AN EFFECTIVE APPROACH TO CORNER POINT DETECTION THROUGH MULTIRESOLUTION ANALYSIS

Yang Bai, Hairong Qi

Department of Electrical Engineering and Computer Science
University of Tennessee
Knoxville, TN 37996
{ybai2,hqi}@utk.edu

ABSTRACT

Feature points are low-level image features representing meaningful image regions and ideal candidates for feature-based image representation, and feature point detection is an essential pre-processing step for high-level computer vision tasks. Existing feature detection algorithms are either computationally intensive (multi-scale detectors) or sensitive to scale variations (single-scale detectors). In this paper, we propose a computationally efficient multi-scale corner detector based on Discrete Wavelet Transform (DWT). We use non-redundant DWT coefficients to build a corner strength map at each scale in a data-compact way and upsample these maps by the Gaussian kernel interpolation to the original image size. By taking the summation of these maps, a corner strength measure is formed. We propose a new scale selection method that utilizes a Gaussian kernel convolution to measure the corner distribution in the vicinity of every corner point. In addition, the so-called "Polarized Gaussian" kernels are introduced to achieve rotational invariance. The high efficiency of the proposed corner detector is shown through both computational complexity analysis and accuracy analysis.

***Index Terms*—** Corner Point Detection, Multiresolution, Discrete Wavelet Transform

1. INTRODUCTION

Feature points are powerful image features in that they indicate junctions of different objects or different areas. Many feature detection methods have been proposed, most of which fall into two categories. Methods in the first category measure the corner strength using image derivatives. Harris et al. constructed a second moment matrix and used its eigenvalues to measure the corner strength [1]. Harris detector is computationally efficient but lacks scale invariance, making it unstable under scale changes. Mikolajczyk et al. extended the Harris detector into scale space with automatic scale selection, making it scale invariant [2]. In Scale Invariant Feature Transform (SIFT), Lowe used the Difference of Gaussian (DoG) to build the image pyramid for detecting keypoints [3]. Speeded-Up Robust Features (SURF), was designed by Bay [4] following the similar principle as SIFT, but with lower computational complexity due to the usage of the integral image for fast convolutions. These multi-scale detectors are built from the Gaussian kernel, making them computationally intensive.

Methods in the second category analyze the local patch of every pixel. Smith et al. proposed the Smallest Univalue Segment Assimilating Nucleus (SUSAN) corner detector [5], where the corner strength measure is defined as a weighted sum of pixel intensities within a disc area surrounding the central pixel. Rosten et al. designed the Features from Accelerated Segment Test (FAST) corner detector [6]. The corner strength measure is defined as the number of consecutive pixels darker/brighter than the central pixel on a Bresenham circle. These single-scale corner detectors are computationally efficient, but are unstable under scale changes.

We propose in this paper a multi-scale corner detector that is both computationally efficient and presents high accuracy. The corner strength measure is calculated from the non-redundant DWT coefficients. The scale parameter of a corner point is defined as a function of the corner distribution at its vicinity. To make the scale calculation efficient, we provide an approximate solution using a Gaussian kernel convolution. By introducing the so-called "Polarized Gaussian" kernels, the proposed corner detection method achieves rotational invariance.

2. CORNER DETECTION FROM DUAL-TREE COMPLEX WAVELET TRANSFORM (DTCWT)

2-D DWT decomposes an image into multiple scales. The orthogonal wavelet based decomposition is non-redundant, thus incurring no additional storage overhead. Fast Wavelet Transform (FWT), an efficient implementation of the DWT, further exploits the relationship of DWT coefficients between adjacent scales, making DWT computationally efficient.

In images, we define a set of connected pixels with different intensities on their two sides as an "edge" and the intersection of edges as a "corner". By using the 2-D DWT decompo-

978-1-4577-1302-6/11/$26.00 ©2011 IEEE

sition, we can find the edges by examining the local extremes of wavelet coefficients along the horizontal, vertical or diagonal directions. Then we can find a corner point by looking at the coordinates where two or three wavelet coefficients along different directions both/all reach a local extreme.

The DWT is not isotropic, meaning the edge detection is not rotationally invariant, and so does the corner detection. Recently Fauqueur et al. proposed a corner detector that applies the Dual-Tree Complex Wavelet Transform (DTCWT) [7]. The DTCWT is directional selective, therefore making the corresponding corner detector rotational invariant. They build a "Keypoint Energy Map" for localizing keypoints from decimated DTCWT coefficients, and the keypoint scale parameter is determined by the gradient minima of the keypoint energy map in its vicinity.

However, the DTCWT analysis is redundant, specifically, the data volume of the decomposition structure doubles the size of the original image, and thus incurs additional computational cost and data storage. The additional overhead to compensating the rotational variance, sometimes could be hazardous, especially for the computer vision algorithms running on low end systems, for example, the resource-constraint wireless visual sensor networks, where both the computational resource and the data storage resource are scarce [8].

3. CORNER DETECTION FROM NON-REDUNDANT DISCRETE WAVELET TRANSFORM

In this section, we detail the design of the DWT-based corner detector. Compared to DTCWT, the proposed corner detector shows advantages from four aspects. Firstly, the image scale space is built from non-redundant DWT, making the decomposition data possessing the same size as the original image, incurring no additional storage space, and therefore no additional data computation. Secondly, we use the Haar wavelet basis, which constitutes a set of orthogonal wavelet bases, and therefore the DWT decomposition can be implemented through FWT, having favorable computational speed. Thirdly, the scale parameter calculation is implemented through approximating the scale function using Gaussian kernel convolution, which is more computationally efficient. At last, the non-isotropic property of DWT is compensated by introducing the 6 polarized Gaussian kernels.

3.1. Determine the Corner Point Locations

The DWT decomposition of an image of size $m \times n$ using Haar wavelet results in a decimated dyadic decomposition up to scale J. This means for each pixel in the original image, at each decomposition scale s, we have three wavelet coefficients denoted as $W1_s(i,j)$, $W2_s(i,j)$ and $W3_s(i,j)$, respectively. We follow the same formulation as in [7] to define the value of the corner strength map C_s at coordinate (i,j) as

$$C_s(i,j) = (\prod_{t=1}^{3} |Wt_s(i,j)|)^{\frac{1}{3}} \tag{1}$$

where $i = 1, 2, \cdots, m/2^s$, $j = 1, 2, \cdots, n/2^s$.

Due to the decimated decomposition, the size of each C_s differs by 2^s. Detecting local maxima on C_s separately at each scale would result in poor corner point localization in the original image. As suggested in [7], we instead interpolate C_s at each scale up to the original image size $m \times n$ using a Gaussian kernel (i.e., upsample by the factor of 2^s for C_s) with the standard deviation proportional to the scale factor 2^s. Denote the interpolated corner strength map C_s as IC_s. The corner strength measure at the original resolution is then

$$C = \sum_{s=1}^{J} IC_s. \tag{2}$$

The DWT is not isotropic, therefore the corner strength measure C is not rotational invariant. This means when a straight line rotates, due to the discretization, the same line will show consecutive "zig-zag" patterns and these will be detected as crowed corner points. To achieve rotational invariance, we firstly introduce the following *scale parameter* to identify (and thus remove) the crowed corner points and then introduce the *polarized Gaussian kernels* to achieve rotational invariance. By using this two-step procedure, we leave the most computationally intensive process, i.e., the convolution with the 6 polarized Gaussian kernels, to the very end, which applies only on a small set of pixels, thus reducing the computational complexity to a great extent.

3.2. Determine the Corner Point Scales

The scale parameter calculation is designed based on the observation that the false corners on an inclined discrete straight line are usually clustered together. Therefore, it is appropriate to assign small scale parameters to corners where they are densely distributed. In other words, the corner scale is related to the corner distribution at its vicinity. The two factors involved in describing the corner distribution for a reference corner are: How many corners are within the neighborhood and how far away these neighbor corners are from the reference corner. Then the value of the corner scale map S at (i,j) is defined as

$$S(i,j) = \frac{\sum_{c \in N_c(i,j)} d_c(i,j)}{|N_c(i,j)|} \tag{3}$$

where $d_c(i,j)$ is the Euclidean distance between the reference corner at (i,j) and its neighbor corner c, $N_c(i,j)$ is the set of all neighbor corners of the reference corner, $|N_c(i,j)|$ is the number of corners in $N_c(i,j)$.

Although the two factors in Eq. 3 can be calculated explicitly, finding these accurate values would be time consuming.

To speed up the scale parameter calculation, we propose a new method to approximate Eq. 3. We name C_m the "Corner Map", which is an $m \times n$ matrix with binary entries, with 1 indicating a corner point at that coordinate and 0 indicating no corner. The Corner Map C_m is the resulting matrix of taking the local maximum from the corner strength measure C and then binarizing. We convolve C_m with a Gaussian kernel, and only record the convolution results at the coordinates that correspond to corners. For a single corner point on C_m, the convolution result equals to a weighted sum of 1s (if there is a neighbor corner) and 0s (if there is no neighbor corner) at its vicinity and the weight is decided by the Gaussian kernel. For a specific corner, the closer other corners to it, the larger the summation is; the more neighbor corners it has, the larger the summation is. At last, the reciprocals of the Gaussian convolution results are used as the scale parameter for corners.

To finally achieve the rotational invariance, we define the so-called "Polarized Gaussian" kernels. By examining the convolution responses to these kernels, the false corners from the rotation can be removed and the DWT-based corner detector becomes rotational invariant.

A polarized Gaussian kernel is created by element-wise multiplying a Gaussian kernel with a binary mask. The elements in the binary mask are either "1"s (the elements in the upper half and the right side of the horizontal central line of the binary mask) or "-1"s (the rest elements). Figure 1 shows the calculation for a polarized Gaussian kernel. The element summation inside a polarized Gaussian kernel along any direction passing the center is zero, thus making this kernel have small responses to false corners, and large responses to real corners when convolving with the corner map C_m. A real corner may have small response for a polarized Gaussian kernel if its two rays fall into the two polarizations separately, but by providing another five rotated polarized Gaussian kernels, this possibility has been significantly minimized. Figure 2 shows the 6 polarized Gaussian kernels by rotating the first one 30°, 60°, 90°, 120°, and 150°, respectively.

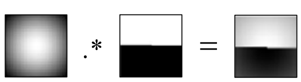


Fig. 1. Building a polarized Gaussian kernel by multiplying a Gaussian kernel with a binary mask, where ".∗" represents the element-wise multiplication.

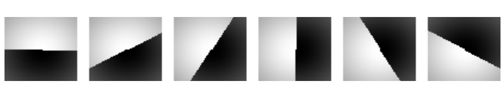


Fig. 2. The six polarized Gaussian kernels.

The scale selection algorithm is summarized as follows

- Step 1: Convolve C_m with a Gaussian kernel, and the reciprocal of the result is used as an approximation to the scale parameter for each corner point, $S(i, j)$. Most real corners would have a larger scale parameter and all corners detected from the discretization effect (i.e., edge points) as well as some very few real corners would have a smaller scale parameter.
- Step 2: To separate edge points and some real corners with a smaller scale parameter, convolve these points with the 6 polarized Gaussian kernels, respectively. Remove corners that have small responses for all 6 polarized Gaussian kernels as these would indicate the edge points.

4. EXPERIMENTAL RESULTS

We evaluate the performance of the proposed DWT-based corner detector in terms of both computational complexity and the accuracy measured by the repeatability rate. The proposed algorithm is compared with three state-of-the-art multi-scale feature point detectors, namely, the detector in SIFT, the detector in SURF, and the multi-scale Harris detector.

4.1. Computational Complexity

To evaluate the computational efficiency, we compare the number of operations (addition, multiplication and comparison) for the four multi-scale feature point detectors, as a function of the image size. Fig. 3 shows the results. All these four detectors have computational complexities which are linear functions of the image size. Among the four detectors, the detector in SIFT is most computationally intensive, followed by the detector in SURF and the multi-scale Harris detector. The DWT-based corner detector is the most efficient compared to other multi-scale detectors, because the scale space from a DWT decomposition is more compact than the Gaussian scale space.

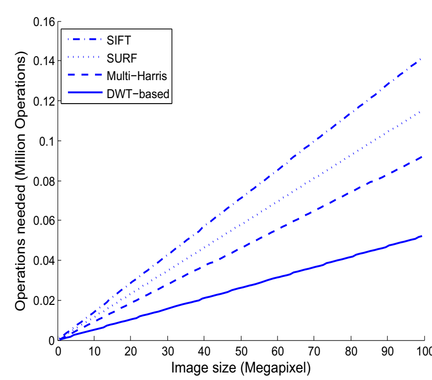


Fig. 3. Number of operations (multiplications, additions and comparisons) as a function of the image size. Image sizes varies from low resolution of 600×800 to high resolution up to 100 Megapixels.

We then apply the four detectors on the 6 "graffiti" images in the Oxford dataset [1], and record their average running

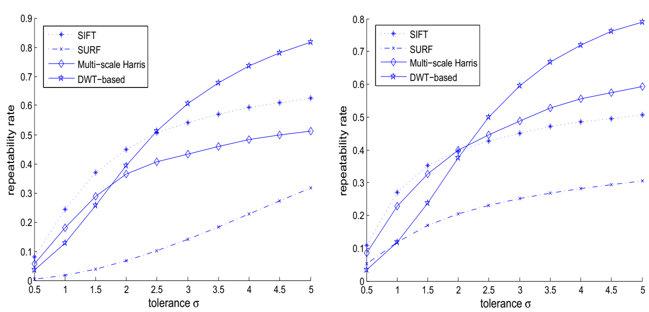
[1] http://www.robots.ox.ac.uk/∼vgg/research/affine/

times. The detector in SIFT takes 7.45 seconds to detect 419 feature points, the detector in SURF takes 1.47 seconds to detect 516 feature points, the multi-scale Harris detector takes 4.77 seconds to detect 458 corners, and the DWT-based detector takes 1.65 seconds to detect 536 corners. Because the multi-scale Harris detector and the DWT-based corner detector are implemented in Matlab code, but the detectors in SIFT and SURF are implemented in C code, these values are listed for illustration purpose only.

4.2. Repeatability

Repeatability is used to evaluate the accuracy of the feature detectors, which measures the geometrical stability of the detected feature points between different images of the same scene taken under different viewing conditions [9]. The repeatability is a function of the tolerance σ (repeated detections do not exactly overlap, but are σ pixels away). To calculate the repeatability, we need images from a planar scene, so that these images can be related by a homography transform.

There are only a few image datasets containing images from planar scenes and Schmid et al. generated 2 image sets of their own in evaluating different feature point detectors [9]. We use the 6 "graffiti" and 6 "wall" images from the Oxford dataset, both taken from planar scenes, but under various geometric transformations, including translation, scaling, rotation, and perspective changes. The repeatability rates are calculated on every 2-image pairs, then the average is taken from the 15 image pairs, as shown in Fig. 4 (a) for the "graffiti" images, and Fig. 4 (b) for the "wall" images.



(a) (b)

Fig. 4. Repeatability rate as a function of the tolerance (σ) for the 6 "graffiti" images (a) and the 6 "wall" images (b).

As shown in Fig. 4, the DWT-based corner detector has the highest repeatability rate, once the σ is greater than 2.5. This is because the scale space from the DWT decomposition is not as fine as the scale space using Gaussian function, a price paid for the data compactness. However, this tolerance is acceptable in most situations, because mature feature descriptors, like the histogram of local gradients used in SIFT, can compensate for the feature point localization error at this level [3].

5. CONCLUSION

We presented a computationally efficient method for multiresolution corner detection using Haar wavelet based DWT. The contribution of this paper is two-fold: Firstly, non-redundant DWT decomposition is utilized to build a compact scale space for multi-scale corner detection, reducing both data storage space and computational cost; secondly, the drawback of using non-redundant DWT, i.e., the non-isotropic property, is compensated by introducing a scale parameter and polarized Gaussian kernels, with little overhead in computational cost. Experimental results showed our approach has the lowest computational complexity compared to state-of-the-art multi-scale feature detectors, and has the best performance in terms of repeatability, under a reasonable tolerance.

6. REFERENCES

[1] C. Harris and M. Stephens, "A combined corner and edge detector," *Proceedings of the 4th Alvey Vision Conference*, pp. 77–116, 1988.

[2] Krystian Mikolajczyk and Cordelia Schmid, "Scale & affine invariant interest point detectors," *International Journal on Computer Vision*, vol. 60, no. 1, pp. 63–86, 2004.

[3] David G. Lowe, "Distinctive image feature from scale-invariant keypoints," *International Journal of Compputer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[4] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool, "Speeded-up robust features (surf)," *Computer Vision and Image Understanding*, vol. 110, pp. 346–359, 2008.

[5] S.M. Smith and J.M. Brady, "Susan - a new approach to low level image processing," *International Journal of Computer Vision*, vol. 23, pp. 45–78, 1997.

[6] E. Rosten and T. Drummond, "Machine learning for high-speed corner detection," in *European Conference on Computer Vision*, 2006, vol. 1, pp. 430–443.

[7] Julien Fauqueur, Nick Kingsbury, and Ryan Anderson, "Multiscale keypoint detection using the dual-tree complex wavelet transform," *IEEE International Conference on Image Processing*, pp. 1625–1628, 2006.

[8] Yang Bai and Hairong Qi, "Redundancy removal through semantic neighbor selection in visual sensor networks," *Third ACM/IEEE International Conference on Distributed Smart Cameras*, pp. 1–8, 2009.

[9] Cordelia Schmid, Roger Mohr, and Christian Bauckhage, "Evaluation of interest point detectors," *International Journal of Computer Vision*, vol. 37, pp. 151–172, 2000.