THE ANNALS OF "DUNAREA DE JOS" UNIVERSITY OF GALATI
FASCICLE XIV MECHANICHAL ENGINEERING, ISSN 1224-5615
2012

ANALYSIS OF THE SIFT FEATURE DESCRIPTORS ALGHORITHM IMPLEMENTATION USED FOR OBJECT PIECES RECOGNITION

Drd. Inf. Catalin Constantin Moldovan
Prof. Dr. Eng. Ionel Staretu
"Transilvania" University of Brasov

ABSTRACT

Recognition algorithms based on local features such as: are widely used in a large number of computer vision applications object tracking, object reconstruction from images, detection and recognition of objects and people .In the last years, this research domain gains a lot of attention from the computer science research community. In this work, a study of feature description SIFT algorithm is presented along with a comparative analysis of the total running time and the best performance for future applications in industrial engineering.

KEYWORDS: object recognition, SIFT, local descriptors vibration

1. INTRODUCTION

Local features algorithms are widely used in computer vision application and researches. With the help of these types of algorithms some tasks such as recognition of object class, pattern matching, tracking or tridimensional reconstruction can be easily performed and in a repeatable manner. One of the most important aspects of the local features is that the algorithms will be invariant to various light conditions, object orientation or changing the viewport. The algorithms based on local features must perform in a robust way even though the objects are slightly changed. The main idea behind algorithms based on local descriptors is to detect image regions covariant to a class of transformation. Further these regions will be used to compute invariant descriptors.

In this paper, the evaluation of the SIFT descriptor algorithm implementation in different computer languages is performed by matching and recognizing the same object observed under viewing state modified and in different scenes.

In the specialized literature of this specialty, a large number of possible descriptors exist. In this paper some implementation of SIFT algorithm performance will be measured.

2. THE SIFT ALGHORITHM

Scale-invariant feature transform (or SIFT) is an algorithm used in computer vision applications to detect and describe local features in images. The algorithm was first published by David Lowe in the year 1999.

The research carred out by Lowe in 1999 started by identifying the location of image of scale space that is invariant to translation, scale and rotation and is minimalyl affected by noise and small modification.

2.1. Localization of key points

By using the assumptions from the research made by [2] that demonstrates the Gaussian kernel and its derivates are the only possible way of analysis of scale space, in [3] the author achieves rotation invariance at a high level of efficiency by selecting key locations at maxima and minima of a difference of Gaussian

function applied in scale space. The high level of efficiency was achieved by building an image pyramid with re-sampling between each level. In this way key points at regions and invariant to scale variations key points were located. Scale space of an image is defined as a function $S(x,y,\sigma)=G(x,y,\sigma)*I(x,y)$. Lowe used the Gaussian function in two directions, horizontally and vertically:

$$g(x)=\frac{1}{\sqrt{2\pi\sigma}}e^{-\frac{x^2}{2\sigma^2}} \quad (1)$$

To localize key, smoothing operations are performed using $\sigma=\sqrt{2}$ that was approximated using 1D kernel with 7 sample points.

The convolution operator is applied between the input image and the Gaussian operator having the parameter $\sigma=\sqrt{2}$.

The result will be Image A'. Incrementing the smoothing means to apply again the convolution operation on the Image A' also with the $\sigma=\sqrt{2}$. The result will be image A''.

The effective smoothness of image A'' will be $\sigma=2$.

The difference of the Gaussian function is obtained by calculating the difference of A'' – A'. This will result in a ratio of $\frac{2}{\sqrt{2}}=\sqrt{2}$ between the Gaussian of the image A'' and the image A'.

To go to the next pyramid level, the image A'' is resampled using bilinear filtering [4] on X and Y directions.

The bilinear interpolation is a extension of linear interpolation for interpolation functions of two variables (example: $f(x,y)$ on a regular bi-dimensional grid. The main idea behind bi-linear interpolation is to perform linear interpolation in the X direction and then again perform linear interpolation on the Y direction.

Trying to find the function f at the point $A''(x,y)$, and knowing the values of the function $f(x,y)$ at the points (x_1,y_1), (x_1,y_2), (x_2,y_1), (x_2,y_2).

The linear interpolation on the X direction will result:

$$f(x,y_1)\approx\frac{x_2-x}{x_2-x_1}f(x_1,y_1)+\frac{x-x_1}{x_2-x_1}f(x_2,y_1);$$

$$f(x,y_2)\approx\frac{x_2-x}{x_2-x_1}f(x_1,y_2)+\frac{x-x_1}{x_2-x_1}f(x_2,y_2).$$

The interpolation on the Y direction:

$$f(x,y)\approx\frac{y_2-y}{y_2-y_1}f(x,y_1)+\frac{x-y_1}{y_2-y_1}f(x,y_2)$$

and using $f(x,y_1)$ and $f(x,y_2)$ in the above $f(x,y)$ will result:

$$f(x,y)\approx \frac{f(x_1,y_1)}{(x_2-x_1)(y_2-y_1)}(x_2-x)(y_2-y)+\frac{f(x_2,y_1)}{(x_2-x_1)(y_2-y_1)}(x-x_1)(y_2-y)+\frac{f(x_1,y_2)}{(x_2-x_1)(y_2-y_1)}(x_2-x)(y-y_1)+\frac{f(x_2,y_2)}{(x_2-x_1)(y_2-y_1)}(x-x_1)(y-y_1).$$

The minimum and maximum of this function are determined by comparing each pixel in the pyramid with the neighbors. A pixel is compared at the same level of the pyramid with 8 neighbors.

If a maxima or minima are at the same level, then the closest pixel location is calculated to the next level of the pyramid, of course, having in mind the re-sampling parameter. If the pixel is the maximum or minimum (to the closest pixel and the 8 neighbors), the test is repeated at the above level.

2.2.Stabilizing the key points, removing bad key points

To stabilize the key points, for the image A', at each level of the pyramid, the calculations are made to extract image gradients and orientation. For each pixel $P_{i,j}$ the gradient magnitude $G_{i,j}$ and orientation $O_{i,j}$ are calculated using pixel differences:

$$G_{i,j}=\sqrt{(P_{i,j}-P_{i+1,j})^2+(P_{i,j}-P_{i,j+1})^2}; \quad (2)$$

$$O_{i,j}=a\tan 2(P_{i,j}-P_{i+1,j},P_{i,j+1}-P_{i,j}).$$

The two argument function atan2 is a variation of the arctangent function. For any

real number (e.g., floating point) arguments x and y not both equal to zero, atan(y, x) is the angle in radians between the positive x-axis of a plane and the point given by the coordinates (x, y) on it. The angle is positive for counter-clockwise angles (upper half-plane, y > 0), and negative for clockwise angles (lower half-plane, y < 0).

From [3], it can be observed that by, enhancing the threshold of $G_{i,j}$ by a value of 0.1 times the maximum of possible gradient value, the algorithm will be robust to changes in illumination. In this way, even though the changes of illumination for a surface with 3D relief may drive to changes in gradient magnitude, the orientation will not be affected.

For each key location, a canonical orientation is assigned, in this way; the images will be invariant to rotation. To maximize the function which determine if a pixel is good or not, the orientation is determined by the peak of the histogram created with the local image gradient orientations.

The orientation histogram is created using the Gaussian weighted window with $\sigma = \sqrt{2}$ parameter equal to three times the smoothing scale.

These weights will be multiplied with the threshold of the gradient values and calculated in the location corresponding to orientation $O_{i,j}$ for each pixel $P_{i,j}$. The histogram will have 36 bins; in this way it will cover the 360-degree range of rotations.

Also [3] in his research determined that the stability of the resulting keys can be tested by subjecting natural images to affine projections, contrast and brightness changes and addition of noise. The location of the key points detected in the first image can be deducted in the transformed image if the transformations are known.

In Figure 1, the numbers of keys detected using the algorithm SIFT over a two octave range are displayed:

Fig. 1. Key points detected

2.3. Generated features

Having all the necessary elements (stable location, scale and orientation for each key), it is possible to describe the image regions in a way that is invariant to image transformation.

To index the key points the SIFT keys for sample images are stored and then, the stored keys are used to match the keys in the new images. This can raise problem of performance because of the high complexity of the search algorithm but in the initial SIFT implementation from [3] it was observed that by using k-d tree search algorithm (best bin first search method described in [1]) the nearest neighbors were detected with a high probability and with a limited amount of computations.

The affine invariance is determined by using an affine transformation of a model point as it is described in [3].

The affine transformation of a model point $[x \quad y]^T$ to an image point $[u \quad v]^T$ can be written as:

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} m_1 & m_2 \\ m_3 & m_4 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix} \quad (3)$$

where the model translation is $[t_x \quad t_y]^T$ and the affine rotation, scale and stretch are represented by m_i parameters. The equation can be written as:

$$\begin{bmatrix} x & y & 0 & 0 & 1 & 0 \\ 0 & 0 & x & y & 0 & 1 \\ & & \dots & \dots & & \\ & & \dots & \dots & & \end{bmatrix} \begin{bmatrix} m_1 \\ m_2 \\ m_3 \\ m_4 \\ t_x \\ t_y \end{bmatrix} = \begin{bmatrix} u \\ v \\ . \\ . \end{bmatrix} \quad (4)$$

This system can be reduced at $Ax = B$ where the parameter X can be determined using least-square solution by solving the normal equation:

$$x = [A^T A]^{-1} A^T b \quad (5)$$

3. CONCLUSIONS AND EXPERIMENT

In this paper a .Net solution was created

based on the theory of the SIFT algorithm. The experiment made on this paper trained the SIFT algorithm for a industrial piece and recognized them in changes of: scale, rotation (Figure 2 and Figure 3), affine transformation and partial occlusion (Figure 4 and Figure 5).

Fig. 2. Rotation – vertically

Fig. 3. Rotation – horizontally

Fig. 4. Partial occlusion – lower part

Fig. 5. Partial occlusion – upper part

REFERENCES

[1] Beis, Jeff, and David G. Lowe, Shape indexing using approximate nearest-neighbour search in high-dimensional spaces, Conference on ComputerVision and PatternRecognition, Puerto Rico, pp. 1000–1006, 1997.

[2] Lindeberg, T., Scale-space theory: A basic tool for analysing structures at different scales, Journal of Applied Statistics, 21, 2 pp. 224–270, 1994.

[3] Lowe, David G., Object recognition from local scaleinvariant features, International Conference on Computer Vision, Corfu, Greece, pp. 1150–1157, 1999.

[4] *** : http://en.wikipedia.org/ wiki/ Bilinear_interpolation, accessed on September 2012.