Image key points detection and matching

Mikhail V. Medvedev
Technical Cybernetics and Computer Science Department
Kazan National Research Technical University
Kazan, Russia
mmedv@mail.ru

Mikhail P. Shleymovich
Technical Cybernetics and Computer Science Department
Kazan National Research Technical University
Kazan, Russia
shlch@mail.ru

***Abstract*—In this article existing key points detection and matching methods are observed. The new wavelet transformation based key point detection algorithm is proposed and the descriptor creation is implemented.**

***Keywords*—key points, descriptors, SIFT, SURF, wavelet transform.**

I. Introduction

Nowadays the information technology based on artificial intelligence develops rapidly. Typically database with sample based retrieval becomes the major component of such intelligent systems. Biometrical identification systems, image databases, video monitoring systems, geoinformation systems, video tracking and many other systems can be considered as an example of intelligent systems with such databases.

For intelligent systems database retrieval the sample of data is defined, major characteristics are extracted and then the objects with similar characteristics are found in the database. In many cases images become the database objects. So we need some mechanism of characteristics extraction and their following comparison for finding identical or similar objects.

At the same time the intelligent systems of object 3D reconstruction are widely spread. Such systems can be used in robotic technology, architecture, tourism and other spheres. There are two major approaches to the 3D reconstruction problem solving: active and passive methods. In active methods depth sensors are used. They should be attached to the object directly, but in many cases this is impossible because of inaccessibility of an object. Such systems become very complex and demand additional equipment.

In passive method case photo camera is used as the sensor. Camera gets photos of an object for different points of views. It is not necessary to use depth sensors in this approach, and that's why it can be applied in any cases under all conditions. However, the object reconstruction accuracy substantively depends on the quality of collected images and the reconstruction algorithm. The first step of such an algorithm is to compare the images and identify the same key points for the further 3D reconstruction scheme evaluation. For solving such problems we need a computationally simple mechanism for image comparison and their similarity finding. The key point based description of and object is not very complex and rather reliable, that's why it can be used in object identifying tasks.

So we can see that the problem of identifying the same object in different pictures becomes very actual.

Key points or salient points concern the major information about the image. They can be found in the areas, where the brightness of the image pixels significantly changes. The human eye finds such points in the image automatically. These points can be characterized with two major features: the amount of key points mustn't be very big; their location mustn't change accord to the changing of the image size and image orientation; key point position must not depend on the illumination. In this paper we discuss the most popular SIFT and SURF method, and also present the new method based on wavelet transformation.

II. Existing key point detection methods

A. Harris Corner Detector

Harris corner detector [2] uses corners as the key points, because they are unique in two dimensions of the image and provide locally unique gradient patterns. They can be used on the image, when we have a small movement. The corner detection method looks at an image patch around an area centered at *(x,y)* and shifts it around by *(u,v)*. The method uses the gradients around this patch. The algorithm can be described in the following steps.

1. The calculation of the weighted sum of square difference between the original patch and the translated patch.

$$I(u+x,\ v+y)\approx I(u,v)+I_x(u,v)x+I_y(u,v)y \qquad (1)$$

2. Approximation by a Taylor expansion.

$$S(x,y)\approx\sum_u\sum_v w(u,v)\left(I_x(u,v)x+I_y(u,v)y\right)^2 \qquad (2)$$

3. Construction of weighted local gradients in matrix form, where I_x and I_y are partial derivatives of I in the x and y directions.

$$A=\sum_u\sum_v w(u,v)\begin{bmatrix} I_x^2 & I_xI_y \\ I_xI_y & I_y^2 \end{bmatrix}=\begin{bmatrix} \langle I_x^2\rangle & \langle I_xI_y\rangle \\ \langle I_xI_y\rangle & \langle I_y^2\rangle \end{bmatrix} \qquad (3)$$

4. Choosing the point with two "large" eigenvalues *a* and *b*, because a corner is characterized by a large variation of *S* in all directions of the vector [*x,y*].

5. If a=0, b=0, then the pixel (x,y) has no features of interest.

If a=0, b>>0, the point is counted as an edge.

If a>>0, b>>0, a corner is found.

However, the eigenvalues computation is computationally expensive, since it requires the computation of a square root. In the commercial robotics world, Harris corners are used by state-of-the-art positioning and location algorithms, various image-processing algorithms for asset tracking, visual odometry and image stabilization.

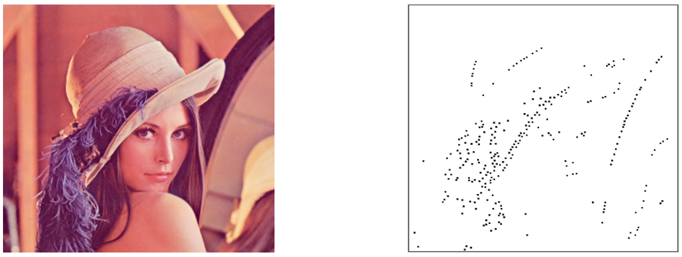


Fig. 1. Original image and Harris corner key points.

B. SIFT (Scale Invariant Feature Transform)

The most popular method for key point extraction is SIFT. Features are invariant to image scaling, translation, rotation, partially invariant to illumination changes, and affine transformations or 3D projection. It uses Differences of Gaussians (DoG) with fitted location, scale and ratio of principal curvatures for feature detection. These features are similar to neurons located in the brain's inferior temporal cortex, which is used for object recognition in primate vision. Features are efficiently detected through a staged filtering approach that identifies stable points in scale space. Image keys are created that allow for local geometric deformations by representing blurred image gradients in multiple orientation planes and at multiple scales.

The algorithm can be described in following steps.

1. The convolution of image and Gauss filter is made with different σ values.

$$\eta(x,y,\sigma)=\frac{1}{2\pi\sigma^2}\exp\left(\frac{-x^2-y^2}{2\sigma^2}\right) \qquad (4)$$

where k – scale coefficient, and * - convolution. The candidates for key points are formed by $D(x,y,\sigma)$ extremal points calculation.

$$D(x,y,\sigma)-(\eta(x,y,\sigma))*I(x,y) \qquad (5)$$

2. The points allocated along the edges are excluded with the help of Hesse matrix, calculated in candidate points of the previous step.

$$H=\begin{pmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{pmatrix} \qquad (6)$$

Because of the fact that the main curving along the edges have larger values, than in case of normal direction and the fact that Hesse matrix eigenvalues are proportionat to the main curving of the $D(x, y, \sigma)$, we need only to compare the Hesse matrix eigenvalues.

3. For the rotational invariance the orientation histogram is calculated over the key point neighbourhood with chosen step. For every σ the algorithm finds the orientation histogram extremal values.

$$\Theta(x,y)=arctg\frac{L(x,y+1)-L(x,y-1)}{L(x+1,y)-L(x-1,y)} \qquad (7)$$

$$L(x, y) = \eta(x, y, \sigma)*\ I(x, y) \qquad (8)$$

For invariant description of the key point the following algorithm is used.

1. Choosing the neighbourhood around the key point.
2. Calculation of the gradient value in the key point and its normalizing.

The neighbourhiood describing salient feature pattern is formed with the help of the replacemetn of a gradient vector by the number of its main components. It conduces to the salient feature number reduction and the affine transformation invariance is achieved, because the first main components are located along the main axes in computed gradients space. Fig. 2 illustrates the result of SIFT key point detection.

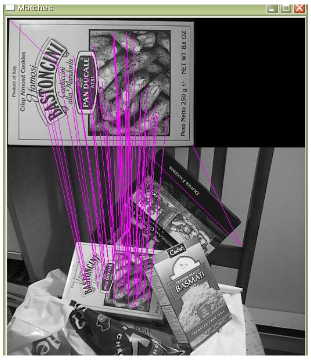


Fig. 2. Key points detection and matching using SIFT method

The major disanvantage of SIFT is that the algorithm takes too long to run and computationally expensive. In some cases it produces too few features for tracking.

C. SURF (Speeded Up Robust Features)

Another useful method of key point extraction is SURF (Speeded Up Robust Features) [1]. The descriptor comes in two variants, depending on whether rotation invariance is desired or not. The rotation invariant descriptor first assigns an orientation to the descriptor and then defines the descriptor within an oriented square. The other version, called U-SURF, for Upright-SURF, which is not rotation invariant, simply skips the orientation assignment phase. In this method the search of key point is made with the help of Hesse matrix.

$$H(f(x,y))=\begin{bmatrix}\frac{\partial^2 f}{\partial x^2} & \frac{\partial^2 f}{\partial x \partial y}\\ \frac{\partial^2 f}{\partial x \partial y} & \frac{\partial^2 f}{\partial y^2}\end{bmatrix}, \qquad (9)$$

$$detH=\frac{\partial^2 f}{\partial x^2}\frac{\partial^2 f}{\partial y^2}-\left(\frac{\partial^2 f}{\partial x \partial y}\right)^2.$$

The Hessian is based on LoG (Laplasian of Gaussian) using the convolution of pixels with filters. This approximation of Laplasian of Gaussian is called Fast-Hessian.

The Hessian reaches an extreme in the points of light intensity gradient maximum change, that's why it detects spots, angles and edges very well. Hessian is invariant to the rotation, but not scale-invariant. For this reason SURF uses different-scale filters for Hessian finding.

The maximum light intensity change direction and the scale formed by Hesse matrix coefficient are computed for each key point. The gradient value is calculated using Haar filter (Fig. 3).

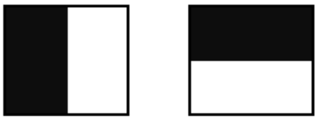


Fig. 3. Haar filters

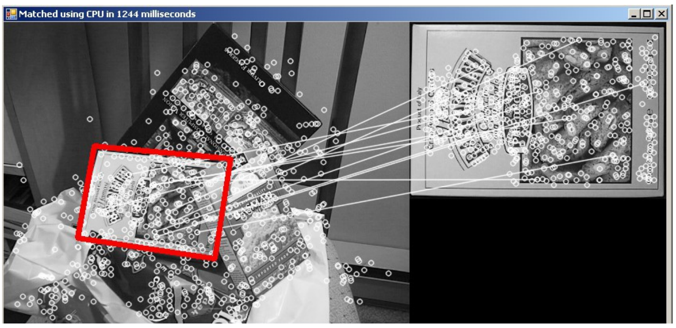
For effective Hesse and Haar filter computation image integral approximation is used.

$$II(x,y)=\sum_{i=0,j=0}^{i\le x,\, j\le y} I(i,j) \qquad (10)$$

where $I(i,j)$ — light intensity of image pixels.

After key points are found, SURF algorithm forms the descriptors. Descriptor consists of 64 or 128 numbers of for each key point. These numbers display a fluctuation of a gradient near a key point. The fact that a key point is a maximum of Hessian guarantees the existence of the regions with different gradients [1]. Fig. 4 illustrates the results of SURF key point detection.

The rotation invariance is achieved, because gradient fluctuations are calculated by the gradient direction over the neighborhood of a key point. The scale invariance is achieved by the fact that the size of the region for descriptor calculation is defined by the Hesse matrix scale. Gradient fluctuations are computed using Haar filter.



Matched using CPU in 1244 milliseconds

Fig. 4. SURF method key points detection.

SURF approximated, and even outperformed, previously proposed schemes with respect to repeatability, distinctiveness, and robustness. SURF also computed and compared much faster than other schemes, allowing features to be quickly extracted and compared. But for some classes of images with homogeneous texture it shows low level of key points matching precision.

III. Key points descriptors

For detected features matching we need key points descriptors. Key point descriptor is a numerical features vector of the key points neighborhood.

$$D(x)=[f_1(w(x))...f_n(w(x))] \qquad (11)$$

Feature descriptors are used for making the decision of images identity. The simplest descriptor is a key point neighborhood itself.

The major property of any feature matching algorithm is distortion varieties, which an algorithm can manage with. The following distortions usually are considered:

1) scale change (digital and optical zoom, movable cameras etc.);

2) image rotating (camera rotating over the object, object rotating over the camera);

3) luminance variance.

A. Scale change invariance.

While using scale-space feature detector it can be possible to achieve scale change invariance. Before descriptor calculation normalizing is held according to feature local scale. For example, for the scale-space coefficient of 2 we need to scale the feature neighborhood with the same value of scale coefficient.

If descriptor consists of equations only with normalized differential coefficients, space scaling operation is not necessary. It is sufficient to calculate differential coefficients for the scale associated with the feature.

B. Rotating invariance.

The simplest way to achieve rotating invariance is to use descriptors formed of rotating invariant components.

The major disadvantage of such an approach lies in the fact that it is impossible to use components with rotating dependence, but the amount of rotating invariant components is restricted.

The second way to achieve the rotating invariance is previous key point neighborhood normalizing for rotate compensation. For key point neighborhood normalizing we need feature orientation estimation. There are a lot of feature local orientation estimation methods, but all of them are connected with feature neighborhood gradient direction calculation. For example, in SIFT method the rotation invariance is achieved as follows.

1) All gradient directions angles from 0 to 360 degrees are divided into 36 equal parts. Every part is associated with a histogram column.

2) For every point from the neighborhood a phase and a vector magnitude are calculated.

$$grad(x_0,\delta)=(L_{x,norm}(x_0,\delta)L_{y,norm}(x_0,\delta)) \quad (12)$$

$$\Theta=L_{y,norm}(x_0,\delta)/L_{x,norm}(x_0,\delta) \quad (13)$$

$$A=|grad(x_0,\delta)| \quad (14)$$

$$H[i_\Theta]=H[i_\Theta]+Aw \quad (15)$$

where i – index of gradient phase cell, w – weight of a point. It can be possible to use the simplest weight of 1 or use Gaussian with the center in point *a*.

3) After that for every key point neighborhood direction $\varphi=i*10^o$ is chosen, where i is index of maximum from histogram elements. After orientation calculation the normalizing procedure is produced. A key point neighborhood rotates over the neighborhood center. Unfortunately, for some features orientation becomes wrong, and that descriptors cannot be used in further comparison. For every point from the neighborhood a phase and a vector magnitude are calculated.

C. Luminance invariance

For luminance invariance measurement we need the model of image luminance. Usually an affine model is used. It considers the luminance of the pixels changes according to the rule:

$$I_L=a*I(x)+b \quad (16)$$

This luminance model doesn't conform to real actuality correctly, and the luminance processes are much more complex, but it is sufficient for small local regions luminance representation.

According to affine luminance model to avoid luminance influence on pixels values in the key point neighborhood.

$$I_{mean}(w(x))=I(w(x))-mean(I(w(x))) \quad (17)$$

$$I_{result}(w(x))=I_{mean}(w(x))/std(I(w(x))) \quad (18)$$

where $mean(I(w(x)))$ and $std(I(w(x)))$ denote sample average and mean square deviation in neighborhood of w, $I_{mean}(w(x))$ – the translated neighborhood and $I_{result}(w(x))$ - the resulting neighborhood, which must be used for luminance invariance calculation.

IV. Wavelet Transformation Based Key Point Detection

Another way of key points extraction is using of discrete wavelet transformation. Discrete wavelet transformation produces a row of image approximations. For image processing Mall algorithm is used. The initial image is divided into two parts: high frequency part (details with sharp luminance differences) and low-frequency part (smoothed scale down copy of the original image). Two filters are applied to the image to form the result. It is an iterative process with the scaled down image copy as the input.

A. Descrete Wavelet Transformation

Wavelet transformation is rather new direction of theory and technique of signal, image and time series processing. It has been discovered at the end of the XX century and now is used in different spheres of computer science such as signal filtration, image compression, pattern recognition etc. The reason of its widely spread using is based on wavelet transformation ability of exploring inhomogeneous process structure.

Discrete wavelet transformation produces a row of image approximations. For image processing Mall algorithm is used (Fig. 5). The initial image is divided into two parts: high frequency part (details with sharp luminance differences) and low-frequency part (smoothed scale down copy of the original image). Two filters are applied to the image to form the result. It is an iterative process with the scaled down image copy as the input. [5]

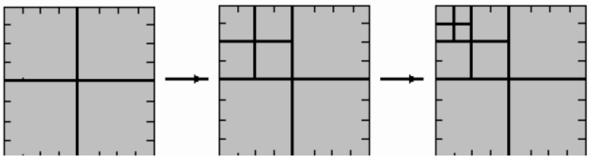


Fig. 5. Mall algorithm

B. Key Points Detection

Wavelet image transformation can be used for key points detection. The saliency of the key point is formed by the weights of wavelet coefficients. [3]

In the method proposed in this article the key point extraction algorithm calculates the weight of every image pixel using the following equation:

$$C_i(f(x,y))=\sqrt{dh_i^2(x,y)+dv_i^2(x,y)+dd_i^2(x,y)} \qquad (19)$$

where $C_i(f(x,y))$ – the weight of the point on the level i of detalization, $dh_i(x,y)$ – horizontal coefficient on the level i, $dv_i(x,y)$ – vertucal coefficient on the level i, $dd_i(x,y)$ – diagonal coefficient on the level i. At the first step all weights are equal to zero. Then wavelet transformation is carried out until it reaches the level n. Each rather large wavelet coefficient denotes a region with a key point of the image. Weight is calculated using the following formula (19) then recursive branch is exercised.

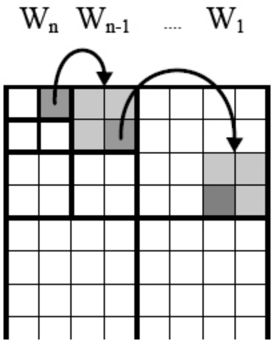


Fig. 6. Key point weight calculation.

This algorithm repeats for all decomposition levels. The final value of the weight of pixel is formed by the wavelet coefficients of previous levels. After key points sorting the point larger than the desired threshold are chosen.

In the image on Fig. 1 and Fig. 4 the key points are detected using Harris detector and wavelet based method. In case of Harris detector key points are located in the corners and have little dispersion over the image. In the case of wavelet based method the image is covered with key points proportionally.

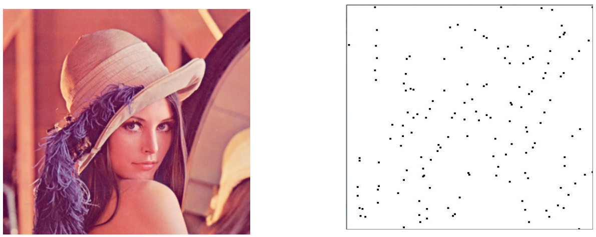


Fig. 7. Original image and wavelet based key points.

C. Key Points Descriptor

For points matching we need to create a descriptor, which can describe the point and tell the differences between them. SIFT and SURF descriptors are based on pixels luminance over the region in the point neighborhood. In this paper we offer using the wavelet coefficient, which are produced form the luminance are stable for various luminance changes.

The descriptor is formed from the pixel wavelet coefficients, received from the wavelet decomposition of key point neighborhood. Each neighbor is characterized by 4 wavelet coefficients: the base coefficient, horizontal, diagonal and vertical ones. The dimension of in the descriptor is fixed on 4*16. The size of the neighborhood region depends on the size of image and the wavelet decomposition level. The experiments have shown that it is possible to use the depth of wavelet decomposition equal or greater than 3. For example, for the region size of 64 neighbors we need the 3rd decomposition level to form the descriptor of 4*8 and in the case of the dimensionality of neighborhood increase we should also increase the level of transformation. Experiments have shown that such an increase takes more time for computation, but in some cases it allows to avoid matching errors.

Fig. 8 illustrates wavelet based key point extraction and matching result. The software application was implemented with the use of C# programming language in Microsoft Visual Studio 2008.

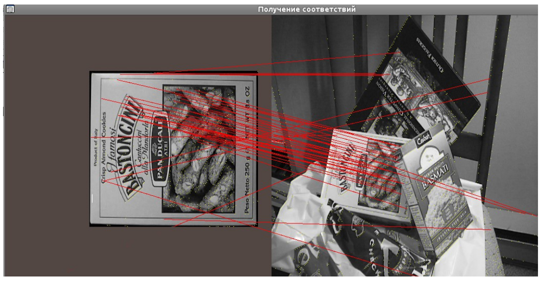


Fig. 8. Key points detection and matching using wavelet based method

D. Segmentation using wavelet key points

Wavelet based key points detection algorithm can be used for image segmentation. On the the first step wavelet based key points retrieval is carried out. All key points are marked with black color, and other points of the image are marked with white color. Then connected components retrieval algorithm is applied. This algorithm considers the black color of key points as the background and the white color of ordinary points as objects on the foreground. After that key points spaceless sequence surrounded pixels joining is produced. For different segments marking the algorithm of connected components line-by-line marking is used.

Described above segmentation algorithm is computationally efficient. It can be used in systems with restricted resources. The computational efficiency is reached because of the fact that this algorithm finds only large objects on the image. Wavelet transformation explorers an image on different scales and finds only the points, which have saliency on all levels. This property is owned by the points with major luminance change. In resulting image we “loose” all little components and see only the larger ones. Fig. 9 shows the segmentation results produced. The software application was implemented with the use of C# programming language in Microsoft Visual Studio 2008 and was evaluated on mobile devices with restricted resources. (The photos are made of the mobile device emulator on PC.)

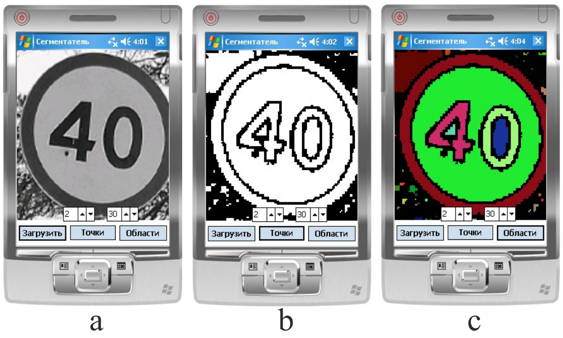


Fig. 9. Segmentation result on mobile device: a –original image, b – wavelet based key points, c – segmented image.

E. *Future Work*

Future work will be referred to the improvement of the proposed method of wavelet based key point detection. It is necessary to increase the accuracy of key point matching and to decrease the computational complexity of descriptor finding. For another thing we need more experiment results for detecting rotation, scale and luminance invariance of the proposed method.

References

[1] H. Bay, A. Ess, T. Tuytelaars, and L. Gool, "SURF: Speeded Up Robust Features," *Computer Vision and Image Understanding*, vol. 110, no. 3, pp. 346-359, 2008.

[2] C. Harris and M. Stephens, "A Combined Corner and Edge Detector," in

[3] Proceedings of the 4th Alvey Vision Conference, 1988, pp. 147-151.

[4] E. Loupias, N. Sebe. “Wavelet-based Salient Points: Applications to Image Retrieval Using Color and Texture Features.” Lecture Notes in Computer Science.

[5] D. G. Lowe, "Object Recognition from Local Scale-Invariant Features," in Seventh IEEE International Conference on Computer Vision, vol. 2, Kerkyra, Greece, 1999, pp. 1150-1157.

[6] E.J. Stollnitz, T. DeRose, and D. Salesin, "Wavelets for computer graphics - theory and applications", ;presented at The Morgan Kaufmann series in computer graphics and geometric modeling, 1996, pp.1-245.