IOP Conf. Series: Materials Science and Engineering **439** (2018) 032104 doi:10.1088/1757-899X/439/3/032104

# Clothing image feature extraction based on SURF

**L L Chen[1], R P Han[2*], Q C Meng[1]**

[1]School of Information Engineering, Beijing Institute of Fashion Technology, Beijing, 100029, China

[2]Chinese Fashion & Technology Research Institute, Beijing Institute of Fashion Technology, Beijing, 100029, China

[*]Corresponding author's e-mail: gxyhrp@bift.edu.cn

**Abstract**. The SURF (Speeded Up Robust Features) is one of the most commonly used artificial feature extraction algorithms and has a good robustness. SURF is widely used in image processing and machine vision. This paper introduces the implementation of SURF in a more comprehensive way, and applies it to the feature extraction of clothing images. The clothing features extracted by using SURF can be applied in clothing classification, identification, retrieval and matching in combination with machine learning method. It can also be used as input sample of deep neural network to improve classification or recognition accuracy.

## 1. Introduction

Feature extraction is an important part of image processing, it is the basis for the follow-up, such as image recognition, retrieval and fusion. Scale Invariant Features Transform(SIFT) is an algorithm for image feature extraction and description. SIFT was proposed by David. G. Lowe[3] in 1999 and perfected in 2004. SIFT is considered to be more effective and commonly used feature extraction algorithm. Speeded Up Robust Features(SURF) is proposed by Herbert Bay et al.[5] in 2006, it's a novel scale- and rotation-invariant detector and descriptor. SURF inherits the advantages of SIFT and it also has good robustness. When the image has scale and rotation change or affine transformation, it can still get the features description effectively and steadily. Related experiments show that the SURF is about 3 times faster than SIFT at running speed, and its comprehensive performance is better than SIFT[2].

## 2. Features extraction by SURF

The process of features extraction by SURF is divided into three parts. The first part is to transform gray image into integral image and the SURF relies on integral image for conducting convolutions operation at faster speed. The second part is to build image pyramid by approximating the Hessian matrix. The third part is to obtain features descriptor.

### *2.1. Integral image*

Integral image allow for fast computation of box type convolution filters. The value $I(x,y)$ of pixel point $(x,y)$ in the integral image is the sum of the grayscale values of all the pixel points in the rectangular area formed from the upper left point of the image to the point $(x,y)$. Therefore, the sum $S$ of the grayscale values in a rectangular area of the image can be obtained from equation (1).

$$S = I(D) - I(B) - I(C) + I(A) \tag{1}$$

In formula (1), *I(A)*, *I(B)*, *I(C)*, and *I(D)* are the integral image values at the *A*, *B*, *C*, and *D*, respectively. The pixel point *A*, *B*, *C*, and *D* are the upper left, upper right, lower left, and lower right points of the rectangular area.

### *2.2. Hessian matrix approximation*

The Hessian matrix approximation is used to detect interest points. Its determinant value reflects the local information of the image. The Hessian matrix $H(x,y,\sigma)$ for the pixel point $(x,y)$ in the image is defined as follows:

$$H(x,y,\sigma) = \begin{bmatrix} L_{xx}(x,y,\sigma), L_{xy}(x,y,\sigma) \\ L_{xy}(x,y,\sigma), L_{yy}(x,y,\sigma) \end{bmatrix} \tag{2}$$

In formula (2), $L_{xx}(x,y,\sigma)$, $L_{yy}(x,y,\sigma)$ and $L_{xy}(x,y,\sigma)$ are the convolution results of image at pixel point $(x,y)$ with Gaussian second-order partial derivative in *x*, *y* and *xy* direction respectively. Let $D_{xx}(x,y,\sigma)$, $D_{yy}(x,y,\sigma)$ and $D_{xy}(x,y,\sigma)$ are the convolution results of image at pixel point $(x,y)$ with approximated Gaussian second-order partial derivative in *x*, *y* and *xy* direction respectively. And the approximated Gaussian second-order partial derivatives in *x*, *y* and *xy* direction are called box filters. Then the determinant of the Hessian matrix is simplified as follows[2]:

$$\begin{aligned} Det(H) &= L_{xx}L_{yy} - (L_{xy})^2 = D_{xx}\frac{L_{xx}}{D_{xx}}D_{yy}\frac{L_{yy}}{D_{yy}} - D_{xy}\frac{L_{xy}}{D_{xy}}D_{xy}\frac{L_{xy}}{D_{xy}} \\ &= D_{xx}D_{yy}\left(\frac{L_{xx}}{D_{xx}}\frac{L_{yy}}{D_{yy}}\right) - D_{xy}D_{xy}\left(\frac{L_{xy}}{D_{xy}}\right)^2 = A\left(\frac{L_{xx}}{D_{xx}}\frac{L_{yy}}{D_{yy}}\right) - B\left(\frac{L_{xy}}{D_{xy}}\right)^2 \\ &= (A - B(\frac{L_{xy}}{D_{xy}})^2(\frac{D_{xx}}{L_{xx}}\frac{D_{yy}}{L_{yy}}))(\frac{L_{xx}}{D_{xx}}\frac{L_{yy}}{D_{yy}}) = (A - BY)C \end{aligned} \tag{3}$$

In theory, for different $\sigma$ and the box filter size, *Y* is different. In order to simplify the operation, the value of *Y* is set to constant 0.9. Based on the above assumption, the formula (3) is further simplified to formula (4).

$$Det(H_{approx}) = D_{xx}D_{yy} - (0.9D_{xy})^2 \tag{4}$$

In formula (4), $Det(H_{approx})$ is the blob response of the pixel points. Using the box filters to traverse all the pixels in the image, we can obtain the filter response map at a certain scale.

Figures 1-4 demonstrate Gaussian function, its second-order partial derivatives and corresponding box filters. Figure 1 shows the 3D and 2D graphs of the Gauss function. In figures 2-4, the left are 3D graphs of the Gaussian second-order partial derivative in the *x*, *y* and *xy* direction, respectively. The middle are 2D graphs of the Gaussian second-order partial derivative in the *x*, *y* and *xy* direction, respectively, and the right are the box filters in the *x*, *y* and *xy* direction, respectively. The values in the black area, white area and gray area of the box filters are different. The size of the box filters here is 9 ×9 (the size of box filters will be described in detail in the next section). The Gaussian function with $\sigma$ = 1.2 and its second-order partial derivative function in different directions shown in figures 1-4 are drawn by using MATLAB.

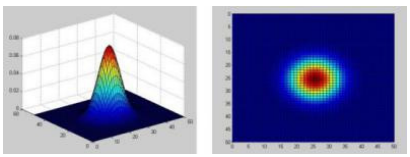


Figure 1. Left to right: the 3D and 2D graphs of Gauss's function.

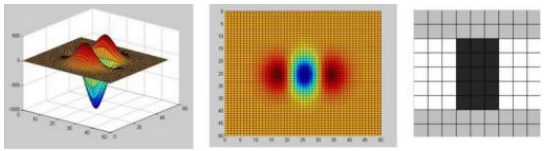


Figure 2. Left to right: the 3D graph, the 2D graph and box filter of the Gaussian second-order partial derivative in *x* direction.

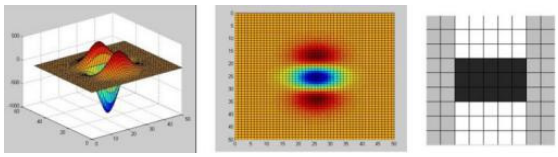


Figure 3. Left to right: the 3D graph, the 2D graph and box filter of the Gaussian second-order partial derivative in *y* direction.

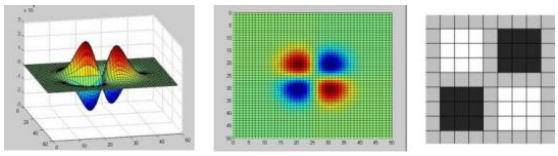


Figure 4. Left to right: the 3D graph, the 2D graph and box filter of the Gaussian second-order partial derivative in *xy* direction.

*2.3. Using box filters to establish scale space representation*

Scale space representation is the description of image at multiple scales. It is the basis for detecting invariant features. Koenderink[9], Lindeberg[10], Florace[11] and others used different mathematical methods to prove that the Gaussian kernel is the only transform kernel to realize scale transformation. In the SURF, scale space representation is usually implemented as an image pyramid.

The scale space is divided into many octaves. An octave represents a series of filter response maps obtained by convolving the same input image with filters of increasing size. Each octave is subdivided into a constant number of scale levels. In general, one octave contains four scale levels. The relationship between the octave index *o* and the scale level index *t*, the box filter response length *l*, and the box filter size length *L* is shown in formulas (5)-(7)[1]. Using formulas (5)-(7), box filter response length and box filter size length between different octaves and scale levels can be calculated. For example, the box filter response length for the 0th scale level of the 0th octave is 3, the box filter size length $L$ is 9, and its scale $s$ is 1.2; it is used as the lowest scale for computing the blob response maps. Based on formulas (5)-(7) , the box filter size length in the next scale levels can be computed as 15, 21, 27. If the size of the original image is still larger than the size of the box filter, a higher octave should be built. The number of detected interest points in the image will decay rapidly as the scale increases.

$$l = 2^{o+1}(t+1)+1 \tag{5}$$

$$L = 3\times l = 3\times(2^{o+1}(t+1)+1) \tag{6}$$

$$s = 1.2\times(L/9) \tag{7}$$

*2.4. The generation of features descriptor*

The generation of features descriptor is divided into three steps, the first step is to detect interest points, the second step is to assign the orientation of interest point, and the third step is to obtain features descriptor.

*2.4.1. Localization of interest point.* According to the formula (4), the blob response of the image can be obtained, and then the extreme points in the blob response can be determined using 3D non-

maximum suppression. These extreme points are candidate interest points. The 3D non-maximum suppression is to take the pixel point to be detected as the center point, and compare it's blob response value not only with the neighboring pixels' at the same scale but also with the neighboring pixels' in the same position at the upper and lower scales. When the blob response value of the center point is the maximum or the minimum, it can be used as a candidate interest point. Since the input image is a discrete one, the candidate interest point is not necessarily the true interest point. Therefore, interpolation operations should be applied in the scale space to find the position of the true interest points. To this end, the Taylor series is used to estimate Hessian determinant at interest point based on formula (8):

$$H(X_1) \approx H(X_0) + \frac{\partial H^T}{\partial X}(X_1 - X_0) + \frac{1}{2}(X_1 - X_0)^T \frac{\partial^2 H}{\partial X^2}(X_1 - X_0) \tag{8}$$

where $X_0=(x_0,y_0,s_0)^T$ gives the location information of the candidate interest point, and $X_1=(x_1,y_1,s_1)^T$ gives the location information of the interest point. The derivative $\frac{\partial H}{\partial X}$ at the interest point is computed approximately by using $\left.\frac{\partial H}{\partial X}\right|_{X=X_1} \approx \frac{\partial H}{\partial X} + \frac{\partial^2 H}{\partial X^2}(X_1 - X_0) = 0$, so the offset $\Delta X$ can be calculated based on the formula (9).

$$\Delta X = X_1 - X_0 = -\left(\frac{\partial^2 H}{\partial X^2}\right)^{-1} \frac{\partial H}{\partial X} \tag{9}$$

where $\Delta X$ is the offset of rows, columns and scales. When $\Delta X$ is less than the threshold value 0.5, $X_1$ represents a interest point; if there is a non-conformity, the new candidate interest point that is formed by adding $X_0$ and the integer part of $\Delta X$ together is judged again until the value of the offset is less than 0.5; Putting $\Delta X$ into the formula (8), we can get the blob response value of the interest point. If the offset is still bigger than 0.5 after the specified number of iterations, the candidate interest point is abandoned.

*2.4.2. Orientation assignment of interest point.* In order to ensure the rotation-invariance, the orientation of the interest point needs to be found. First, the circle with a radius of 6*s* around the interest point should be extracted, *s* represents the scale at which the interest point is detected. Second, a sliding fan window of size π/3 in the circle is selected to calculate the Haar wavelet response *dx*, *dy* in *x*-direction and *y*-direction respectively. Then the two summed response $\sum dx$ and $\sum dy$ are obtained to get the local orientation vector $(m_\omega,\theta_\omega)$ based on formula (10) and formula (11). The figure 5 shows the Haar wavelet filters. Third, the sliding fan window slides in every step of 0.2 radian, and the Haar wavelet response in the new region are obtained to get a new local orientation vector. Finally, the longest one ( $max\{m_\omega\}$) within all the vectors defines the orientation angle $\theta$ of the interest point (see formula (12)). Figure 6 shows the process of obtaining the orientation vector by taking Haar wavelet responses within three areas as an example.

$$m_\omega = \sum_\omega dx + \sum_\omega dy \tag{10}$$

$$\theta_\omega = \arctan\left[\frac{\sum_\omega dx}{\sum_\omega dy}\right] \tag{11}$$

$$\theta = \theta_\omega \,|_{\max\{m_\omega\}} \tag{12}$$

Figure 5. Haar wavelet filters.

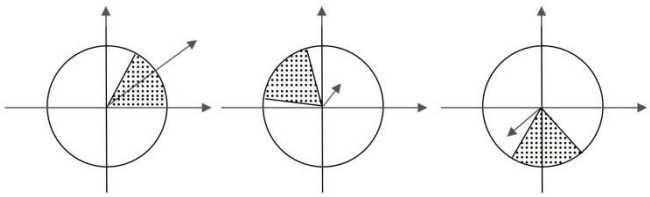


Figure 6. Orientation assignment map of interest point.

*2.4.3. Feature descriptor.* The first step for establishing the descriptor is to extract a rectangular area which has a size of $20s \times 20s$ and is centered on the interest point and oriented along the orientation of the interest point. The second step is to split the rectangular area into 16 parts, with each part having the size of $5s \times 5s$. In the third step, the $5s \times 5s$ area is divided into 25 samples. In the fourth step, not only the Haar wavelet response $\sum dx'$ and $\sum |dx'|$ along the orientation, but also the Haar wavelet response $\sum dy'$ and $\sum |dy'|$ along the direction perpendicular to the orientation are calculated at these 25 samples. These four responses($\sum dx'$,$\sum |dx'|$, $\sum dy'$,$\sum |dy'|$) constitute the feature descriptor of the $5s \times 5s$ area. Thus, the feature descriptor for each interest point is a 64-dimensional vector. In figure 7, the left figure shows the rectangular area centered on the interest point, and the right figure shows the Haar wavelet response within each area of 5s×5s.

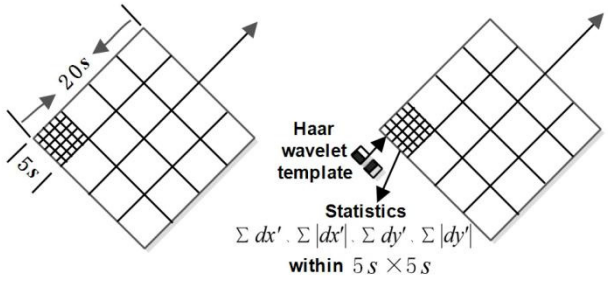
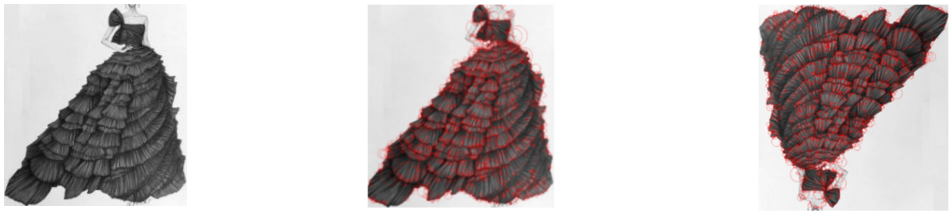



Figure 7. Feature vector description of interest point.

## 3. Features extraction of clothing images

Feature extraction of clothing images is implemented based on SURF algorithm mentioned above. The features extraction effects are shown in figures 8-10. Figures 8-10 (a) are the original costume image downloaded from https://www.deepfashion.cn/. Figures 8-10 (b) show detected interest points for the original images. The circles in figures (b) are the detected interest points, the size of the circle represents different scales. The number of detected interest points in the three figures (b) is 1768, 3870, 1050, respectively. Figures 8-10 (c) give detected interest points for the original image that is subjected to rotation or scale change. The number of interest points in the three figures (c) is 1767, 964, and 1041, respectively. Experiment results demonstrate that the SURF can still extract the features accurately and effectively after the image is rotated or scaled.



(a) The original image (b) Detected interest points for the original image (c) Detected interest points for the original image that is subjected to rotation

Figure 8. Detected interest points for dress image.

(a) The original image (b) Detected interest points for the original image (c) Detected interest points for the original image that is subjected to rotation and scale change

Figure 9. Detected interest points for satchel image.

(a) The original image (b) Detected interest points for the original image (c) Detected interest points for the original image that is subjected to rotation

Figure 10. Detected interest points for shoes image.

## 4. Summary and Prospect

This paper introduces the SURF implementation process in detail, and uses this algorithm to extract features of clothing images, verifying that it is an excellent local feature descriptor and has advantages of scale- and rotation-invariance. The experiment results show that it has a good effect on the features extraction of clothing images.

Image feature extraction technology is one of the most important contents in computer vision. With the continuous improvement and optimization of feature extraction algorithm, the artificial algorithm has gradually shifted to deep neural network. The deep neural network starts "end-to-end" learning directly from the image data. It learns the inherent regularity of image by stacking layers of networks, so it can extract more abstract high-level semantic information. Figure 11 shows the image features extracted from 16 convolution layers of the VGG-19 pre-training model[12] with the upper left showing the first convolution layer's output and the lower right the 16th convolution layer's output. It can be seen from the figures that the outputs of the first few layers are mainly concentrated in the edges and texture parts; with the deepening of the layers, the features are more and more abstract and concise.

The advantages of deep neural network in image feature extraction, image classification, image recognition, etc. are based on medium and large-scale data. However, in many applications, it is difficult to obtain a large amount of labeling data or the cost for getting the data set is too high. Even if the training data is a large-scale one, it is difficult to cover all situations, for example, a training data set may do not include the affine transformation of various magnitudes. In view of the advantages of SURF, such as local features description, scale- and rotation-invariance, affine-invariance, some scholars are exploring the combination of SURF and deep neural network to obtain the higher precision feature descriptor.

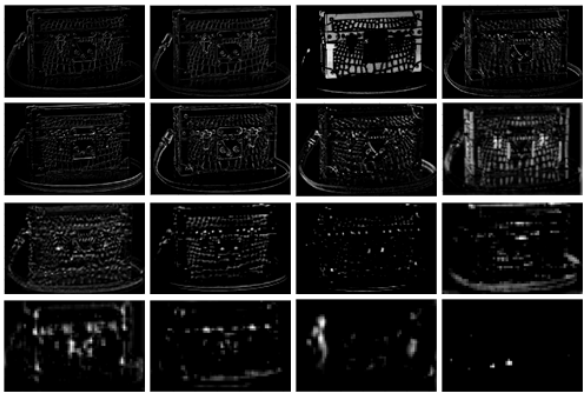


Figure 11. Convolutional output features of VGG-19 pre-training model.

**References**

[1] Wang, Y. M. , Wang, G. J. (2010). Guijin Wang.Image local invariant features and descriptors. National Defense Industry Publishing House, Beijing.
[2] Zhao, X. C. (2012). The modern digital image processing technology and its application in practice(MATLAB Version). Beihang University Press, Beijing.
[3] Lowe, D. G. (1999). Object recognition from local scale-invariant features. In Computer vision, 1999. The proceedings of the seventh IEEE international conference on (Vol. 2, pp. 1150-1157). Ieee.
[4] Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision, 60(2), 91-110.
[5] Bay, H., Tuytelaars, T., & Van Gool, L. (2006, May). Surf: Speeded up robust features. In European conference on computer vision (pp. 404-417). Springer, Berlin, Heidelberg.
[6] Bay, H., Ess, A., Tuytelaars, T., & Van Gool, L. (2008). Speeded-up robust features (SURF). Computer vision and image understanding, 110(3), 346-359.
[7] Chiu, L. C., Chang, T. S., Chen, J. Y., & Chang, N. Y. C. (2013). Fast SIFT design for real-time visual feature extraction. IEEE Transactions on Image Processing, 22(8), 3158-3167.
[8] Bai, X., Dong, X., & Su, Y. (2015). Edge Propagation KD-Trees: Computing Approximate Nearest Neighbor Fields. IEEE Signal Processing Letters, 22(12), 2209-2213.
[9] Koenderink, J. J. (1984). The structure of images. Biological cybernetics, 50(5), 363-370.
[10] Lindeberg, T. (1994). Scale-space theory: A basic tool for analyzing structures at different scales. Journal of applied statistics, 21(1-2), 225-270.
[11] Florack, L. M., ter Haar Romeny, B. M., Koenderink, J. J., & Viergever, M. A. (1992). Scale and the differential structure of images. Image and vision computing, 10(6), 376-388.
[12] Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.
[13] Jiao, L. C., Zhao, J., Yang, S. Y. (2017). Deep Learning, Optimization and Recognition.

Tsinghua University Press, Beijing.
[14] Huang, W. J., Tang, Y. (2017). TensorFlow in practice. Publishing House of Electronics Industry, Beijing.
[15] Zheng, Z. Y., Liang, B. W., Gu, S. Y. (2018). Applying Tensorflow, google deep learning framework, in practice. Publishing House of Electronics Industry, Beijing.
[16] Zhou, Z. H. (2016). Machine learning.Tsinghua University Press, Beijing.