COMPARISON OF IMAGE ALIGNMENT ON HEXAGONAL AND SQUARE LATTICES

Tetsuo Shima, Shigeki Sugimoto, and Masatoshi Okutomi

Tokyo Institute of Technology
2-12-1-S5-22 Ookayama, Meguro-ku, Tokyo 152-8550 Japan

ABSTRACT
A hexagonal lattice has been researched to improve computer vision and image processing. However, image alignment on the lattice has not been fully discussed yet. In this paper, we perform image alignment on hexagonal and square lattices. Then we compare and evaluate them. We used hexagonal lattices of two sizes. One has the same pixel interval between adjacent pixels as the square lattice, which results in higher resolution than the square one. Another has the same pixel area as the square lattice, which has the same resolution as the square one. The results show that the image alignment of both the hexagonal lattices outperforms that of the square lattice with respect to accuracy and the success rate. Importantly, the results also show that converting an existing large image on the square lattice into the smaller image on the hexagonal ones of both the sizes could improve image alignment.

Index Terms— image alignment, image registration, hexagonal lattice, square lattice, consistent gradient operator

1. INTRODUCTION
Images are usually constructed by square lattices in computers and image capture devices. However, being square is not the only solution to construct images. On square lattices, the distances from a pixel to its eight adjacent pixels are not the same, and only the distances to its four adjacent pixels are the same. On the other hand, on hexagonal lattices, the distances to all six adjacent pixels are equal. Consequently, a hexagonal lattice has higher isotropy in comparison to a square lattice. From the aspect of biomimic approach, the human retina has a lattice structure that is similar to hexagon rather than square. Because of these advantages of isotropy and biological affinity of hexagonal lattices, many studies of a hexagonal lattice have been presented. Kamgar-Parsi et al. [1] compared quantization errors in sampling on hexagonal and square lattices. Staunton and Storey [2] compared the lattices for pipeline image processing. Allen [3] proposed hexagonal perfect reconstruction filter banks. Van De Ville et al. [4] presented a spline family for a hexagonal lattice. Bhattacharje and Narayanan [5] presented hexagonal geometry clipmaps for terrain mapping on a sphere. Hauschild et al. [6] presented a test chip for CMOS optical sensor on a hexagonal lattice. Middleton and Sivaswamy [7] gave a comprehensive survey on image processing on a hexagonal lattice.

For years, many previous works about image processing on hexagonal lattices have been proposed, and the papers above are only a part of that. Although many studies of hexagonal lattices have been proposed, image alignment on the lattice has not been fully discussed yet. Image alignment is an important preprocessing for image processing and computer vision systems. A typical approach to image alignment is a direct method such as the Lucas-Kanade method [8]. Baker and Matthews [9] proposed inverse compositional image alignment algorithm that is equivalent to the Lucas-Kanade method and achieved higher efficiency [10].

Solving image alignment problem using direct methods must extract gradient values from input image. The Sobel operator is a frequently used gradient operator; nevertheless, it has inconsistency. Consistent gradient operators on square lattices [11] minimized the inconsistency, and the operators on hexagonal lattices [12] have also been proposed. We expect that higher isotropy by hexagonal lattices and the consistent gradient operator will improve image alignment.

In this paper, we compare and evaluate hexagonal and square lattices with respect to accuracy and success rate of image alignment using those gradient operators. The results show that the hexagonal lattice with the consistent gradient operator outperforms the square one. The results also show that converting an existing large image on the square one into the smaller image on the hexagonal one improves accuracy of image alignment in comparison to the conversion that only uses the square lattice.

2. DEFINITION OF HEXAGONAL LATTICE
We presume the horizontal and vertical distances between the adjacent pixels on a square lattice are 1, and pixel width of the square lattice is also 1, as in Figure 1(a). On the other hand, we adopt the hexagonal lattice that has concatenated pixels in the horizontal direction, as shown in Figure 1(b). Moreover, we use two hexagonal lattices that have different sizes of pixels. First one has the same pixel width with the square lattice, and second one has the same pixel density (resolution) with the square one. We define the hexagonal lattices of two different sizes: Hex(W) and Hex(A). First, Hex(W) has width 1
for a pixel. Second, Hex(A) has width $\sqrt{2}/3^{1/4}$ for a pixel and a pixel area is 1. When $100 \times 100$ pixels are used on a square lattice to cover a given area, Hex(W) should have $100 \times 115$ pixels to cover the same area, and Hex(A) should have $93 \times 107 = 9951$ pixels to cover the given area. Hex(W) simulates image sensor of the same width of existing sensor on the square lattice. We mainly presume that Hex(A) is used for the downscaled image that is obtained from the large image on the square lattice, instead of simply downsampling on the square lattice.

3. IMAGE ALIGNMENT ON A HEXAGONAL LATTICE

We perform image alignment on the hexagonal lattices, because higher isotropy of the lattices will improve the results of that. Both the lattices Hex(W) and Hex(A) have higher isotropy than the square lattice. The hexagonal lattice Hex(W), moreover, has higher resolution than the square one. We specifically examine using direct methods to solve problems of image alignment. When using direct methods, it is necessary to extract gradient values from an input image. Consequently, appropriate gradient operator will improve the results of image alignment. Frequently used gradient operators such as the Sobel operator have inconsistency of directions. On hexagonal and square lattices, consistent gradient operators are designed to minimize the inconsistency in direction of the operators. Consistent gradient operators are the most accurate gradient operators in terms of that we prefer isotropy and equivalence to continuous gradient as possible. For more details, refer to the papers presented on square lattices [11] and on hexagonal lattices [12]. Then, on the hexagonal lattices Hex(W) and Hex(A), we use the consistent gradient operator of radius 1. On the square lattice, we use the consistent gradient operator of $3 \times 3$, and also use the Sobel operator for comparison. Both the radius 1 and $3 \times 3$ are the symmetric operator shape that achieve the minimum number of pixels to construct an operator and have a center pixel on each lattice.

More specifically, we perform image alignments as follows. We use inverse compositional image alignment (ICIA) algorithm [9, 10] to perform image alignment on an image pair. Let $I$ and $T$ respectively signify the input and template images, let $x$ be point coordinate, let $W$ be warping, and let $p$ be a homography matrix of projective transformation. Basically, ICIA optimizes

$$\sum_x |T(W(x, \Delta p)) - I(W(x, p))|.$$  

(1)

with respect to the homography matrix $p$. The procedure of ICIA needs to calculate the gradient of $T$. On the square lattice, rotating the operator in the x direction $90^\circ$ yields the operator in the y direction. We compute the image gradients by application of the Sobel operator or the consistent gradient operator in the x and y directions. On the hexagonal lattices, rotating the operator in the x direction $60^\circ$ and $120^\circ$ respective yield the operator in the $60^\circ$ and $120^\circ$ directions. We compose a gradient operator in the x direction as $J_x = 2/3 \cdot (J_0 + 0.5 \cdot (J_1 - J_2))$ and the operator in the y direction by $J_y = 1/\sqrt{3} \cdot (J_1 + J_2)$ where $J_0$, $J_1$ and $J_2$ are respective gradient operator in the $0^\circ$, $60^\circ$ and $120^\circ$ directions. Then we compute the image gradients by application of these operators in the x and y directions.

The procedure of ICIA includes image warping, we must interpolate subpixels. We use bilinear interpolation for the square lattice and use linear interpolation using three pixels for the hexagonal lattices. The latter case is described as follows. On a hexagonal lattice, any point can be enclosed by three pixels that form an equilateral triangle. We perform linear interpolation for a point in the triangle on a hexagonal lattice using these three pixel values.

4. EXPERIMENTS AND RESULTS

4.1. Image constructions and experiments

We generate input images of image alignment by two ways. First, we simulate directly captured image on the hexagonal or square lattices by using mathematically defined functions and picking the true values on each lattice. Each region of interest (ROI) of these images is shown in Figure 2(a1, a2). Second, we consider downscaled image from an existing large image on a square lattice. To calculate a pixel value on the downscaled image, we average over all pixels that are included in the corresponding area on the original large image. The ROIs of these reconstructed images are shown in Figure 2(b1, b2). The latter case of image construction is frequently used for running image alignment of large images on the square lattice. We examine it on the hexagonal and square lattices.

On the square lattice, the pixel number of the ROI is $100 \times 100$. On the hexagonal lattice Hex(W) and Hex(A), the pixel number of the ROIs are, respectively, $100 \times 115$ and $93 \times 107$. For each image, we generate image pairs by warping the image source. The warping is calculated by displacing the four vertices of the ROI by a set of eight values that were generated randomly using zero-mean Gaussian white noise with a standard deviation $\sigma = 1$. The unit of $\sigma$ is the distance...
Fig. 2. Input images clipped by ROI for ICIA. (a1) and (a2) are generated from the mathematically defined function. (b1) and (b2) are reconstructed from a large captured image. Images only for the square lattice are shown.

Fig. 3. Mean of point position errors for all trials at 100th iteration step in ICIA. The hexagonal lattices outperformed the square lattice. The order of accuracy (first is better) was Hex(W), Hex(A), Ando, Sobel.

Fig. 4. Success rate for each threshold. Each figure corresponds to each input image in Figure 2. Horizontal axes show thresholds. Each vertical axis indicates the success rate.

4.2. Accuracy and success rate
We evaluated the accuracy of the image alignment by averaging point position errors for all trials. Each error is the dis-
distance between the ground truth of the warped four vertices and the resultant position of the four vertices that is computed from estimated homography. In the template image, the four vertices correspond to the four corners of the ROI. We compared the mean of errors at the 100th step of the iteration in ICIA. The results for accuracy are presented in Figure 3. The results showed that the hexagonal lattices outperformed the square one. On the hexagonal lattices, Hex(W) was better than Hex(A). On the square lattice, Ando was slightly better than Sobel.

We evaluated the success rate for each result of ICIA trials. We define the success rate as the ratio of successful trials to all trials where a successful trial for given threshold is that achieved smaller point position error than the threshold at least once in the iteration process in ICIA algorithm. Figure 4 shows the success rates of respective cases for given thresholds. The evaluation of the success rate showed that the hexagonal lattices outperformed the square one, and the order of better success rate was Hex(W), Hex(A), and the square lattices. The success rate of Ando and Sobel were similar.

5. DISCUSSION AND CONCLUSION

We compared and evaluated the square lattice and hexagonal lattices of two sizes with respect to image alignment using ICIA. A hexagonal lattice Hex(W) has the same pixel width with the square lattice and another hexagonal lattice Hex(A) has the same pixel area with the square lattice. To compare and evaluate, we simulated and implemented images on these lattices by using a mathematically defined function or reconstruction from a larger captured image. We used the Sobel and Ando operators for the square lattice, and the consistent gradient operator of radius 1 for the hexagonal lattices. The results reflect the hexagonal lattices outperformed the square one, and the order of accuracy as point position error (first is better) was Hex(W), Hex(A), Ando, Sobel. The superiority of Hex(W) is reasonable because it has higher resolution than other cases; however, the Hex(A) also achieved higher accuracy than the square cases for its isotropy. On a square lattice, the Ando and Sobel operators achieved almost identical accuracy.

Regarding the success rate, the results showed the hexagonal lattices outperformed the square one, and Hex(W) achieved higher success rate than Hex(A). In the results on the square lattice, the Ando and Sobel operators showed an equivalent success rate for all input images.

We conclude that the hexagonal lattices of two sizes are superior to the square lattice for image alignment with respect to accuracy and the success rate. On the hexagonal lattices, the lattice of the same pixel width as the square one outperforms the lattice of the same pixel area as the square one because of the higher resolution. The results show that more accurate image alignment will be available by using hexagonal image sensors that have the same pixel width or area as the sensor of the square lattice. The results also show that image alignment on downsampled images is improved by converting an existing large image onto the square lattice into the smaller image on either the hexagonal lattices of both the sizes instead of using only the square lattice. Importantly, using the hexagonal lattice of the same pixel area as square one even improves image alignment while maintaining the resolution. Consequently, we showed there is a possibility for improving accuracy of image alignment by converting large image into the smaller image of hexagonal pixels, instead of that of square pixels, before running image alignment.

6. REFERENCES


