Color Filter in SIFT Matching

HAN Yechen, Ritsumeikan University, gr0119xr@ed.ritsumei.ac.jp
Shinichi HIRAI, Ritsumeikan University

SIFT matching is a quite robust matching approach for the usage of robust SIFT descriptor. In order to overcome the disadvantages brought by gray images, many color SIFT methods have been proposed. However those methods mainly focused on the improvement of SIFT descriptors. The excessive number of feature points, especially those useless points on background rather than a target, on the other hand, is not paid enough attention yet. This paper proposes an approach to improve the efficiency in SIFT matching. Using color filter to eliminate feature points with different color from those on a target. Therefore much calculation on descriptors and matching could be reduced so that the efficiency could be improved.
Key words: computer vision, SIFT matching, color filter

1. Introduction

SIFT has been proved to be a quite robust local invariant feature descriptor and widely used in computer vision. SIFT matching based on these robust descriptors also provided us with an excellent performance. SIFT algorithm is developed mainly to deal with gray images, however color in images also provide valuable information. To make use of color information, numbers of color SIFT algorithm has been developed [1].

HSV-SIFT[2] computes SIFT descriptors over all three channels of HSV color space, which gives 3×128 dimensions per descriptor. HSV-SIFT descriptor increases the complex of descriptor but has no invariance properties.

HueSIFT[3] introduced a concatenation of the hue histogram with SIFT descriptor. HueSIFT descriptor, compared with HSV-SIFT, is scale-invariant and shift-invariant.

OpponentSIFT[4] makes use of opponent color space with SIFT. The information in the O_3 channel is equal to the intensity information, while the other channels describe the color information in the image. OpponentSIFT descriptor is invariant to changes in light intensity.

CSIFT[5] introduced color invariance as input for SIFT algorithm and based on opponent color space. Because of the division by intensity, the scaling in the diagonal model will cancel out, making CSIFT scale-invariant with respect to light intensity, but not shift-invariant.

Transformed color SIFT[6] applied same normalization to the RGB channels as for the transformed color histogram. And descriptor is computed for each channel. This descriptor is scale-invariant, shift-invariant and invariant to light color changes and shift.

RGB-SIFT[6] calculates SIFT descriptor for each RGB channel independently, just like HSV-SIFT. Also, an interesting property of this descriptor is that its descriptor values are equal to the transformed color SIFT.

All these approaches introduced color information to build various descriptors for SIFT and improved the performance of color image processing. However the efficiency is not so optimistic. The robustness of SIFT matching originates from two parts: the robust descriptor and the excessive number of feature points. The focus on descriptor provides robustness on color images, but the size of descriptor and transformation will cost extra time so that the result might not so efficient. In this paper, we use color information in a different approach. Considering that the number of feature points has a great influence on time cost in SIFT matching. This paper proposes to use color information to filter feature points so that the number could be reduced. Meanwhile using original SIFT descriptor for matching to improve the efficiency of SIFT matching.

2. SIFT Algorithm and Implementation

2.1 Review of SIFT algorithm

SIFT algorithm, as described in Reference [7], consists of four major stages:

(1) Scale-space extrema detection: The first stage of computation searches over all scales and image locations. It is implemented efficiently by using a difference−of−Gaussian (DoG) function to identify potential interest points that are invariant to scale and orientation.

(2) Keypoint localization: At each candidate location, a detailed model is fit to determine location and scale. Keypoints are selected based on measures of their stability.

(3) Orientation assignment: One or more orientations are assigned to each keypoint location based on local image gradient directions. All future operations are performed on image data that has been transformed relative to the assigned orientation, scale, and location for each feature, thereby providing invariance to these transformations.

(4) Keypoint descriptor: The local image gradients are measured at the selected scale in the region around each keypoint. These are transformed into a representation that allows for significant levels of local shape distortion and change in illumination.

2.2 Implementation based on Rob Hess library

The implementation of SIFT matching is based on the Rob Hess library [8], which is divided into five stages: (1) Construction of Gaussian pyramid; (2) Feature points detection; (3) Scales and orientation; (4) Descriptor computation; (5) Feature matching (using voting algorithm). Time assignment is shown in Table 1.

Tab. 1 Time assignment of SIFT matching

Gaussian pyramid	Feature points	Scales and Orientation	Descriptor	matching	Total
0.638s	0.717s	<0.001s	1.064s	0.436s	2.856s

This time assignment is based on the default parameter setting of Rob Hess library. In the first stage, the time spent on the construction of Gaussian pyramid would not change no matter how parameters change. Time cost in the third stage is quite small. Therefore these two stages could be out of consideration. On the other hand, the number of feature points affects the descriptor computation and feature matching almost linearly, and the second stage, feature points detection, is the key stage to decide the number of feature points. Considering the large number of useless feature points on background, the reduction of those feature points could provide us with more efficient feature points and focus on the target.

3. Color Filter in feature points detection

During the second stage, feature points detection, two parameters are used as filter: contrast threshold and curve threshold. These two filters allow points to be rejected that have low contrast (and are therefore sensitive to noise) or are poorly localized along an edge. In this paper, we introduced a third filter, color filter, to provide a further reduction of the feature points.

3.1 Principle of color filter

A basic SIFT matching would find out all the feature points from the target image and the input images (images with target on backgrounds). During the process of the feature point detection in the target image, feature points were detected as usual. While at the same time, color filter collects the color information of all feature points and then a range of color (RoC) could be created. Then the process goes on to the input image. After filtered by contrast and curve threshold, color filter would judge if the color of a candidate feature point is in RoC. Thus those feature points with different color from the target could be eliminated. The flow chart of feature point detection is shown as Fig. 1.

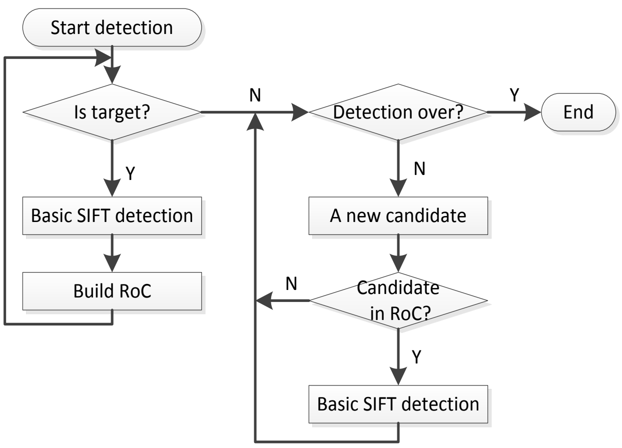

Fig. 1 Flow chart of feature point detection using color filter

3.2 Color Space

For color filter, the RoC (range of color) is quite important because the accuracy of feature point's elimination depends on this range. Two commonly used color space have been evaluated in this paper: RGB color space and HSV color space.

RGB color space RGB color space is the simplest approach to build a RoC. All images are in RGB so that transform is unnecessary. Vector $\boldsymbol{C}=[r,g,b]^T$ is used to represent the color information of a point. To build the RoC of target image, all color information vectors $\boldsymbol{C}_i=[r_i,g_i,b_i]^T$ are recorded. Then the expectation $\boldsymbol{E}(\boldsymbol{C})$ and variance $Var(\boldsymbol{C})$ could be calculated:

$$\boldsymbol{E}(\boldsymbol{C}) = \frac{1}{n}\sum_{i=1}^{n}\boldsymbol{C}_i = [r_E, g_E, b_E]^T \quad (1)$$

$$Var(\boldsymbol{C}) = \frac{1}{n}\sum_{i=1}^{n}(\boldsymbol{C}_i - \boldsymbol{E})^T(\boldsymbol{C}_i - \boldsymbol{E}) \quad (2)$$

Here we select $Var(\boldsymbol{C})$ as the threshold of color filter. When the process applies to the input image, firstly the distance $D^2(\boldsymbol{E}(\boldsymbol{C}), \boldsymbol{C})$ between $\boldsymbol{E}(\boldsymbol{C})$ and color vector $\boldsymbol{C}$ of a candidate feature point is computed. If the distance $D^2(\boldsymbol{E}(\boldsymbol{C}), \boldsymbol{C})$ (a squared distance to fit $Var(\boldsymbol{C})$ and to save processing time) is smaller than $Var(\boldsymbol{C})$, the color of this candidate is in the RoC, so this candidate could be selected as a feature point.

HSV color space HSV, which stands for hue, saturation and value (also called HSB, B for brightness), is considered more similar to human sense on color performance. Hue, like spectrum of light, defines what a color is (Fig. 2).

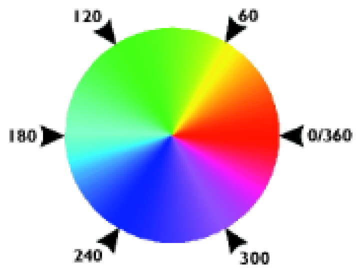

Fig. 2 Hue scale

Considering the change of saturation or value in color image, we use hue channel only to build the RoC. During the process of target image, RGB vector is firstly transferred to HSV vector with following formula (hue only).

$$h = \begin{cases} 0, & \text{if } \max = \min \\ 60\times\dfrac{g-b}{\max-\min}, & \text{if } \max = r \text{ and } g \ge b \\ 60\times\dfrac{g-b}{\max-\min}+360, & \text{if } \max = r \text{ and } g < b \\ 60\times\dfrac{b-r}{\max-\min}+120, & \text{if } \max = g \\ 60\times\dfrac{r-g}{\max-\min}+240, & \text{if } \max = b \end{cases} \quad (3)$$

Also considering that target might have various colors so that RoC would also be quite huge, expectation and variance are not used in HSV filter. Besides, HSV space provides an accurate definition of what a color is so one or more main color bands together are selected to build the RoC. Main color bands could be selected by voting algorithm or manually setting. When feature point detection is running on input images, similar to RGB filter, the candidate feature points with different color from RoC would be discarded so that the number of feature points could be reduced and the efficiency of SIFT matching could be improved.

4. Experimental Evaluation

To evaluate the proposed approach, we use a juice pack as target to move over three different backgrounds. Juice pack is simple in shape but has various colors. Three backgrounds include simple background, background with similar color, and complex background in similar color.

4.1 Simple background

The first experiment used newspaper as background, which owns simple and different color from target. In Fig. 3 (a), many feature points detected on the background are considered as "useless points". Much time is wasted on the descriptor calculation and matching on these points. On average, basic SIFT matching cost about 2.898 seconds, and over 1000 feature points are detected with about 10-20 match pairs. Fig. 3 (b) is the situation RGB filter. Obviously many points on background have been eliminated but still some of them remains. On average, RGB filter saved about 28% total matching time and detected feature points are reduced to about 330, with about 12 match pairs. Fig. 3 (c) is the result after HSV filter is applied. The background is quite clear and the feature points detected are focused on our target. On average, HSV filter saved about 25.5% total matching time and the number of feature points has been reduced to less than 200, with about 15-20 match pairs.

(a) Basic SIFT matching

(b) Using RGB filter

(c) Using HSV filter

Fig. 3 Experimental result of simple background

Table 2 Evaluation in simple background

	Feature point number	Matching number	Total time cost (s)
Basic SIFT	1064	14	2.898
RGB filter	327	11	2.055
HSV filter	186	16	2.159

4.2 Similar color in background

The second experiment used a little more complex background with similar color (orange and yellow). Because of the similar color in background, feature points on orange and yellow background remained from color filter, as shown in Fig. 4(b) and Fig. 4(c), but those on while background were eliminated as expected. The effectiveness was weakened by the similar color in background but color filter still works. Detailed data is listed in Table 3.

(a) Basic SIFT matching

(b) Using RGB filter

(c) Using HSV filter

Fig. 4 Experimental result of background with similar background

Table 3 Evaluation in background with similar color

	Feature point number	Matching number	Total time cost (s)
Basic SIFT	550	12	1.897
RGB filter	207	7	1.822
HSV filter	216	7	1.809

4.3 Complex background in similar color

The third experiment is done using a complex background in similar color. Such complex backgrounds have a lot of feature points, also in our example many of them have a similar color to the target. Thus color filter does not work as effectively as in former experiments, as shown in Fig. 5. Huge number of feature points in similar color lead to an opposite result. Color filter did eliminate some feature points, but the dealt with color cost too much time that cannot be covered, as shown in Table 4.

(a) Basic SIFT matching

(b) Using RGB filter

(c) Using HSV filter

Fig. 5 Experimental result of complex background

Table 4 Evaluation in background with similar color

	Feature point number	Matching number	Total time cost (s)
Basic SIFT	2503	31	2.976
RGB filter	1148	4	4.599
HSV filter	1498	8	3.485

5. Conclusion

In this paper, we introduced color filter into SIFT matching process, specifically, in the stage of feature point detection. By applying color filter, feature points would be eliminated (which are considered as useless points) before the calculation of descriptor and matching process, and thus the efficiency could be improved. Experimental evaluation proved that when the color in background is quite different from that in target, the best profit could be achieved (Section 4.1). Even if the background has similar color to target, color filter still works (Section 4.2), unless the background is too complex and has a similar color to target (Section 4.3).

References

[1] Koen E.A van de Sande, Theo Gevers, and Cees G.M Snoek, “Evaluating Color Descriptors for Object and Scene Recognition”, IEEE Trans. *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 32, no. 9,pp. 1582-1596, Sep. 2010.

[2] A. Bosch, A. Zisserman, and X. Muñoz, “Scene Classification Using a Hybrid Generative/Discriminative Approach,” *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 30, no. 4,pp. 712-727, Apr. 2008.

[3] J. van de Weijer, T. Gevers, and A. Bagdanov, “Boosting Color Saliency in Image Feature Detection,” *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 28, no. 1, pp. 150-156, Jan. 2006.

[4] T. Tuytelaars and K. Mikolajczyk, “Local Invariant Feature Detectors: A Survey,” *Foundations and Trends in Computer Graphics and Vision*, vol. 3, no. 3, pp. 177-280, 2008.

[5] A.E. Abdel-Hakim and A.A. Farag, “CSIFT: A SIFT Descriptor with Color Invariant Characteristics,” Proc. *IEEE Conf. Computer Vision and Pattern Recognition*, pp. 1978-1983, 2006.

[6] J.M. Geusebroek, R. van den Boomgaard, A.W.M. Smeulders, and H. Geerts, “Color Invariance,” *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 23, no. 12, pp. 1338-1350, Dec. 2001.

[7] D.G. Lowe, “Distinctive Image Features from Scale Invariant Keypoints,” *Int’l J. Computer Vision*, vol. 60, no. 2, pp. 91-110, 2004.

[8] http://robwhess.github.com/opensift/