A New Kind of Anti-fake Label Application Based on SIFT Algorithm

Jian LIU*, Shi-qing WANG and Si-li LI

The Engineering & Technical College of Chengdu University of Technology,
Leshan Sichuan 614007, China
*Corresponding author

Keywords: Anti-fake label, Texture feature recognition, SIFT algorithm, Minimum Euclidean-distance method.

Abstract. The traditional anti-fake label is easy to fabricated because it has not a unique feature. To satisfy the trace-to-source requirement of nowadays, a new printed label was proposed which combined the advantages of unique QR code and physically unique texture feature of traditional texture label. Adopting SIFT algorithm, the texture feature of this kind of label was extracted and stored in database, and then labels were matched by using minimum Euclidean-distance method. Experimental Results showed that a good identification and matching accuracy can be obtained even labels’ image were deformed. Finally, a new kind of anti-fake label system were completely conformed by texture paper and QR code, SIFT texture feature recognizing module, texture matching module, database and mobile phone App. In this paper, the emphasis was focused on the middle two parts.

Introduction

Nowadays people pay more attention to the safety of such as medicine, food and they want to confirm what they consumed is genuine or fake. Then the demand of new anti-fake technology which is convenience to use, low cost, difficult to counterfeit become attractive. As the high-speed development of computer and the network technology, especially the popularized using of machine vision and intelligent mobile phone, many kinds of new anti-fake product presented[1].

Aimed at the above demand, a kind of new anti-fake label system was proposed here, which combined the advantages of traditional texture paper, the modern QR code, computer database, network, machine vision technology and mobile intelligent computing. When it was widely used, it would be low cost, highly reliable and convenient to use.

Obviously, how to get the label’s texture feature is the most important problem in this system. For this problem, the SIFT algorithm was applied in the system to realize feature recognizing and the minimum Euclidean-distance method[2,3] was used to match two labels’ feature. The following discussing was focused on SIFT algorithm and its application on real labels’ feature recognizing. At last, the experimental results were presented and analyzed.

The Label & SIFT Algorithm

As described above, the new anti-fake label is consisted of nature texture paper and QR code. The image of real labels is as Figure 1.

Figure 1. Label with physically random texture & QR code.

In present machine vision and image processing field, there are many methods can effectively figure out and match feature of object. Owning to the difficulty and commonly using of these method, the SIFT algorithm was selected here. It was presented by David Lowe[2,3]. As it can tolerate a high
level affine, illumination fluctuation, noise jamming, it is applied widely in target identification, Robot map perception, image seaming, digital watermarking, etc[4-6].

SIFT is a searching algorithm based on the scale space of image and local feature. It is consisted of five steps: constructing the scale space of an image, searching the discrete extrema in scale space, optimizing the location and scale of extrema and then the remaining extrema were considered be key points, assigning the orientation of the key points, creating a descriptor of 128 element for every key point. Following, we describe these steps in more detail.

Construct the DOG Scale Space of Image

Using the Gaussian pyramid method to construct the LOG scale space for an image and then derived the DOG scale space. Firstly expanding the original image into double pixels wide, and let it convolute with a Gaussian filter, to get the No.0 new image. And then down sample the No.0 image by factor 2, to get the No.1 new image. Repeating this process until to get the No. n new image. These n image are called pyramid series and each of it is called a level. The number n is determined by Eq.1,

\[
 n = \log_2\left[\min(M,N)\right] - t, \\
t \in [0, \log_2\left\{\min(M,N)\right\}]
\]  

(1)

M, N are the pixel number on x and y direction of the original image respectively. Now all the images are separated by scale factor 2. In order to remain the scale continuous, adopting a Gaussian function with variable scale factor \( \sigma \) as the scale space kernel, and let it do convolution with each of the new images respectively, then LOG scale space is obtained[7,8],

\[
 L(r, \sigma) = G(r, \sigma) \ast I(r)
\]

(2)

where \( I(r) \) is the input image’s intensity. By Eq.2, a new series images for each level of pyramid series are gotten and they have same pixels and different \( \sigma \). Each of these series, we call it is an octave.

In discrete space, a \( m \times n \) size 2D Gaussian kernel can be write as

\[
 G(x,y) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-m/2)^2+(y-n/2)^2}{2\sigma^2}}
\]

(3)

Obviously, in a same octave, the bigger \( \sigma \) is corresponding to the stronger smooth effect to the input image, and the result image is more indistinct. In Lowe’s paper, the size of Gaussian kernel is recommended as \((6\sigma + 1)(6\sigma + 1)\) wide. To enhance the computing efficiency, one may replace the LOG with DOG in succeeding processing. Let the nearest top image subtract the next one in a same octave, a new series image can be obtained and formed the DOG scale space.

Detect the Local Extrema

As we know, the key points in fact are some local extrema in DOG scale space who satisfy special constrains. At first, comparing \( I(x, y) \) of each pixel with its nearest 8 pixels in the same \( \sigma \) image and the nearest 9×2 pixels in neighbor \( \sigma \) images in the same octave, the local extrema can be obtained. These extrema are the key point candidates.

However, the location and \( \sigma \) of these candidates maybe are not accurate enough. So one should fit their location and \( \sigma \) to rather accurate values. Assuming the original values are estimated value, from the Taylor expansion, the DOG function in the scale space can be written as:

\[
 D(X) = D + \frac{\partial D^T}{\partial X} X + \frac{1}{2} X^T \frac{\partial^2 D}{\partial X^2} X
\]

(4)
where $X=(x, y, \sigma)^T$ is the vector of location and scale factor in discrete space. Suggest the offset of $X$ and the real one is $\hat{X}$, and then solve the derivation of $D(X)$ about $X$ and let it equal zero, $\hat{X}$ can be written as:

$$\hat{X} = -\frac{\partial^2 D^{-1}}{\partial X^2} \frac{\partial D}{\partial X}$$

(5)

If $\hat{X}$ on any dimension is bigger than 0.5, it means the real value is the next point of this dimension. If the new point is out of the border of the scale space then delete it. Do this procedure at the new point repeatedly, the accurate point can be obtained. Because when $|D(x)|$ is very small, the point’s location is easily disturbed by noise, then if $|D(x)|$ is littler than 0.03- 0.04, the candidate extreme point shall be deleted.

**Eliminate the Edge Responses Point**

Because DOG operating produce strong edge response, the extrema which are responding to this effect shall be deleted too. As the Hessian matrix of $D(x)$ is

$$H = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{yx} & D_{yy} \end{bmatrix}$$

(6)

the characteristic value of $H$ can be suggested as $\alpha$, $\beta$. Now $\alpha$ and $\beta$ indicate the gradient on $x$ & $y$ direction respectively. Let

$$Tr(H) = D_{xx} + D_{yy} = \alpha + \beta, \quad Det(H) = D_{xx}D_{yy} - D_{xy}D_{yx} = \alpha\beta$$

(7)

and suggest $\alpha$ is the bigger one, then $\alpha = r\beta$, so

$$\frac{Tr(H)^2}{Det(H)} = \frac{\alpha + \beta)^2}{\alpha\beta} = \frac{(r + 1)^2}{r}$$

(8)

It’s evident that if $r$ is bigger, the ration of the two characteristic value is greater. This means the gradient is big on one direction and is small on the other direction. This result just responds the edge response. So one can estimate whether the point is an edge response by a threshold value of $r$. In Lowe’s paper the threshold was $r=10$.

After the above processing, the remained extrema points were the key points that we wanted.

**Assign Orientation to Key Points**

To make the key points were invariant to image rotation, a consistent orientations based on local image property should be assigned to it. Calculating the gradient in a $3\sigma$ pixels region at key point in the Gaussian pyramid, there were

$$m(x, y) = \frac{[L(x+1, y) - L(x-1, y)]^2 + [L(x+1, y) - L(x-1, y)]}{\sqrt{[L(x+1, y) - L(x-1, y)]^2 + [L(x+1, y) - L(x-1, y)]^2}}$$

$$\theta(x, y) = \tan^{-1}\{[L(x, y+1) - L(x, y-1)]/[L(x+1, y) - L(x-1, y)]\}$$

(9)

In Eq.9, $L$ was the scale factor. And then, counting $m(x, y)$ in sort of $\theta(x, y)$ by dividing the circular direction $\theta$ into 36 parts, an orientation histogram was obtained. The maximum of histogram correspond to the dominant directions of local gradients and it was selected to be the key point’s main orientation. Sometimes, to ensure the direction was more detailed, other direction whose value was with 80% of the main orientation, could be selected to be the assistant orientation. Now, the location, scale, orientation of all the key points were obtained.
Creating SIFT Descriptor for Key Points

To getting a better robustness to illumination change, affine, 3D viewpoint and so on, the final step was to create a SIFT descriptor for each key point. This descriptor included not only the key point information but also include its around pixels’ contribution. Omit the detailed procedure here as it was described in many papers. At last, we got an 128 element feature vector for each key point. These vectors constituted the texture feature.

After got the texture feature, two pictures of label could be compared by their feature. Here the minimum Euclidean-distance method was applied.

Experimental Results

A large number of experiment were done, and a few typical results were listed and analyzed.

Figure 2 shows the result of feature recognizing and matching of a completed label image and its copy which is rotated 180 degree.

![Figure 2](image2.png)

Figure 2. The key point and matching result of label images including QR code region.

In this case, the image contains not only the texture of background but also the QR code region. The key points detected is a very large number. In the original one(left), the key points detected are 1966, and in the rotated one(right), the number is 1805. When the distance factor equal 0.6, there are 698 matched key point pairs between them. In Figure 2, it is evident the key points and matched pairs are major in the QR code region. However, the key points in the background texture region is most significant to application. So a conclusion is obtained that when recognizing the label’s texture feature there shall not include the QR code region in the image.

The next experiment was done to a image without the QR code region. Figure 3 shows the remained part in the image only contains the background texture. The left one is the original, and the right one is 90 degree rotated. The result shows in this case, the texture is well identified in significant region. In the original image there are 69 key points are detected, and in the rotated one there are 69 key points too. Between them, there are 63 pairs key points are matched. This results shows the two image are well and efficiently matched, and the computing amount is decreased greatly.

![Figure 3](image3.png)

Figure 3. The key points and matching result of label images without QR code region.

In practical application, there always exists affine, shielding, etc. These cases were experimented too. Figure 4 shows a stretched image (left) and n undeformed but rotated 90 degrees image of a same label. In this case, there are 40 and 41 key points had been detected. And the successfully matched pairs are 39.
Figure 4. Matching result of stretched & undeformed label images.

Summarized a large amount of experiments, the following conclusion was obtained. When recognizing the texture feature of the label by using SIFT algorithm, the QR code region in the original image should be moved before operating. If the key points recognized were more than 20, it is considered that the texture feature are valid. Based on the above conditions, if the key points of labels were matched more than 85%, it was considered these labels are matched successfully and they could be identified as the same one.

Summary

To meet the need of anti-fake consumption, a new printed label system was proposed. It takes the advantages of physically random texture feature of traditional texture paper and the great amount information of QR code. And then the emphasis was on texture feature recognizing and matching for the label, by adopt SIFT algorithm and minimum Euclidean-distance method. The experimental results shows that SIFT algorithm can accurately and efficiently recognize the texture feature and the way here can match labels reliably.

Acknowledgement

The work was funded by The education department of Sichuan province through the project No.14ZA0316.

References