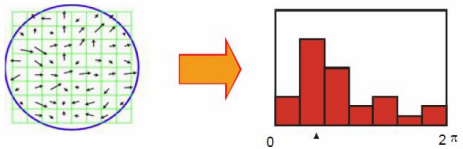
Detector invariance and descriptor non-invariance

Description of special points

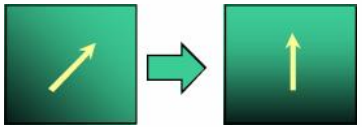
SIFT Descriptor (Scale-Invariant Feature Transform)

- This descriptor is used when using a DoG detector to determine the position and scale of a feature and is resistant to light changes and small shifts.
- The search for the orientation of a singular point is based on the idea of finding the main direction of pixel gradients in the vicinity of a point. **Algorithm:**

1. Calculate the histogram. Each point contribution is weighted with Gaussian centered at the singular point:



2. Rotate the fragment so that the dominant direction of the gradient is directed upwards:



3. If there are several local maxima, we assume that there are several points with different orientations.

David G. Lowe. "Distinctive image features from scale-invariant keypoints." IJCV 60 (2), pp. 91-110, 2004.David G. Lowe. "Distinctive image features from scale-invariant keypoints." *IJCV* 60 (2), pp. 91-110, 2004.

Description of special points

ITMO

SIFT Descriptor (Scale-Invariant Feature Transform)

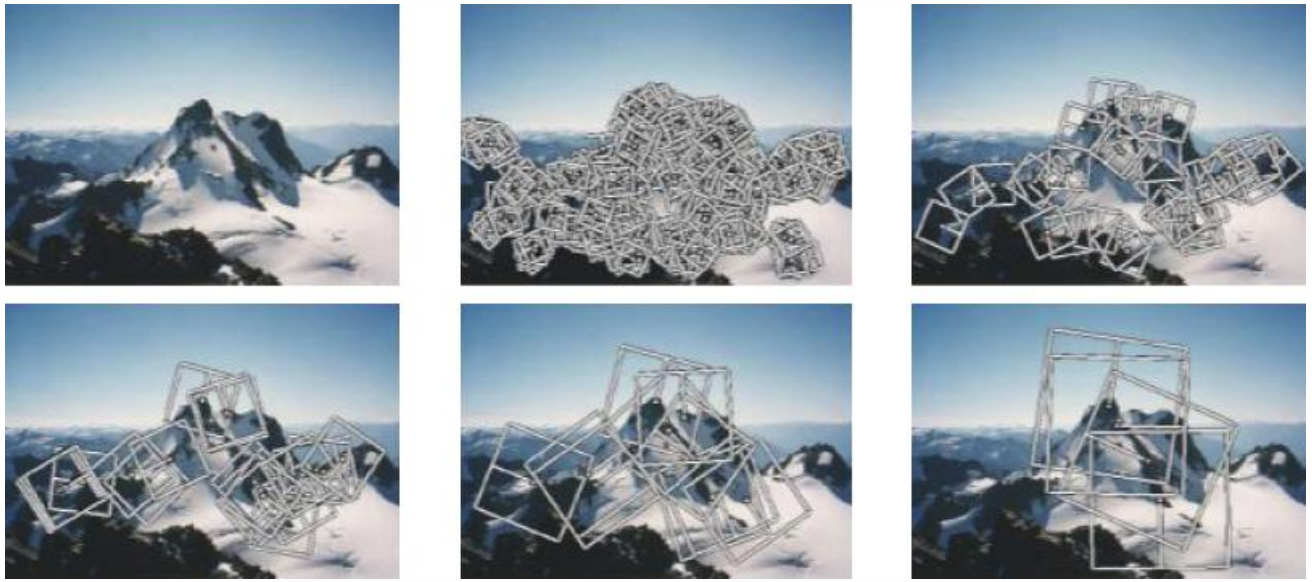
For each feature found, we now know the characteristic scale and orientation.

- Let's select the corresponding rectangular neighborhood (Rotation Invariant Frame)
- Bring the neighborhood to a standard size (scale).

Neighborhood features

Description of special points

SIFT Descriptor (Scale-Invariant Feature Transform)



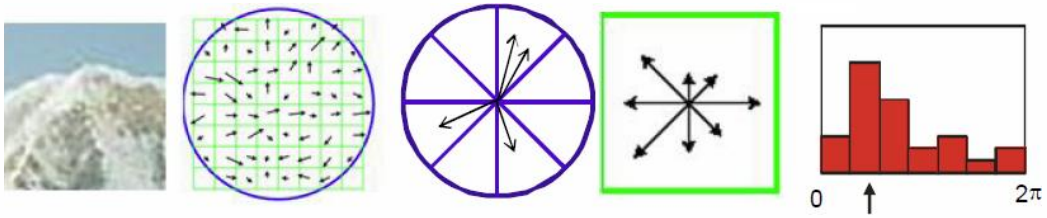
An example of a description of local features

Description of special points

SIFT Descriptor (Scale-Invariant Feature Transform)

Algorithm for constructing a SIFT descriptor:

1. Calculate the direction of the gradient in each pixel;
2. Quantize the orientation of the gradients by 8 cells (directions);
   - Tagging each pixel with a cell number;
3. Calculate the histogram of the directions of the gradients;
   - For each cell, calculate the number of pixels with the same gradient direction;
   - Each point contribution is evaluated with Gaussian, centered at the center of the neighborhood.

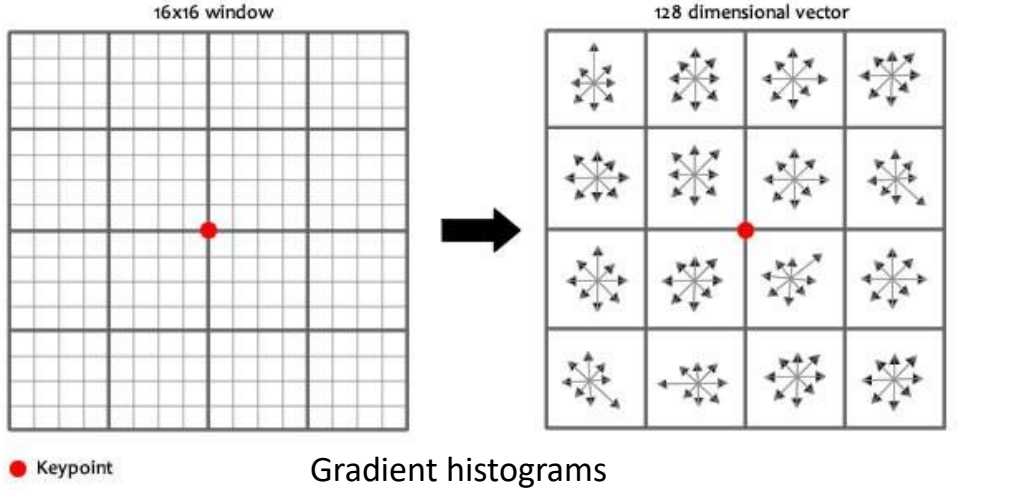


Gradient Orientation Histogram

Description of special points

SIFT Descriptor (Scale-Invariant Feature Transform)

4. Take the local properties into account. Divide the neighborhood into blocks by a grid, in each block we calculate its own gradient histogram.
   - Usually - a 4x4 grid, each histogram with 8 cells.
   - The standard length of a descriptor vector is 128 (4 * 4 * 8).
   - Compare as a vector (different metrics).



Gradient histograms

Description of special points

ITMO

Comparison of SIFT features

A feature vector with a length of 128 is essentially a histogram. The following metrics are used for comparison:

1. Standard metrics L_1, L_2.
2. Special for histograms:
   - Intersection of histograms:

$$D(h_1, h_2) = \sum_{i=1}^{N} \min\big(h_1(i), h_2(i)\big)$$

   - Chi-square distance χ^2:

$$D(h_1, h_2) = \sum_{i=1}^{N} \frac{\big(h_1(i) - h_2(i)\big)^2}{h_1(i) + h_2(i)}$$

Description of special points

Various SIFT modifications are used for color images.

RGB-SIFT

Implies 3 SIFT descriptors for each channel

C-SIFT

Uses channels O_1 and O_2:

$$\begin{pmatrix} O_1 \\ O_2 \\ O_3 \end{pmatrix} = \begin{pmatrix} \frac{R-G}{\sqrt{2}} \\ \frac{R+G-2B}{\sqrt{6}} \\ \frac{R+G+B}{\sqrt{3}} \end{pmatrix}$$

rgSIFT

Uses channels r and g:

$$\begin{pmatrix} r \\ g \\ b \end{pmatrix} = \begin{pmatrix} \frac{R}{R+G+B} \\ \frac{G}{R+G+B} \\ \frac{B}{R+G+B} \end{pmatrix}$$

Koen E. A. van de Sande, Theo Gevers and Cees G. M. Snoek, Evaluating Color Descriptors for Object and Scene Recognition, IEEE PAMI, 2010

Description of special points

SIFT Descriptor (Scale-Invariant Feature Transform)

Advantages:

1. The SIFT descriptor is specific, resistant to changes in lighting, small shifts.
2. SIFT (detector, neighborhood selection, descriptor) schema is a very efficient tool for image analysis.
3. Has become widespread.

Description of special points

PCA-SIFT descriptor (Principal Components Analysis-SIFT)

- For each feature point, a 41 × 41 neighborhood is considered.
- This gives neighborhood gradient vectors containing 2 × 39 × 39 = 3042 elements.
- The vectors are reduced to 32 elements by means of principal component analysis (PCA).

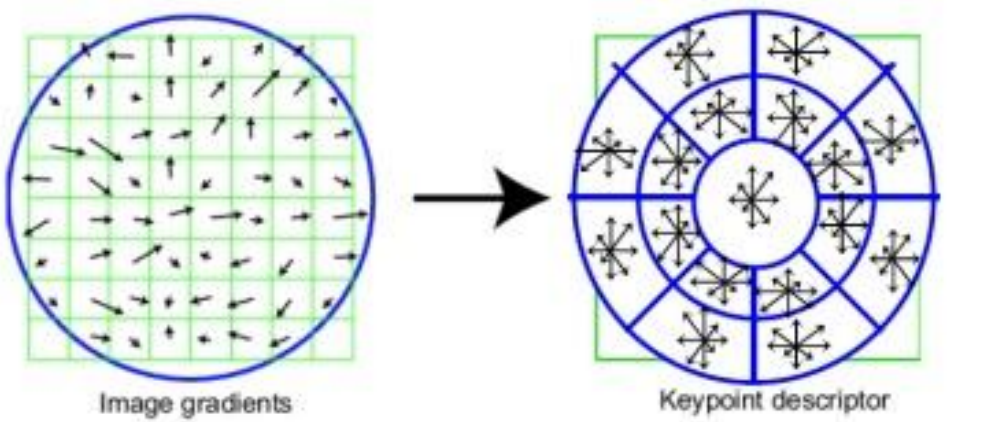
(A1) SIFT: 4/10 correct | (A2) PCA-SIFT (n=20): 9/10 correct | (B1) SIFT: 6/10 correct | (B2) PCA-SIFT (n=20): 10/10 correct

Y. Ke and R. Sukthankar, PCA-SIFT: A More Distinctive Representation for Local Image Descriptors CVPR, 2004.

Description of special points

GLOH descriptor (Gradient location-orientation histogram)

- A polar grid is used for dividing the neighborhood into bins: 3 radial blocks with radii of 6, 11 and 15 pixels and 8 sectors.
- The result is a vector containing 272 components, which is projected into a space of dimension 128 using principal component analysis (PCA).

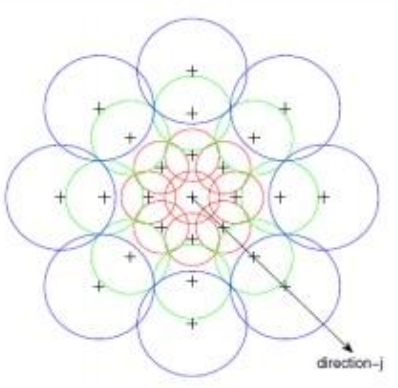


K. Mikolajczyk and C. Schmid,“A performance evaluation of local descriptors ,” IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 27, No. 10, pp. 1615-1630, Oct. 2005

Description of special points

DAISY descriptor

- Works on a dense set of pixels throughout the image.
- Runs 66 times faster than SIFT running on dense pixel counts.
- The ideas of constructing SIFT and GLOH descriptors are used.
- Similarly with GLOH, a circular neighborhood of the singular point is selected, with the bins being represented not by partial sectors, but by circles.



- For each bin, the same actions are performed as in the SIFT algorithm, but the weighted sum of the gradient magnitudes is replaced by the convolution of the original image with the derivatives of the Gaussian filter taken in 8 directions.
- The constructed descriptor has invariance, while solving the matching problem in the case when all pixels are considered special and requires less computational costs.

Tola, Engin, Vincent Lepetit, and Pascal Fua. "A fast local descriptor for dense matching." *2008 IEEE conference on computer vision and pattern recognition*. IEEE, 2008.

Description of special points

BRIEF descriptor (Binary Robust Independent Elementary Features)

Scheme for constructing feature vectors :

1. The image is split into patches (separate overlapping areas). Let's say patch P has dimensions $S \times S$ pixels.
2. A set of pairs of pixels $\{(X, Y)$, for $\forall X, Y$ in the neighborhood, $X, Y = (u, v)\}$ is selected from the patch for which a set of binary tests is constructed :

$$\tau(P, X, Y) = \begin{cases} 1, I(X) < I(Y) \\ 0, else \end{cases}$$

where $I(X)$ is the intensity of the pixel X.

3. For each patch, a set is selected containing n_d pairs of points (X_i, Y_i) that uniquely define a set of binary tests.
4. Based on these tests, a binary string is built:

$$f_{n_d}(P) = \sum_{i=1}^{n_d} 2^{i-1} \tau(P, X_i, Y_i)$$

Calonder, Michael, et al. "BRIEF: Computing a local binary descriptor very fast." *IEEE transactions on pattern analysis and machine intelligence* 34.7 (2011): 1281-1298.

Description of special points

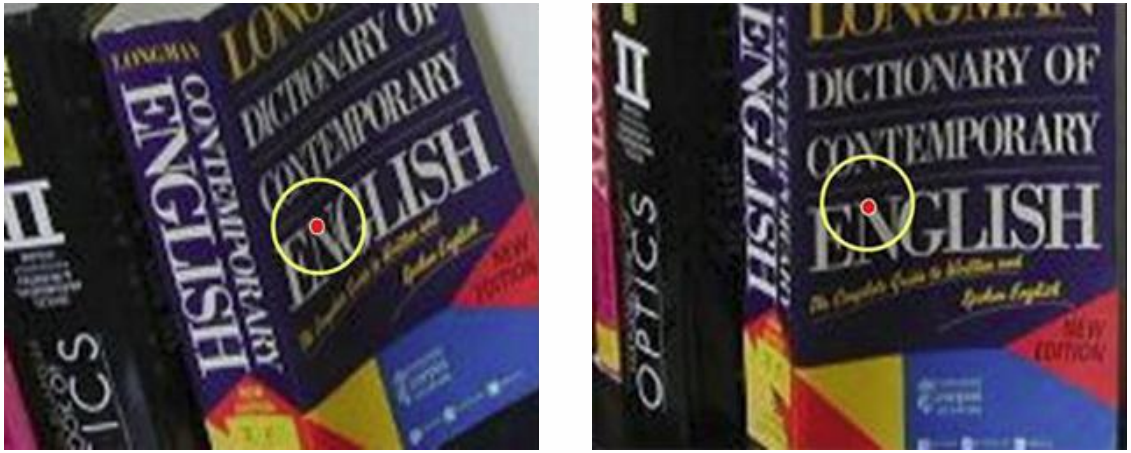
BRIEF descriptor (Binary Robust Independent Elementary Features)

- Provides recognition of the same areas of the image that were shot from different points of view.
- The recognition algorithm is reduced to the construction of a random forest or a naive Bayesian classifier on a certain training set of images and the subsequent classification of areas of test images.
- In a simplified version, the nearest neighbor method can be used to find the most similar patch in the training set.
- A small number of operations is provided due to the representation of the feature vector in the form of a binary string, and, as a consequence, the use of the Hamming metric as a measure of similarity.

Calonder, Michael, et al. "BRIEF: Computing a local binary descriptor very fast." *IEEE transactions on pattern analysis and machine intelligence* 34.7 (2011): 1281-1298.

Neighborhood normalization

Perspective distortion



Different fragments fall into the circular neighborhood -in the left image, half of the letter G is inside the circle, in the right it almost did not hit.

Neighborhood normalization

Perspective distortion

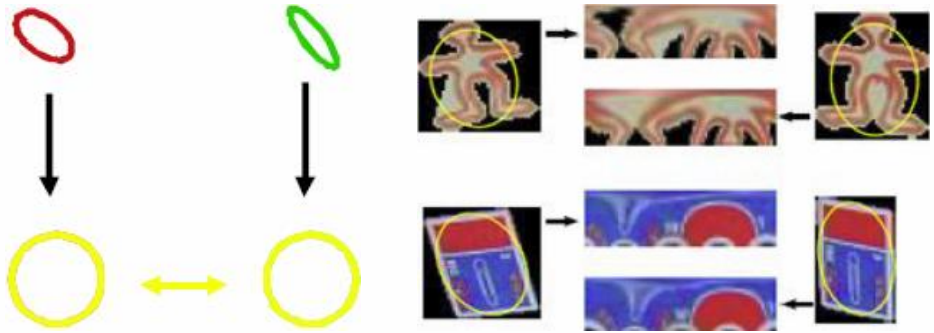
It is necessary to find the appropriate neighborhoods, and describe them with an ellipse, taking affine transformations into account.

Elliptic neighborhood of the characteristic point

Neighborhood normalization

Neighborhood normalization

To facilitate comparison of image fragments, it is necessary to find the parameters of the ellipse around the point of interest or area and bring the ellipses to the "canonical" form – the "common denominator".



Normalizing ellipsoids

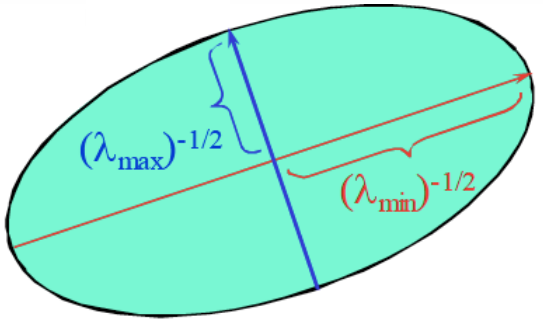
Neighborhood normalization

Affine adaptation

- The matrix M can be represented as an ellipse, in which the lengths of the axes are determined by the eigenvalues, and the orientation is determined by the matrix R.
- The main problem is that we count the matrix M by a round (square) neighborhood. In different images, the content will not match, and we will not be able to select the same areas (ellipses).

$$E(u,v) \approx [u \quad v]\mathbf{M}\begin{bmatrix}u\\v\end{bmatrix},$$

$$\mathbf{M} = \sum_{x,y} w(x,y)\begin{bmatrix}I_x^2 & I_xI_y\\ I_xI_y & I_y^2\end{bmatrix} = R^{-1}\begin{bmatrix}\lambda_1 & 0\\ 0 & \lambda_2\end{bmatrix}\mathrm{R}$$



Neighborhood normalization

ITMO

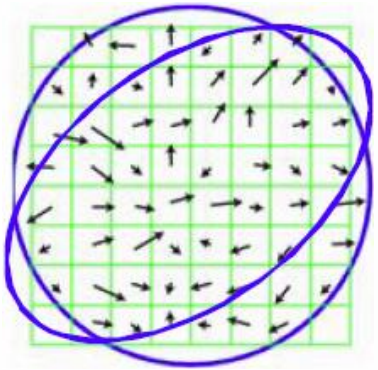
Affine adaptation

Solution: iterative neighborhood adaptation.

In the case of affine distortions, the problem is that the matrix of the second moments determined by the weights w (x, y) should be calculated from the characteristic shape of the region.

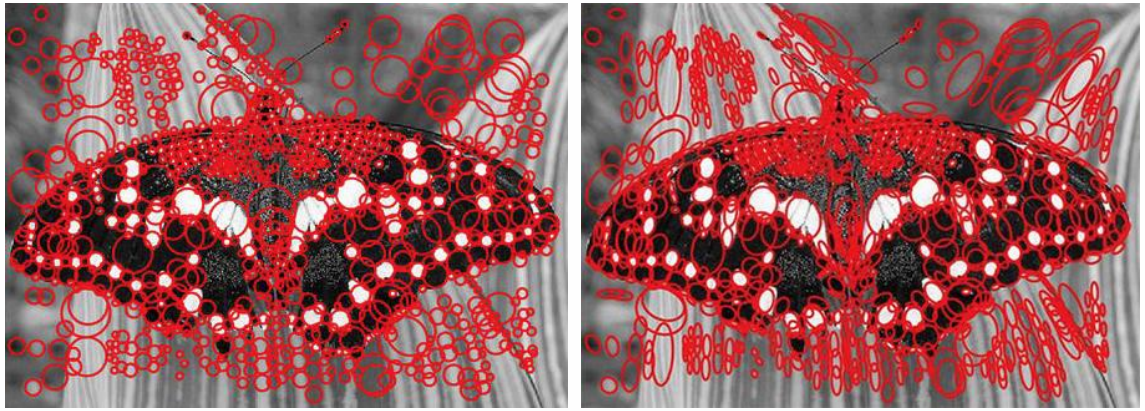
The iterative refinement algorithm is as follows:

1. Calculation of the matrix of moments using a round window.
2. Application of affine adaptation to obtain an elliptical window.
3. Recalculation of the moment matrix along the normalized neighborhood. Go to step 1.



Neighborhood normalization

Affine adaptation

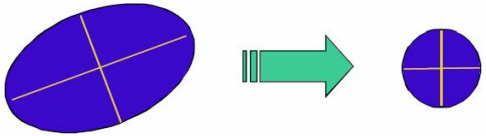


An example of affine adaptation, on the left scale-independent regions (blobs), on the right - refined blob neighborhoods.

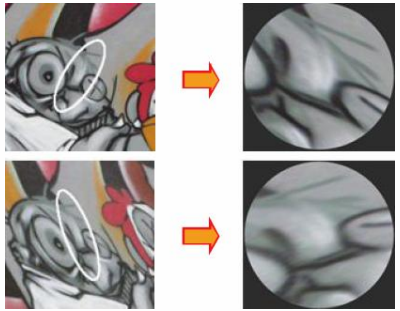
Neighborhood normalization

Normalization

Normalize the neighborhood by converting the ellipses to circles of unit radius. Moreover, the ellipse of the second moments can be considered the "characteristic shape" of the region.

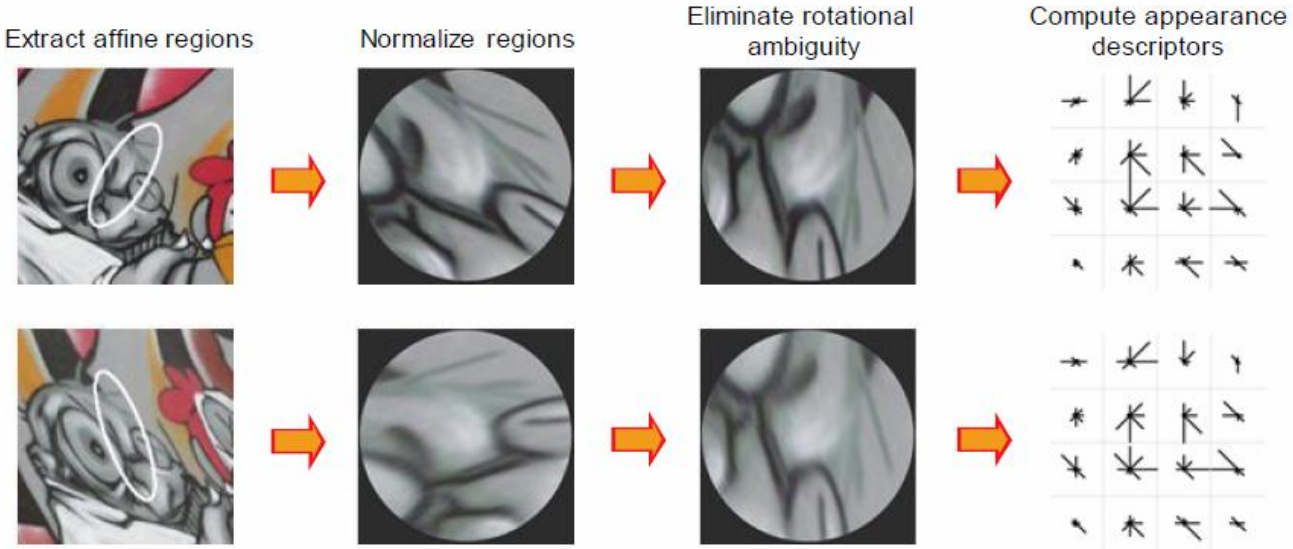


You can rotate and mirror a unit circle and it will remain a unit circle. This property can be used to find the desired orientation of the neighborhood. After normalization, calculate the dominant gradient and rotate the neighborhood.



Neighborhood normalization

Normalization

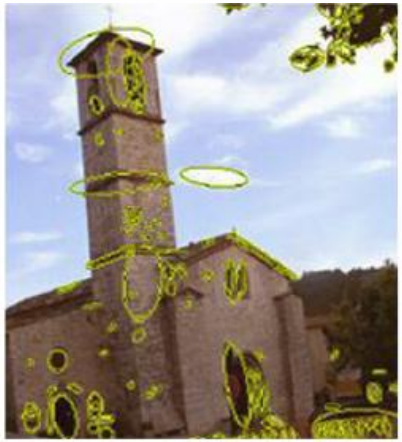


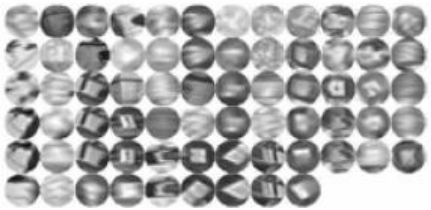
Points matching

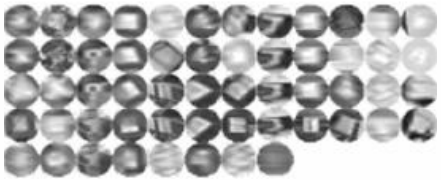
Comparison

We have a set of selected singular points and their descriptors.

How to match the same points in different images?







Points matching

Comparison

To match the points, it is necessary to generate candidate pairs: for each patch in one image, we find several patches most similar in terms of the selected metric in another image.

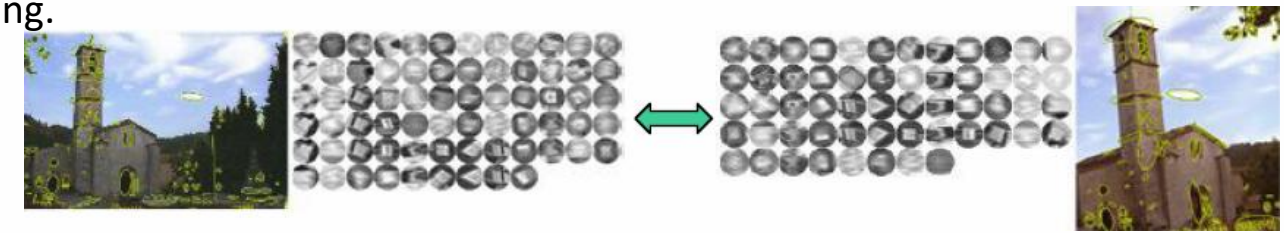
Methods for selecting pairs of candidate points:

1.Full search:

- For each feature, we calculate the distances to all features of the second image and take the best one.

2.Accelerated Approximate Measures:

- Hierarchical structures (kd-trees, vocabulary trees).
- Hashing.



Description of objects

Geometric models of structures

Local features belong to specific geometric structures:

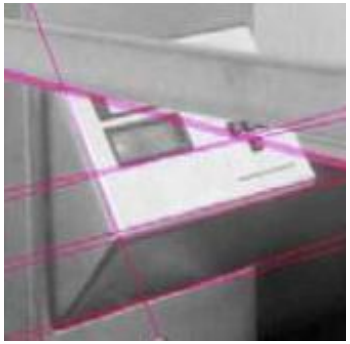
1. the corners of the windows lie on straight lines;
2. the edges of the windows lie on straight lines.

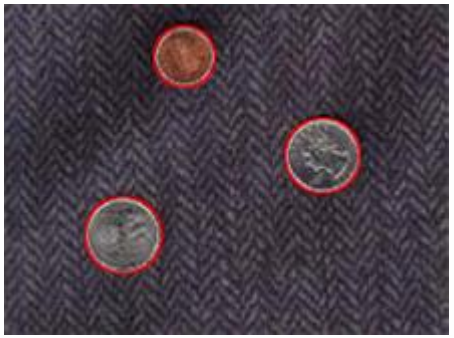
Based on these structures, their geometric models can be calculated.

On the left - highlighted features,
on the right - geometric models.

Description of objects

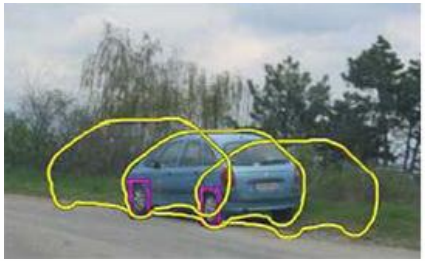
Simple Models - Parametric Curves





On the left are lines, on the right are circles.

Complex model - car



Description of objects

Parametric curves

$F(x, a) = 0$ – parametric model,

where a – the parameters of the model,

x – a vector corresponding to some points in space,

$X = \{x_i\}$ – a set of vectors corresponding to points in space.

Straight line: $F(x, a) = a_1 x_1 + a_2 x_2 + a_3 = 0.$

Circle: $F(x, a) = (x_1 - a_1)^2 + (x_2 - a_2)^2 - a_3 = 0.$

Conica: $F(x, a) = a_1 x_1^2 + a_2 x_1 x_2 + a_3 x_2^2 + a_4 x_1 + a_5 x_2 + a_6 = 0.$

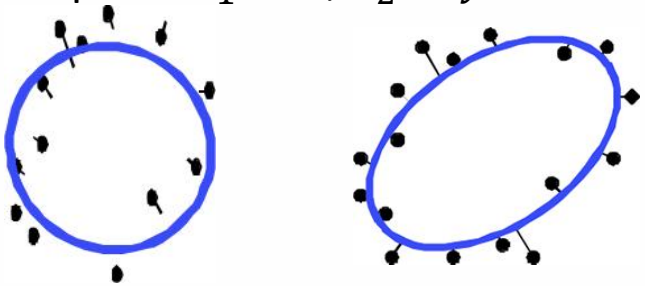
(conical section of a plane with a circular cone or a curve of the second order)

Description of objects

ITMO

Parametric curves

In the case of a two-dimensional plane $x_1 = x, x_2 = y$.



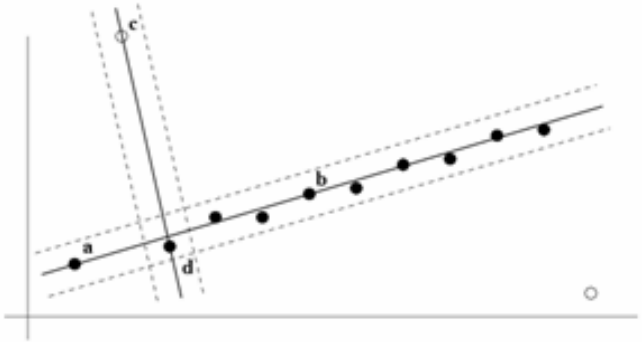
Tasks for estimating the parameters of the model in the image:

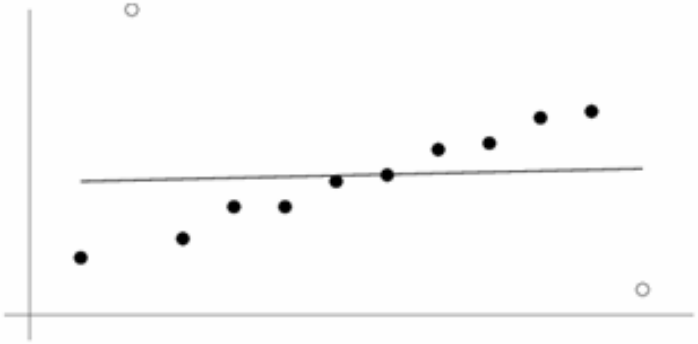
- Points are given that satisfy the model. It is necessary to calculate the parameters of the model.
- A model is given. Determine which points satisfy it, which do not.
- Points are given, some of them satisfy the model (inliers), some do not satisfy (outliers). Calculate the parameters of the model and split the data into inliers and outliers.
- Model fitting

Direct linear transformation

Anomalies are called **outliers**.

The points that **fit** the model are **inliers**.





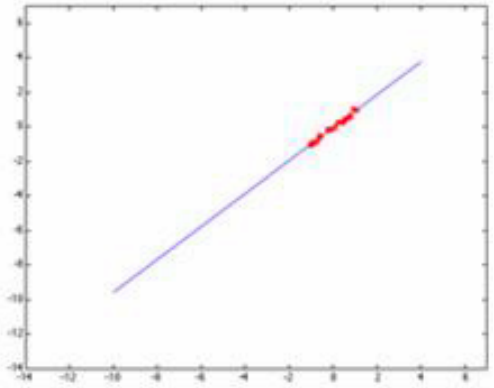
Direct linear transformation

Direct linear transformation (DLT)

Problem: a set of points with coordinates $(x_i, y_i), i = 1, n$ is given on a two-dimensional image. It is necessary to find the line that best approximates them.

Solution:

- using the least squares method, calculate a straight line with the smallest possible sum of squares of distances from points to a straight line;
- probabilistic formulation: search for the maximum likelihood by distance: $\hat{l} = \arg\max_l P[\{(x_i, y_i)\}|l]$.



Direct linear transformation

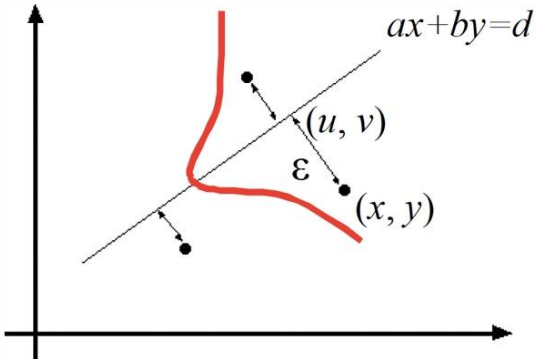
Model of a straight line with noisy Gaussian noise in points perpendicular to the line:

$$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} u \\ v \end{pmatrix} + \varepsilon \begin{pmatrix} a \\ b \end{pmatrix},$$

where the vector $[u \quad v]^{\mathrm{T}}$ – a point on the line,

ε – normally distributed Gaussian noise with zero mathematical expectation and standard deviation σ,

$[a \quad b]^{\mathrm{T}}$ – the normal vector.



Direct linear transformation

Probabilistic approach

- It is necessary to find the points with the maximum likelihood of being on the line (maximum likelihood).

- The likelihood of points with parameters a, b, d is calculated as follows:

$$P(x_1, \dots, x_n | a, b, d) = \prod_{i=1}^{n} P(x_i | a, b, d) \Rightarrow \prod_{i=1}^{n} \mathrm{e}^{-\frac{(ax_i + by_i - d)^2}{2\sigma^2}}$$

- When using the natural logarithm:

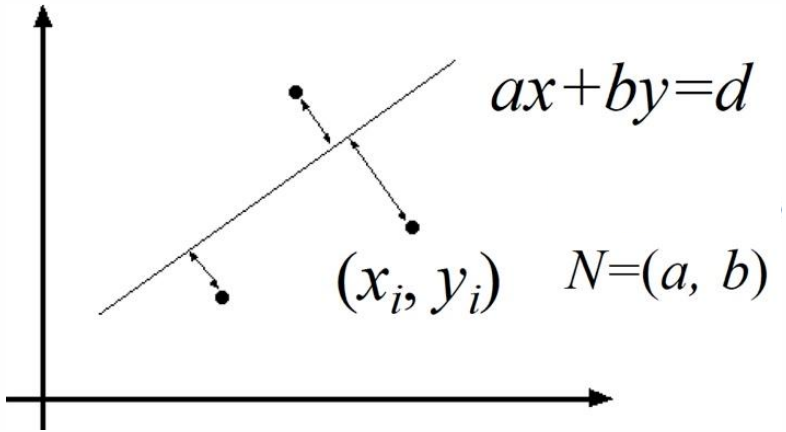
$$L(x_1, \dots, x_n | a, b, d) = -\frac{1}{2\sigma^2} \sum_{i=1}^{n} (ax_i + by_i - d)^2$$

Direct linear transformation

Geometrical approach

- Since the distance from the point (x_i, y_i) to the line along the normal is equal to $|ax + by - d|$, it is necessary to find such parameters a, b, d that minimize the function E:

$$E = \sum_{i=1}^{n} (ax_i + by_i - d)^2$$



Direct linear transformation

We differentiate the function E with respect to d and equate to zero:

$$\frac{\partial E}{\partial d} = \sum_{i=1}^{n} -2(ax_i + by_i - d) = 0,$$

and express d:

$$d = \frac{a}{n}\sum_{i=1}^{n} x_i + \frac{b}{n}\sum_{i=1}^{n} x_i = a\bar{x} + b\bar{y}$$

Substitute the resulting expression into the function E:

$$E = \sum_{i=1}^{n} \big(a(x_i - \bar{x}) + b(y_i - \bar{y})\big)^2 =$$

$$= \left\| \begin{bmatrix} x_1 - \bar{x} & y_1 - \bar{y} \\ \dots & \dots \\ x_n - \bar{x} & y_n - \bar{y} \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} \right\|^2 = (AN)^{\mathrm{T}}(AN).$$

Direct linear transformation

Differentiate $(AN)^{\mathrm{T}}(AN)$ by N:

$$\frac{dE}{dN} = 2\left(A^{\mathrm{T}}A\right)N = 0.$$

It can be seen that the solution to this matrix equation is the eigenvector $A^{\mathrm{T}}A$, corresponding to the minimum eigenvalue provided that $\|N\|^2 = 1$.

The expression $A^{\mathrm{T}}A$ allows you to find the singular values of the matrix A.

Direct linear transformation

SVD procedure

To simplify the search for singular numbers, consider the Singular Value Decomposition (SVD).

The matrix can be expanded as

$$A = UDV^{\mathrm{T}},$$

where U and V – unitary matrices ($UU^{\mathrm{T}} = I$),

D – a diagonal matrix consisting of singular numbers.

The following relations are valid:

$$A^{\mathrm{T}}A = VDU^{\mathrm{T}}UDV^{\mathrm{T}} = VDDV^{\mathrm{T}} = VD^2V^{\mathrm{T}}.$$

Direct linear transformation

ITMO

SVD procedure

We use this decomposition to calculate the least squares. Let the equation be given:

$$Ap = 0,$$

where the norm of the vector p: $\|p\| = 1$.

To find the minimum singular number, it is necessary to minimize the norm: $\left\|UDV^{\mathrm{T}}p\right\|$.

Given the equality on the previous slide:

$$\left\|UDV^{\mathrm{T}}p\right\| = \left\|DV^{\mathrm{T}}p\right\| \cdot \left\|V^{\mathrm{T}}p\right\| = \|p\|.$$

Direct linear transformation

SVD procedure

If $\left\|V^{\mathrm{T}}p\right\| = 1$, then it is necessary to minimize:

$$\left\|DV^{\mathrm{T}}p\right\|.$$

We denote $y = V^{\mathrm{T}}p$, then it is necessary to minimize:

$$\|Dy\|, \text{ if } \|y\| = 1,$$

and in the diagonal matrix D the columns are ordered in descending order.

In this case $y = (0, \dots, 0, 1)^{\mathrm{T}}$,

and $p = Vy$ – the last column of the matrix V.

Direct linear transformation

We use the OLS (Ordinary Least Squares) and SVD (Singular Value Decomposition) procedure to construct lines.

Let a set of points (x_i, y_i) be given.

To draw a line $ax + by = d$, you must:

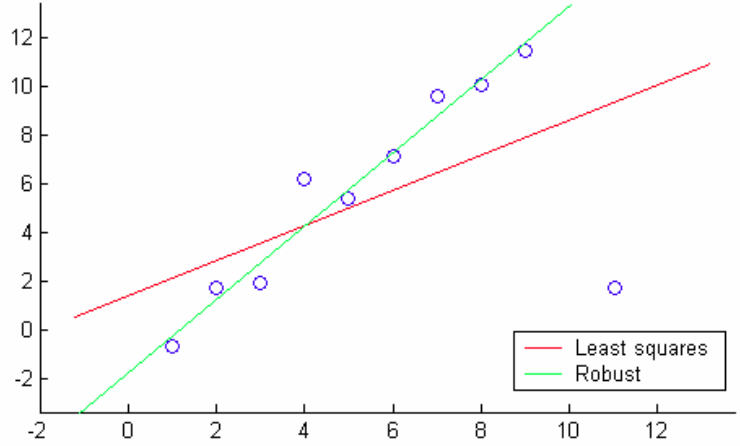
1. Calculate the mean point value $(\bar{x}, \bar{y})$;
2. Form matrix A containing offsets from the mean point value;
3. Execute the Singular Value Decomposition (SVD) procedure $A = UDV^{\mathrm{T}}$;
4. Calculate the parameters a and b from the last column of the matrix V;
5. Find $d = a\bar{x} + b\bar{y}$.

The least squares method for finding the parameters of models is called DLT (Direct Linear Transform)

Direct linear transformation

DLT problems:

- Often, some of the points obtained are not generated by the model $F(x, a)$.
- In such a situation, when estimating by the least squares method, the result can be arbitrarily far from the true one.
- For example, we have a set of pixels selected by the threshold and build a straight line based on them:



M-estimators

To reduce the influence of distant points, we parameterize the points in polar coordinates:

$$x\cos\theta + y\sin\theta = R,$$

then the objective function will take the form:

$$(\theta, R) = \arg\min_{(\theta,R)} \sum_i (x_i\cos\theta + y_i\sin\theta - R)^2 .$$

We denote $\varepsilon_i = x_i\cos\theta + y_i\sin\theta - R$, and modify the objective function:

$$(\theta, R) = \arg\min_{(\theta,R)} \sum_i \rho(\varepsilon_i),$$

where in the case $\rho(\varepsilon) = \varepsilon^2$ we obtain the least squares method.

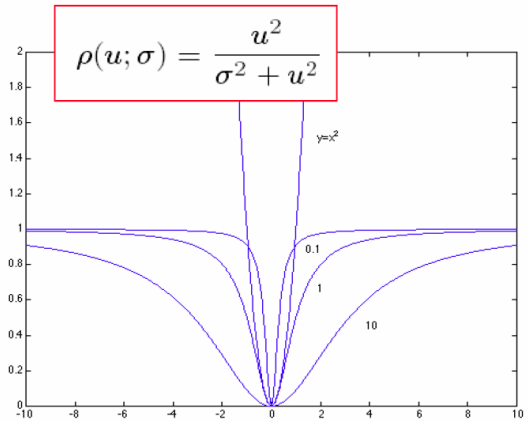
M-estimators

The following function is usually minimized:

$$\sum_i \rho(r_i(x_i, \theta), \sigma),$$

where $r_i(x_i, \theta)$ – the residual of the i point, subject to the model parameters θ,

ρ – a robust function with scale σ.



The robust function ρ behaves like the square of the distance for small values of u and flattens out as the value of increases.

M-estimators

The following functions are used as the most frequently used variants of the robust function ρ:

1. Tukey's function:

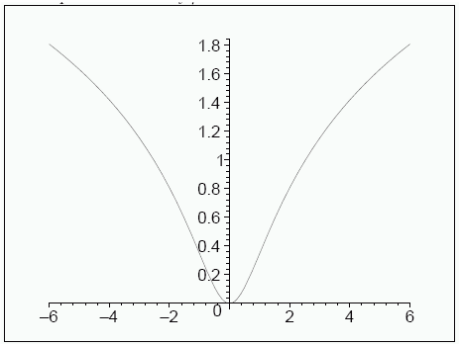
$$\rho(\varepsilon) = \begin{cases} \frac{K^2}{6}\left(1 - \left(1 - \left(\frac{\varepsilon}{K}\right)^2\right)^3\right), \text{if } |\varepsilon| \le K \\ \frac{K^2}{6}, \text{if } |\varepsilon| > K \end{cases}$$

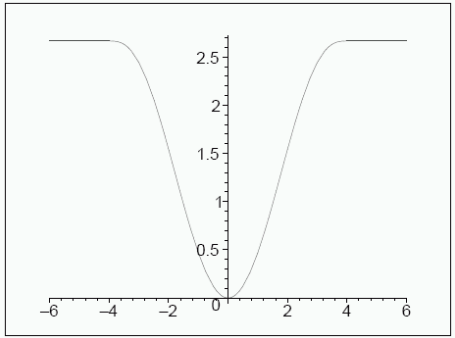
2. Cauchy function:

$$\rho(\varepsilon) = \frac{c^2}{2}\log\left(1 + \left(\frac{\varepsilon}{c}\right)^2\right),$$

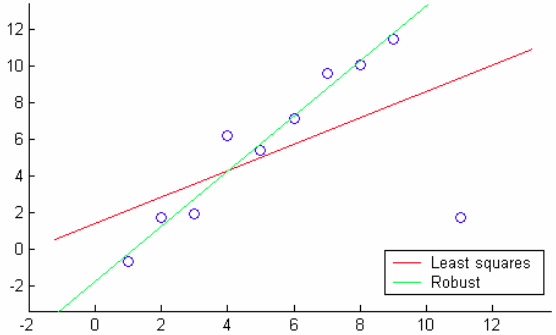
where K and c – tuning constants.

M-estimators





On the left is the Tukey function for K = 4.685, on the right is the Cauchy



Comparison of the least squares method and robust estimation modification

M-estimators

How to find the minimum of the objective function? With certain robust functions, this is extremely difficult to do.

Methods:

1. Nonlinear optimization methods;
2. Weighted Least Squares;
3. Iteratively overweighted least squares method.

Weighted Least Squares

Using the example of searching for straight lines:

$$(a, b, d) = \arg \min_{(a,b):\, a^2+b^2=1} \sum_i w_i (a x_i + b y_i + d)^2,$$

$$\sum_i w_i = 1,$$

where w_i – the weight of each point.

Let's construct a covariance matrix of points, which is a square symmetric nonnegative definite matrix.

- The main diagonal contains the variances of the coordinates of the points.
- Off-diagonal elements are covariance between points.

$$Cov = \begin{bmatrix} \sum_i w_i (x_i - \bar{x})^2 & \sum_i w_i (x_i - \bar{x})\,(y_i - \bar{y}) \\ \sum_i w_i (x_i - \bar{x})\,(y_i - \bar{y}) & \sum_i w_i (y_i - \bar{y})^2 \end{bmatrix}.$$

Weighted Least Squares

In a covariance matrix:

- **the maximum eigenvector** of the matrix Cov specifies the direction of the straight line,
- **the minimum eigenvector –** the direction of the normal $\overrightarrow{(a, b)}$.

The line passes through **the midpoint** $(\bar{x}, \bar{y})$,

where $\bar{x} = \frac{\sum_i w_i x_i}{\sum_i w_i}$,

$\bar{y} = \frac{\sum_i w_i y_i}{\sum_i w_i}$,

$d = -(a\bar{x} + b\bar{y})$.

Iteratively overweighted least squares

1. Get the initial approximation of the model by the least squares method:

$$\Theta^{(0)} = \left(\rho^{(0)}, \theta^{(0)}\right).$$

2. Set the iteration number $t = 1$.
3. For $\Theta^{(t-1)}$ calculate the current noise estimate

$$\sigma^{(t)} = 1{,}4826 \operatorname{median}_i \left| r_i^{(t)}\left(x_i, \Theta^{(t-1)}\right) \right|$$

as an unbiased (for a normal distribution) robust estimate of the mean error.

Iteratively overweighted least squares

4. Calculate the weights of points $w_i^{(t)}$ taking into account the function ρ,
   a. in general:

$$w_i = \frac{\rho'\left(\frac{\varepsilon_i}{\sigma}\right)}{\frac{\varepsilon_i}{\sigma}},$$

   b. in the case of the Tukey function:

$$w\left(\frac{\varepsilon}{\sigma}\right) = \begin{cases} \left(1 - \left(\frac{\varepsilon}{\sigma \cdot K}\right)^2\right), \text{if } \left|\frac{\varepsilon}{\sigma}\right| \leq K, \\ 0, \text{if } \left|\frac{\varepsilon}{\sigma}\right| > K, \end{cases}$$

   c. in the case of the Cauchy function:

$$w\left(\frac{\varepsilon}{\sigma}\right) = \frac{1}{1 + \left(\frac{\varepsilon}{c \cdot \sigma}\right)^2}$$

Iteratively overweighted least squares

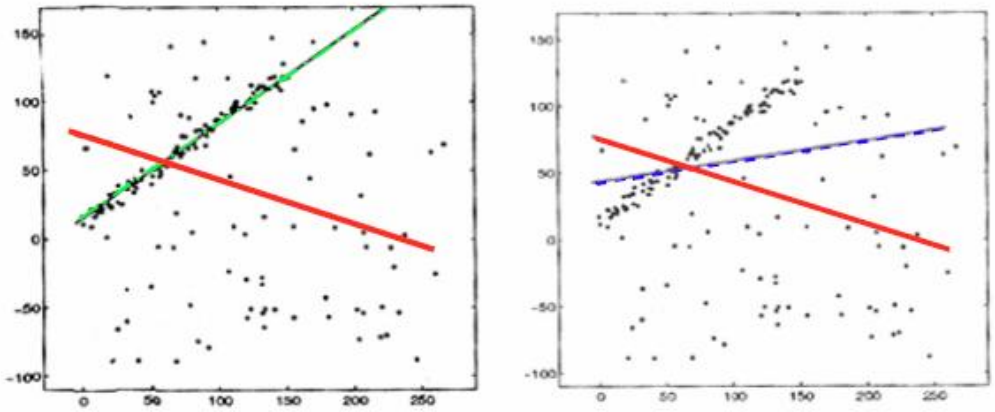
5. Using weighted least squares get $\Theta^{(t)}$.
6. If the desired tolerance is not achieved then go to step 3,

$$\left\|\Theta^{(t)} - \Theta^{(t-1)}\right\| > \varepsilon^*$$

where ε^* — the maximum desired deviation.

Iteratively overweighted least squares

Method results



The influence of the tuning constant c on the construction of the line:
the red line is the first step,
green - c = 1.5,
blue - c = 3.5.

M-estimators

Disadvantages:

1. The need for a good first approximation;
2. It is necessary to correctly calculate the balance of the weights in order for the algorithm to work with sufficient accuracy.

Random Sample Consensus

Random Sample Consensus - RANSAC

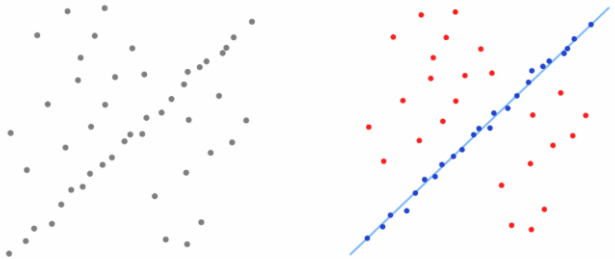
The idea: to estimate not all data, but only a small sample that does not contain outliers.

- Since it is not known in advance which points are outliers and which are not, it is possible to construct many samples at once in a random way.
- Then, for each of the samples, we build a hypothesis. After that, we choose such a hypothesis from among all that best fits with all the data.

Random Sample Consensus

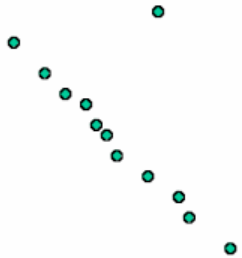
The main problem: the number of such samples is huge, so it is necessary to build hypotheses on the minimum sample size.

- For example, when inscribing a straight line into a set of points on a plane, this method takes as a basis only two points necessary to construct a straight line and use them to build a model.
- After that, it is checked how many points correspond to the model using an estimation function with a given threshold.



Random Sample Consensus

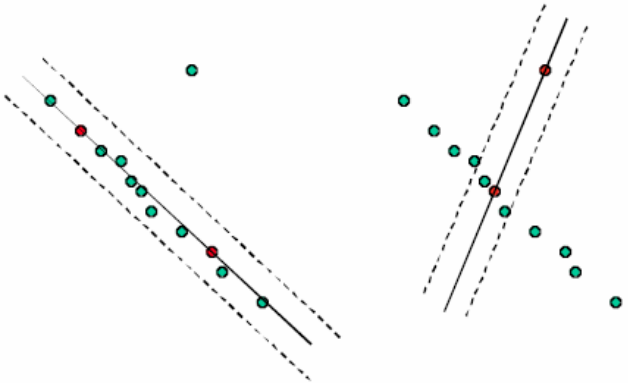
Example



The dataset to which the line must be entered.

Random Sample Consensus

Example

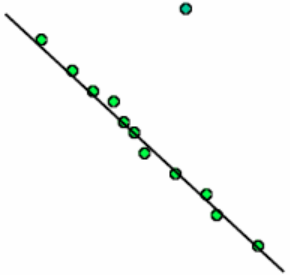


Two minimal samples (two points each) with cutoff along the proposed line.

On the left image, 11 points fell into the area, on the right - 4.

Random Sample Consensus

Example



The left sample more adequately describes the straight line (received more "votes"), and accordingly is the right decision.

Random Sample Consensus

ITMO

Basic scheme of the **RANSAC** algorithm is a loop with *N* iterations:

1. Build a sample $S \subset X$ $(x_i \in X)$. Typically, the sample size is the smallest possible size which is enough for the model parameters estimation.
2. Put forward a hypothesis Θ on a sample S.
3. Evaluate the degree of agreement between hypothesis Θ and set of input data X. Each point is labeled as "outlier" or “inlier".
4. After checking all points, it is checked whether the hypothesis is currently “the best” one or not by comparing to with previous “best” one. If yes, then it replaces the previous “best” hypothesis.

At the end of the cycle, the last best hypothesis is left, from which it is possible to determine the parameters of the model, as well as the points marked as "outliers" and “inliers“.

Random Sample Consensus

- To obtain a model built without outliers with a given probability p, the number of iterations *N* of the cycle can be calculated, if it is possible to indicate a given fraction of “outliers” e.
- The number of samples *N* is chosen so that the probability of choosing at least one sample without outliers would not be lower than the given one (for example, 0.99).

Thus:

$$p = 1 - (1 - (1 - e)^s)^N \Rightarrow$$

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

where *N* is the number of samples (the number of iterations),

p – the probability of getting a good sample in *N* iterations,,

S – the number of elements (points) in the sample,

e – the share of “outliers”.

Random Sample Consensus

To assess the degree of agreement between hypotheses, consider the functions for evaluating hypotheses:

1. **RANSAC**

$$R(\Theta) = \sum_i p(\varepsilon_i(\Theta)^2), \qquad p(\varepsilon_i(\Theta)^2) = \begin{cases} 1, \text{if } |\varepsilon_i| \le T \\ 0, \text{if } |\varepsilon_i| > T \end{cases}, i = \overline{1,n}$$

where $\varepsilon_i(\Theta)$ – the residual of the i-th point and the estimated hypothesis;

p – probability (1 – “inlier”, 0 – “outlier”);

T – the threshold chosen on the basis that the value of the probability of “inlier” is $p \approx 0{,}95$.

Typically, a Gaussian noise model with zero mathematical expectation is used such that $T^2 = 3{,}84\sigma^2$.

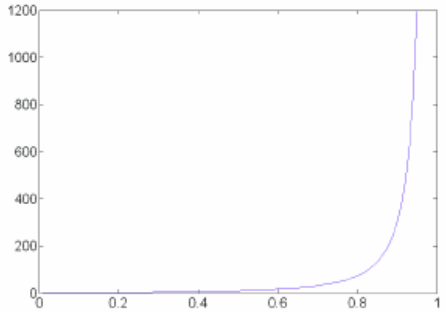
2. **LMS** (Least median squares)

$$R(\Theta) = median(\varepsilon_i(\Theta)^2), i = \overline{1,n}.$$

Random Sample Consensus

The number of samples grows rapidly with the growth of the sample size and the proportion of outliers.

	proportion of outliers e						
s	5%	10%	20%	25%	30%	40%	50%
2	2	3	5	6	7	11	17
3	3	4	7	9	11	19	35
4	3	5	9	13	17	34	72
5	4	6	12	17	26	57	146
6	4	7	16	24	37	97	293
7	4	8	20	33	54	163	588
8	5	9	26	44	78	272	1177



Dependence of the number of samples on the sample size and the proportion of "outliers"

How to minimize the number of samples if the fraction of ouliers is not known in advance?

Random Sample Consensus

ITMO

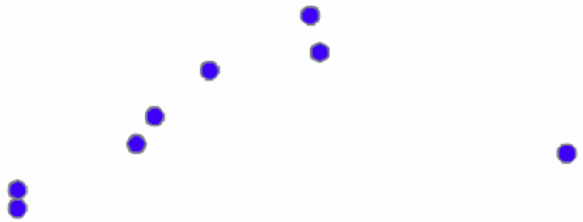
You can start the algorithm with a rough estimate, say 50%, and then refine the number of samples sequentially.

RANSAC Adaptive Algorithm Completion:

```
N=999999, sample_count = 0, p = 0.99;
while(N > sample_count) {
    //Basic RANSAC algorithm:
    //Construction of the sample S (the number of points in the sample), hypotheses, estimation of “inliers" in the inliers sample
    e = 1 - (inliers / S);
    N = log(1 - p) / log(1 - (1 - e)^S );
    sample_count++;
}
```

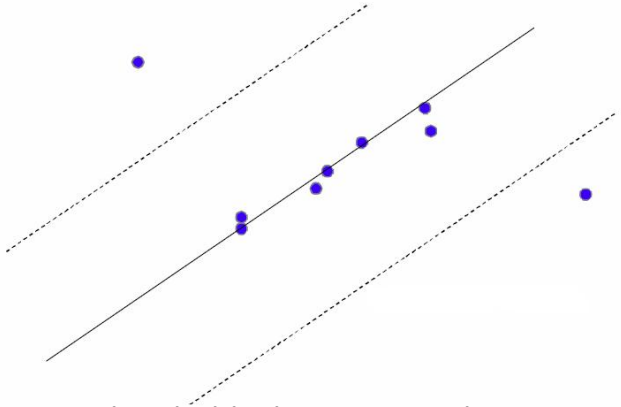
Random Sample Consensus

- One of the disadvantages of the RANSAC method is the uncertainty in the choice of the threshold.
- Both large and small thresholds lead to incorrect results.
- Let a set of points be given:

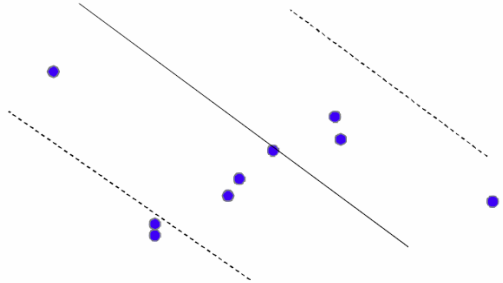


Random Sample Consensus

Threshold selection problem



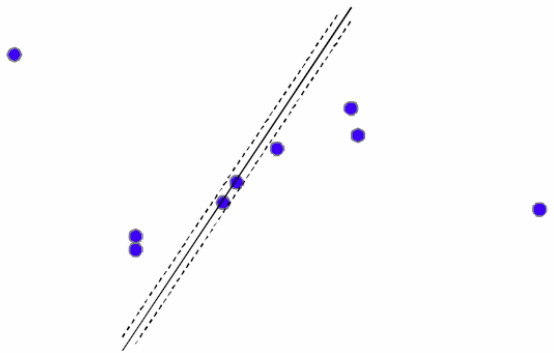
Big threshold. The correct solution.



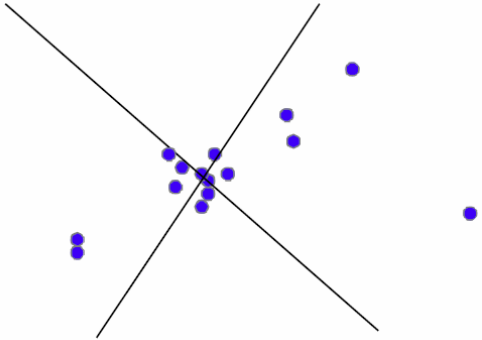
Big threshold. Incorrect solution, equivalent to correct.

Random Sample Consensus

Threshold selection problem



Small threshold. Incorrect solution.



Least Median Squares (LMS) problem. The median error is the same for both solutions

Random Sample Consensus

To assess the degree of agreement between hypotheses, consider the hypothesis evaluation function:

3. **M-SAC (M-estimator Sample Consensus)**

$$R(\Theta) = \sum_i p(\varepsilon_i(\Theta)^2), p(\varepsilon_i(\Theta)^2) = \begin{cases} \varepsilon_i^2, \text{if } |\varepsilon_i| \leq T \\ T^2, \text{if } |\varepsilon_i| > T \end{cases}, i = \overline{1,n}$$

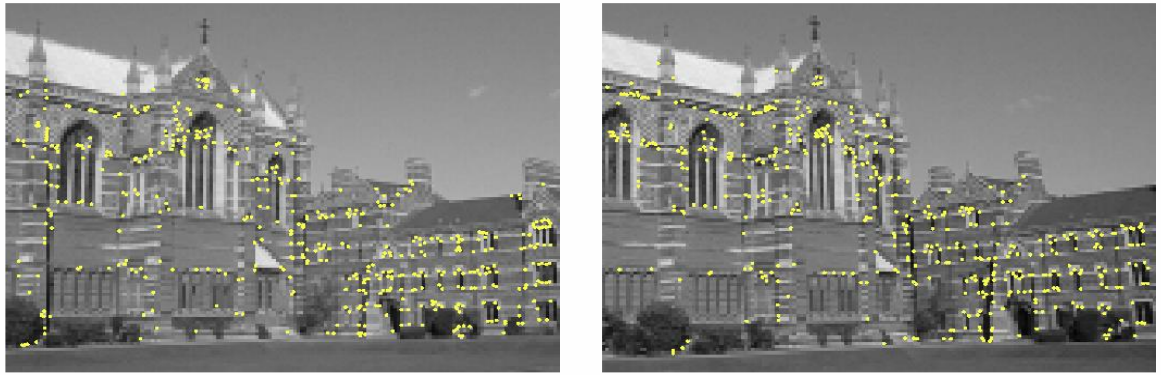
which is similar to the RANSAC function except for the modification of the probability function.

This method gives a more accurate estimate without increasing computational complexity and guarantees the correct solution.

Random Sample Consensus

ITMO

An example of using RANSAC: matching the same feature points.



When matching singular points by descriptors, quite a lot of false pairs will be detected.

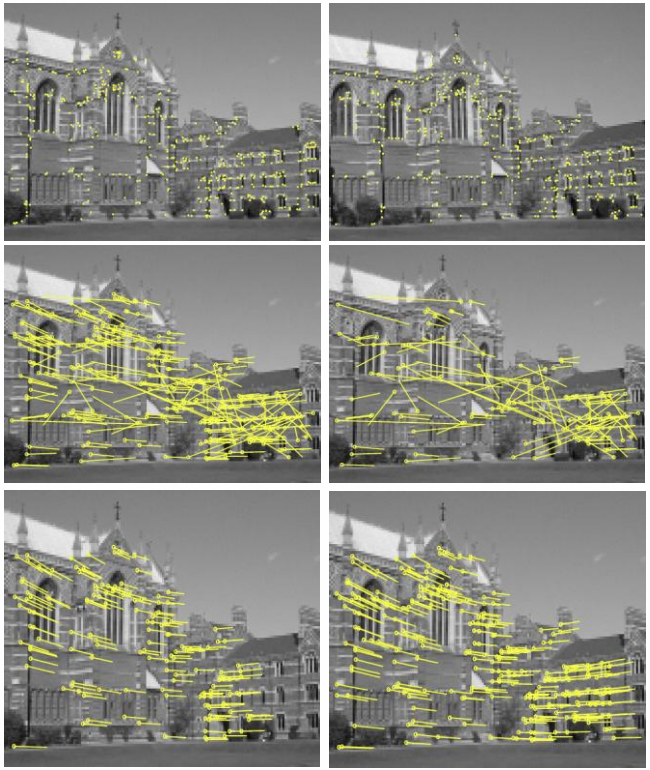
Random Sample Consensus

ITMO

Matching algorithm with using RANSAC:

1. Pairs of points $\{x, x'\}$ are given on the images I and I'.
2. Calculation of the transformation model T by key points between images I and I'.
   - Using the RANSAC schema to build a model T in sets of points.
3. Filtering outliers in $\{x, x'\}$.
4. Refinement of the model by the remaining points.
   - Or by the iterative least squares method;
   - Or by nonlinear minimization.

Random Sample Consensus



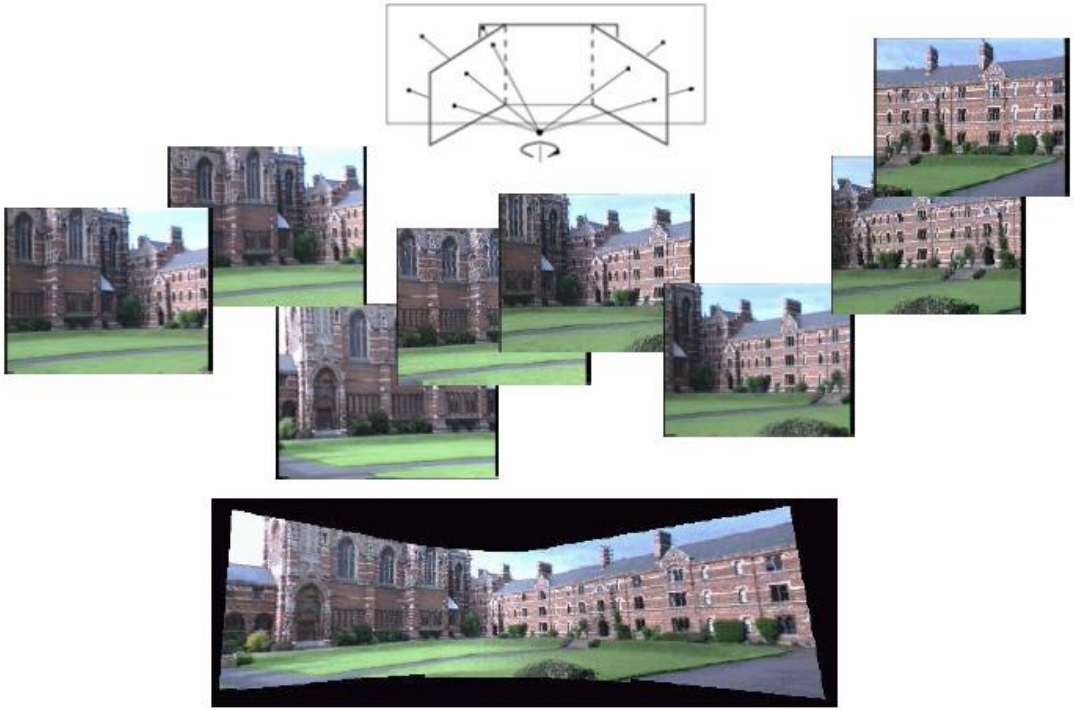
500 feature points were found on two images

Of these, 117 outliers and 268 matches

After filtering, 151 good matches were selected.

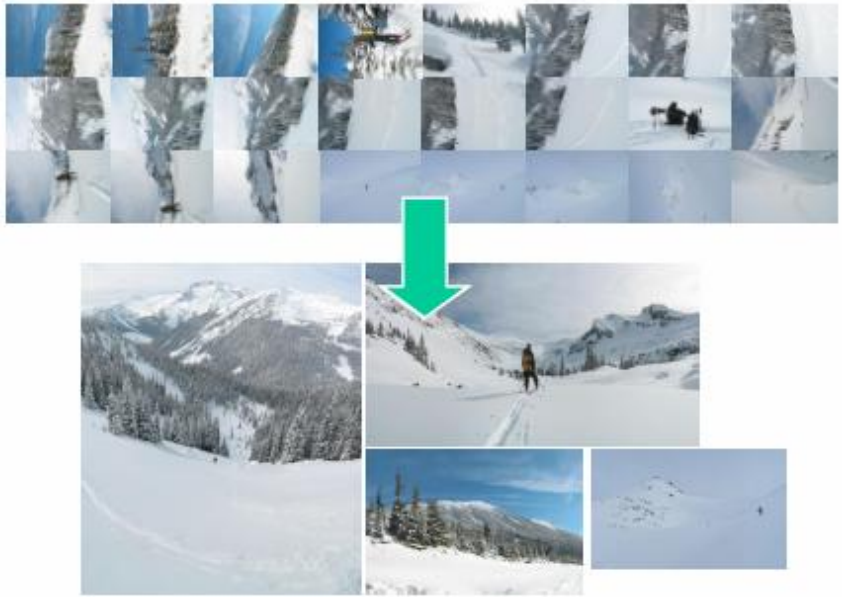
Random Sample Consensus

Use case: building panoramas from a set of photos.



Random Sample Consensus

Use case: building a panorama from an unordered set of photos, so it is necessary to determine which of them belong to one image and which to another.



Random Sample Consensus

RANSAC advantages:

- A simple and general method applicable to a variety of tasks;
- Works well in practice;
- Able to give a reliable estimate of the model parameters, that is, to estimate the model parameters with high accuracy, even if there are a significant number of outliers in the original dataset.

Random Sample Consensus

ITMO

RANSAC disadvantages:

- Many customizable parameters;
- It is not always possible to estimate the parameters well for the minimum sample;
- Sometimes it takes too many iterations;
- Does not work at very high outliers' ratio;
- Method uses uniform probability density distribution function for sampling;
- The absence of an upper bound on the time required to calculate the parameters of the model;
- The RANSAC method can only define one model for a given dataset. As with any single model approach, there is the following problem: when there are two (or more) models in the source data, RANSAC may not find one.

Test

Lecture 5-6 Test

Please scan the code to start the test

THANK YOU
FOR YOUR TIME!

ITMO more than a UNIVERSITY

Andrei Zhdanov
adzhdanov@itmo.ru