LETTER

Fast Detection of Robust Features by Reducing the Number of Box Filtering in SURF

Hanhoon PARK[†a)], **Hideki MITSUMINE**[†b)], *Nonmembers*, *and* **Mahito FUJII**[†c)], *Member*

SUMMARY Speeded up robust features (SURF) can detect scale- and rotation-invariant features at high speed by relying on integral images for image convolutions. However, since the number of image convolutions greatly increases in proportion to the image size, another method for reducing the time for detecting features is required. In this letter, we propose a method, called *ordinal convolution*, of reducing the number of image convolutions for fast feature detection in SURF and compare it with a previous method based on sparse sampling.

key words: *ordinal convolution, fast feature detection, SURF*

1. Introduction

In the literature, the problem of detecting robust distinctive features over images captured in different viewing conditions has been studied extensively [4], [5]. The state-of-the-art methods have been successfully used in applications where the real-time constraint is less significant. However, they are too computationally expensive to be applied to real-time applications.

Bay *et al.* [2] enabled to rapidly detect robust features by employing integral image [6]. However, their method, i.e. *SURF*, requires three convolutions with different shaped box filters for each point and thus its speed greatly drops as the image size increases.

Although having a different framework from SURF, Agrawal *et al.* [1] proposed a method, called *CenSurE*, that also uses integral images for convolution and reduces the whole number of convolutions by first computing an approximated circular Laplacian which is computed by a convolution with a box filter and then computing the Harris measure only at the points where the computed approximated circular Laplacian is larger than a threshold. Afterward, Ebrahimi and Mayol-Cuevas [3] made the CenSurE much faster by further reducing the number of convolutions using the concept of sparse sampling. They called the method *SUSurE*.

In this letter, we are interested in making SURF faster and propose an ordinal convolution method for reducing the number of convolutions in SURF. Then, we compare it with one of applying SUSurE to SURF.

Manuscript received July 1, 2010.
Manuscript revised October 8, 2010.

[†]The authors are with NHK (Japan Broadcasting Corporation) Science & Technology Research Laboratories, Tokyo, 157–8510 Japan.
a) E-mail: hanhoon.park@strlstaff.strl.nhk.or.jp
b) E-mail: mitsumine.h-gk@nhk.or.jp
c) E-mail: fujii.m-ii@nhk.or.jp

DOI: 10.1587/transinf.E94.D.725

2. Ordinal Convolution

In order to detect features in an image, SURF [2] computes the determinants (usually called *Hessian*) of the approximated Hessian matrix at different scales as follows.

$$H(\mathbf{x}, \sigma) = D_{xx}(\mathbf{x}, \sigma)D_{yy}(\mathbf{x}, \sigma) - \{0.9D_{xy}(\mathbf{x}, \sigma)\}^2, \quad (1)$$

where $D_{xx}(\mathbf{x}, \sigma)$ represents the convolution of a box filter (approximating the Laplacian of Gaussian filter) of scale σ with the image at point $\mathbf{x} = (x, y)$ and similarly for D_{yy} and D_{xy}. Here, although each convolution can be accelerated by using integral image [6], the Hessian computation needs three convolutions for each point and for each scale and the number of convolutions increases quadratically as the image size increases (see Fig. 1).

In order to reduce the number of convolutions, we pay attention to the fact that the Hessian value of distinctive features must be large, which indicates that both D_{xx} and D_{yy} must be large and D_{xy} must be small. From the fact, we propose an ordinal convolution method as follows. D_{xx} is computed first for each point. Then, D_{yy} and D_{xy} are not computed at the points where D_{xx} is smaller than a threshold (th_D) and the Hessian value is set to 0. If D_{xx} is larger than th_D, D_{yy} is computed next. Again, D_{xy} is not computed at the points where D_{yy} is smaller than th_D and the Hessian value is set to 0. Finally, D_{xy} is computed at the points where both D_{xx} and D_{yy} are larger than th_D and the Hessian value

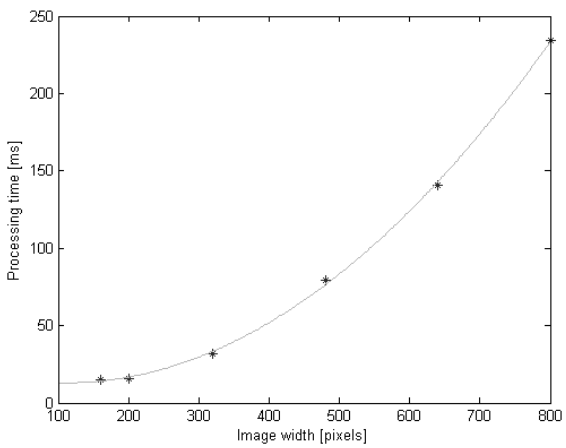

Fig. 1 Processing time of feature detection in SURF vs. image width (height = 0.75*width). The open library [8] was used in a laptop computer equipped with a dual-core 2.8 GHz CPU.

Copyright © 2011 The Institute of Electronics, Information and Communication Engineers

is computed. The pseudo code is shown in Fig. 2.

The order of computing D_{xx}, D_{yy}, and D_{xy} can be changed, e.g. D_{xy} is computed first, D_{xx} is computed then only at the points where D_{xy} is smaller than a threshold, and D_{yy} is computed finally only at the points where D_{xx} is larger than a threshold. However, through preliminary experiments, we could know that the order is not an important factor.

3. Applying SUSurE to SURF

For making this letter more self-contained, we briefly explain SUSurE [3] in this section. Then, we discuss how to apply SUSurE to SURF.

After computing a filter response (the Hessian value in SURF) at a point, if the response (R) is weaker than a threshold (th_R), the filter response is not computed and R is copied for the next N pixels. N is computed as follows.

```
Compute D_xx;
IF D_xx < th_D
   Hessian = 0;
ELSE
   Compute D_yy;
   IF D_yy < th_D
      Hessian = 0;
   ELSE
      Compute D_xy;
      Hessian = D_xx D_yy − 0.9 D_xy D_xy;
   END
END
```

Fig. 2 Ordinal convolution in SURF.

$$N = \begin{cases} 0 & \text{if } R_{x,y} > th_R \\ \left\lfloor \frac{L}{2}(1 - \frac{R_{x,y}}{th_R}) \right\rfloor & \text{if } R_{x,y} \le th_R \end{cases} \quad (2)$$

where $R_{x,y}$ is the filter response at the point (x, y) and L is the filter length.

For applying SUSurE to SURF, we must consider that the interval between the successive filter responses in SURF is doubled for next octave. Therefore, N must be rescaled for each octave as follows.

$$N_k = N/i_k \quad (3)$$

where N_k and i_k are the number of skipped pixels and the interval in the k-th octave, respectively.

4. Experimental Results and Discussion

For experiments, we modified the open library [8] and the images (see Fig. 3) provided by Mikolajczyk [7]. Repeatability (for which the MATLAB code in [7] was used) and processing time (for which a laptop computer equipped with a dual-core 2.8 GHz CPU was used) of two methods, the proposed ordinal convolution method and one of applying SUSurE to SURF, were compared with the Hessian threshold of 0.0004. th_D was set to the values ranging from 0 to

Fig. 3 Images used in experiments. From left, *bikes* (1000 × 700), *boat* (850 × 680), *leuven* (900 × 600), *ubc* (800 × 640), and *wall* (1000 × 700) which include different variations: blur, 2D rotation and zoom, light, JPEG compression, and viewpoint.

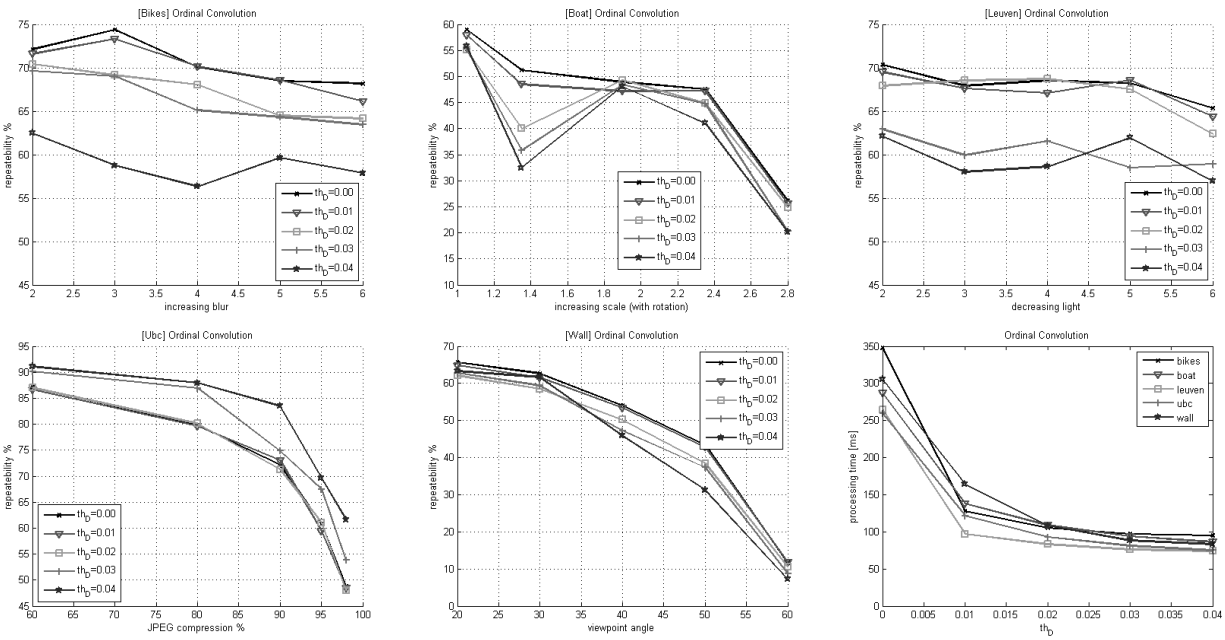

Fig. 4 Repeatability and processing time of the proposed ordinal convolution.

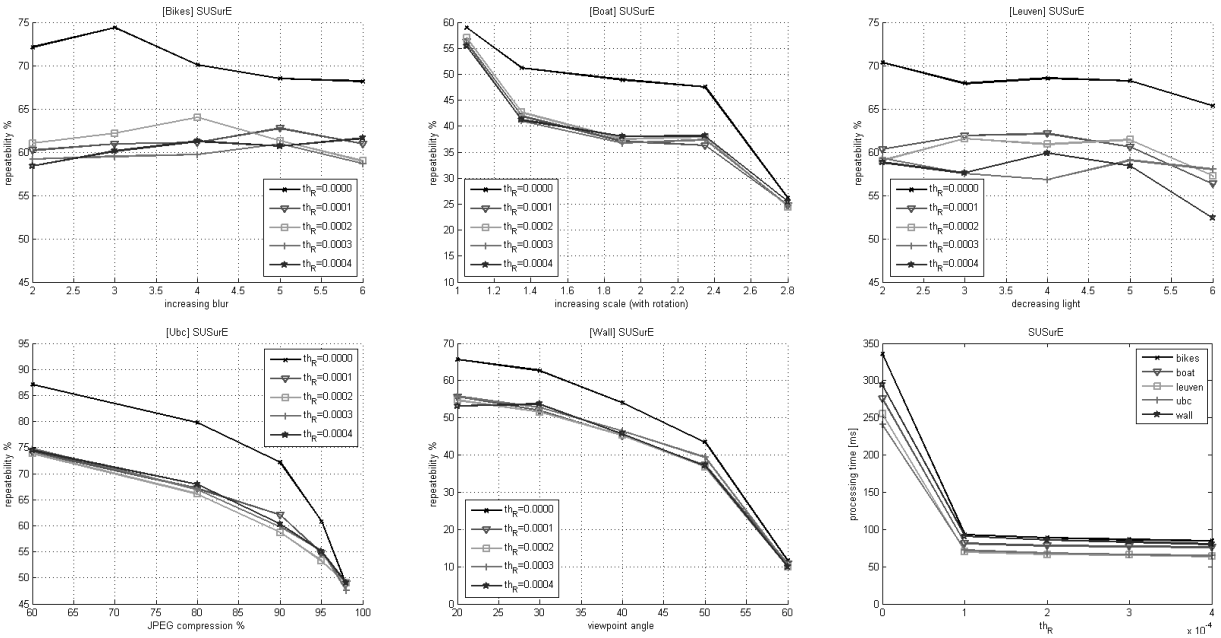

Fig. 5 Repeatability and processing time of SUSurE when L=(filter length).

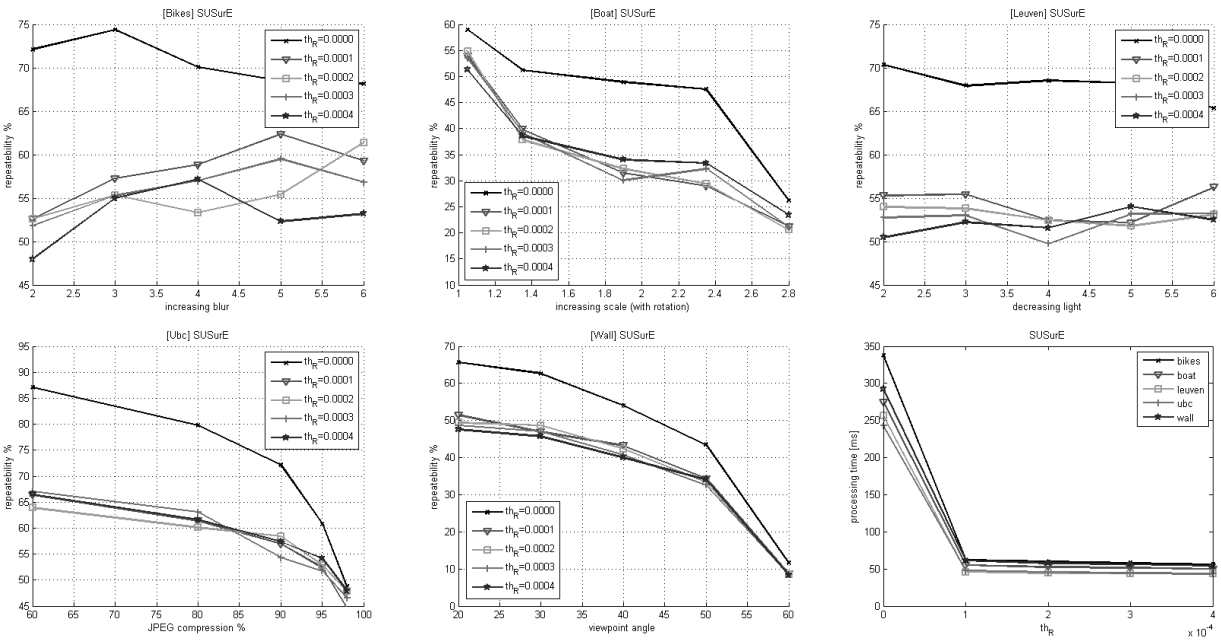

Fig. 6 Repeatability and processing time of SUSurE when L was doubled.

0.04. th_R was set to the values ranging from 0 to 0.0004.

Figure 4 shows the results of the proposed ordinal convolution method and Fig. 5 shows the result of applying SUSurE to SURF. The processing time of detecting features was reduced by about 2-3 times by both methods although being slightly more reduced by SUSurE[†]. Therefore, both methods were useful for reducing the processing time. However, the degree of the involved drop in repeatability was too severe in SUSurE. SUSurE dropped (by 10-15%) the repeatability of detected features so that its repeatability was about 10% lower than that by the proposed ordinal convolution method. In fact, this was predictable from the fact that SUSurE influences the location of the detected features while the proposed ordinal convolution method does not.

The proposed ordinal convolution method showed a good trade-off between the processing time and the repeata-

[†]If setting th_D to a value larger than 0.04, the processing time of the proposed ordinal convolution method could be more reduced but an enough number of features could not be detected.

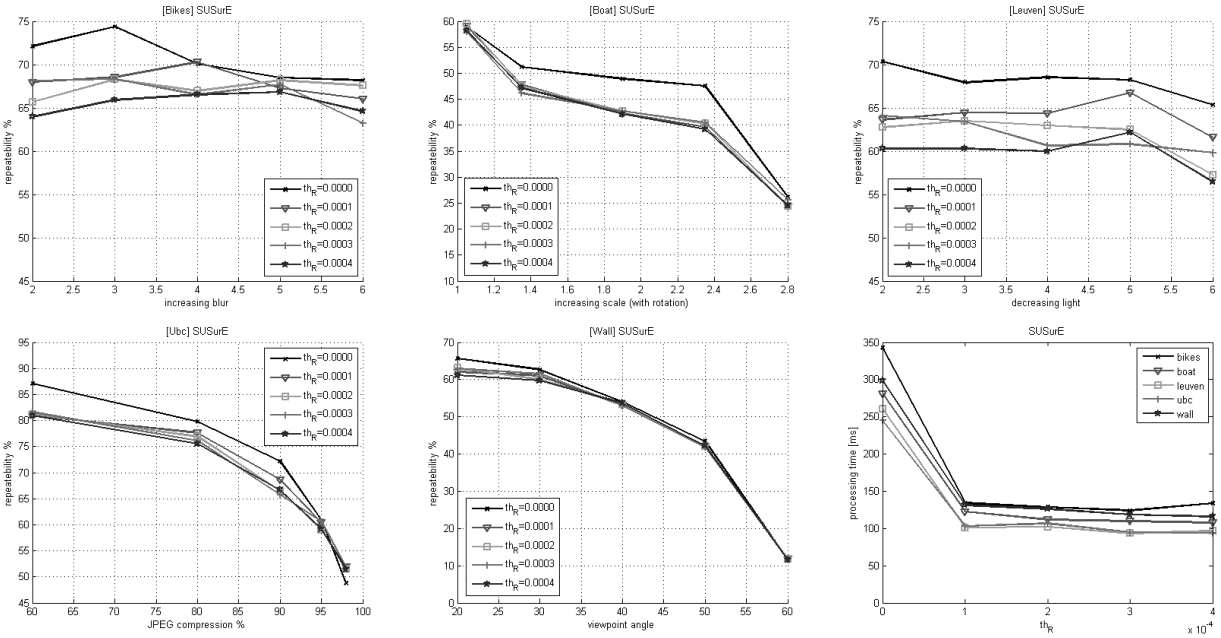

Fig. 7 Repeatability and processing time of SUSurE when L was halved.

bility when th_D=0.02. However, it was difficult for SUSurE to find a good trade-off between the processing time and the repeatability because the results of SUSurE did not change greatly except when th_R is 0 in our experiments†. Therefore, we tried to change L. Figures 6 and 7 show the results of SUSurE with different L values. Doubling L greatly reduced the processing time but made it severer the drop in repeatability. In contrast, halving L improved the repeatability but the repeatability was still lower than that of the proposed ordinal convolution method. In addition, halving L increased its processing time, which was about 20 ms longer than that of the proposed ordinal convolution method.

The tendency in the results was not different for each image but the effectiveness of the proposed ordinal convolution method was more apparent for *bikes*, *leuven*, and *ubc*.

Consequently, in the applications that require strict real-time constraint, SUSurE can be used at the expense of a severe drop in repeatability. However, in general environments, the proposed ordinal convolution method, which reduces considerably the processing time while maintaining the repeatability (when th_D is smaller than 0.03, the drop in repeatability was lower than 5%), will be more desirable.

†In the open library [8], if the Hessian is negative, it is set to 0. Due to this, in Eq. (2), most $R_{x,y}$s satisfying $R_{x,y} \leq th_R$ was 0. Therefore, the results of Eq. (2) were little influenced by the value of th_R. In fact, it may mean that Eq. (2) does not work well for SURF. We think that it is because the computation scheme for $R_{x,y}$ is different from the bilevel LoG filter response in the original SUSurE [3]. Therefore, for fitting SuSurE perfectly to SURF, Eq. (2) should have been modified more carefully, which is beyond our scope. This would be able to explain why the performance of SuSurE in this letter was not as good as in [3].

5. Conclusion

In this letter, we proposed an ordinal convolution method of reducing the number of image convolutions for fast feature detection in SURF and compared it with applying SUSurE to SURF. The proposed ordinal convolution method could greatly reduced the processing time (about 2-3 times) while maintaining the repeatability of detected features (within 5%) and thus outperformed the method of applying SUSurE to SURF (though SUSurE did not fit SURF perfectly).

The proposed ordinal convolution method can be applied to other Hessian-based feature detectors. The related analysis remains a near future work.

References

[1] M. Agrawal, K. Konolige, and M.R. Blas, "CenSurE: Center surround extremas for realtime feature detection and matching," Proc. Euro. Conf. Computer Vision, pp.102–115, 2008.
[2] H. Bay, T. Tuytelaars, and L. Van Gool, "SURF: Speeded up robust features," Proc. Euro. Conf. Computer Vision, pp.404–417, 2006.
[3] M. Ebrahimi and W.W. Mayol-Cuevas, "SUSurE: Speeded up surround extrema feature detector and descriptor for realtime applications," Proc. IEEE Conf. Computer Vision and Pattern Recognition Workshop, pp.9–14, 2009.
[4] D.G. Lowe, "Distinctive image features from scale-invariant keypoints," Int. J. Comput. Vis., vol.60, no.2, pp.92–110, 2004.
[5] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. Van Gool, "A comparison of affine region detectors," Int. J. Comput. Vis., vol.65, no.1/2, pp.43–72, 2005.
[6] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," Proc. IEEE Conf. Computer Vision and Pattern Recognition, pp.511–518, 2001.
[7] http://www.robots.ox.ac.uk/~vgg/research/affine/
[8] http://code.google.com/p/opensurf1/