Color-Ciratefi: A color-based RST-invariant template matching algorithm

Sidnei Alves de Araújo[1,2] and Hae Yong Kim[1]
[1] Escola Politécnica - Universidade de São Paulo – Brazil
[2] Diretoria dos Cursos de Informática - Universidade Nove de Julho – Brazil
{saraujo, hae}@lps.usp.br

Abstract—Template matching is a technique widely used for finding patterns in digital images. An efficient template matching algorithm should be able to detect template instances that have undergone geometric transformations. Similarly, a color template matching should be able to deal with color constancy problem. Recently we have proposed a new grayscale template matching algorithm named Ciratefi, invariant to rotation, scale, translation, brightness and contrast. In this paper we introduce the Color-Ciratefi that takes into account the color information. We use a new similarity metric in the CIELAB space to obtain invariance to brightness and contrast changes. Experiments show that Color-Ciratefi is more accurate than C-color-SIFT, the well-known SIFT algorithm that uses a set of color invariants. In conventional computers, Color-Ciratefi is slower than C-color-SIFT. However Color-Ciratefi is more suitable than C-color-SIFT to be implemented in highly parallel architectures like FPGA, because it repeats exactly the same set of operations for each pixel.

Keywords - template matching; Ciratefi; RST-invariant; color.

I. INTRODUCTION

Color provides high discriminative power. However, most existing template matching techniques were designed for gray-level images [1]. The main problem of color template matching is the color constancy, that is, how to extract color information that remains constant with the illumination change. Changes in illumination can cause changes in object colors acquired by a camera, worsening the performance of pattern recognition algorithms that use color information [2].

Tsai and Tsai [1] presented a technique for matching colored objects, called “color ring-projection”. It uses color features derived from HSI and CIELAB color spaces and is able to detect objects in different conditions of illumination. The main drawback of this technique is the lack of invariance to scale changes.

Geusebroek et al. [3] developed an important set of color invariant features based on Gaussian derivative to deal with illumination changes, shadow, highlights and noise. These set of invariants has been embedded in the well known SIFT (Scale Invariant Features Transform) [4], one of the most efficient methods to extract invariant features from images, yielding a powerful color invariant descriptor [5, 6]. Actually, many color invariants reported in the literature have been plugged in the SIFT, generating many color-based SIFT descriptors such as CSIFT [5], HSV-SIFT, Hue-SIFT, OpponentSIFT, W-SIFT, rgSIFT, Transformed color SIFT [7], SIFT-CCH [8], W-color-SIFT, H-color-SIFT and C-color-SIFT [6].

In this paper we introduce a new color template matching algorithm named Color-Ciratefi, based on our Ciratefi technique (Circular, Radial and Template-Matching Filter) [9]. Color-Ciratefi is invariant to rotation, scaling and translation (RST), and robust to minor viewpoint variations and blur. We propose a color similarity metric designed to be robust to contrast and brightness changes. We did experiments using a dataset of color images with different geometric and photometric transformations [10] and compared the results with those obtained by C-color-SIFT algorithm [6] that combines SIFT descriptor with a set of color invariants proposed by Geusebroek et al. [3]. Experiments show that Color-Ciratefi is more accurate than C-color-SIFT.

In conventional computers, Color-Ciratefi is slower than C-color-SIFT. However, Color-Ciratefi is more suitable than C-color-SIFT to be implemented in highly parallel architectures like FPGA, because Color-Ciratefi repeats exactly the same set of simple operations for each pixel [11]. FPGA implementation of the first Ciratefi filter (Ciratefi has three cascaded filters) can process a 640×480 frame in only 1.06ms [11], far faster than SIFT implementation in FPGA that processes a 320×240 frame in 30ms [12]. Kim [13] developed another RST-invariant template matching based on Fourier transform of the radial projections named Forapro. Forapro is faster than Ciratefi in a conventional computer but the latter seems to be more fit to be implemented in FPGA than the former.

II. CIRATEFI TECHNIQUE

Ciratefi is a grayscale template-matching algorithm composed by three steps of filtering that successively excludes pixels that have no chance of matching the query template [9, 11]. Let A be the grayscale image to be analyzed and T the query grayscale template. The goal of Ciratefi is to find all occurrences of T in A, with respective orientation angle and scale (Fig. 1). The instances of T in A may appear rotated, scaled, shifted and with diverse brightness and contrast. Below we present a brief description of Ciratefi.

A. First step - Circular Sampling Filter (Cifi)

Cifi uses the projections of the images *A* and *T* on a set of circles (Fig. 1b) to detect the "first grade candidate pixels". For each candidate pixel, the "probable scale factor" is also computed. Given an image to be analyzed *A* and a set of *l* radii $\{r_0, r_1, ..., r_{l-1}\}$, a 3D image $C_A[x,y,k]$ is built as:

$$C_A[x,y,k] = \frac{1}{P_k}\sum_{\theta=0}^{P_k-1} A\left(x + r_k \cos\frac{2\pi\theta}{P_k}, y + r_k \sin\frac{2\pi\theta}{P_k}\right) \quad (1)$$

where $0 \le k < l$, and $P_k = \text{round}(2\pi r_k)$. $C_A[x,y,k]$ is the average grayscale of pixels of *A* on the circle ring with radius r_k centered at (x, y).

Given the query template *T* and a set of *n* scales $\{s_0, s_1, ..., s_{n-1}\}$, *T* is resized, generating the resized templates T_0, T_1 … T_{n-1}. Each template T_i is circularly sampled according to the set of radii yelding a matrix of multi-scale rotation-invariant features C_T with *n* rows (scales) and *l* columns (radii):

$$C_T[i,k] = \frac{1}{P_k}\sum_{\theta=0}^{P_k-1} T\left(x_0 + r_k \cos\frac{2\pi\theta}{P_k}, y_0 + r_k \sin\frac{2\pi\theta}{P_k}\right) \quad (2)$$

where (x_0, y_0) is the central pixel of *T* and $0 \le i < n$. In other words, $C_T[i,k]$ is the average grayscale of pixels of template *T* at scale s_i on the circle ring with radius r_k. In small scales, some of the outer circles may not fit inside the resized templates. These circles are represented by a special value in table C_T (say, -1) and are not used to compute the correlations.

Matrices C_A and C_T are used to detect the circular sampling correlation (CisCorr) at the best matching scale for each pixel (x,y):

$$\text{CisCorr}_{A,T}(x,y) = \underset{i=0}{\overset{n-1}{\text{MAX}}}\left[\,\text{Corr}(C_T[i], C_A[x,y])\,\right] \quad (3)$$

where $\text{Corr}(\boldsymbol{a},\boldsymbol{b})$ is the normalized cross correlation coefficient between vectors ***a*** and ***b***.

A pixel (x,y) is classified as a first grade candidate pixel if $\text{CisCorr}_{A,T}(x,y) \ge t_1$, for some threshold t_1. The probable scale of (x,y) is s_i, where *i* is the argument that maximizes CisCorr.

B. Second step - Radial sampling filter (Rafi)

This step uses projections of images *A* and *T* on a set of radial lines (Fig. 1c) to upgrade some of the first grade candidate pixels to the second grade. Rafi also estimates the probable rotation angle for each second grade candidate pixel. The length of the radial lines ($\lambda = r_{l-1}s_i$) is calculated according to the largest circle radius r_{l-1} and the probable scale s_i computed by Cifi.

For each first grade candidate pixel (x,y) of image *A*, the matrix R_A is computed, considering a set of *m* angles $(\alpha_0, \alpha_1, ..., \alpha_{m-1})$, as follows:

$$R_A[x,y,j] = \frac{1}{\lambda}\sum_{t=0}^{\lambda} A(x + t\cos\alpha_j, y + t\sin\alpha_j) \quad (4)$$

In other words, $R_A[x,y,j]$ is the average grayscale of pixels of *A* on the radial line with one vertex at (x,y), length λ and inclination α_j. Then *T* is radially sampled yielding a vector R_T with *m* features:

$$R_T[j] = \frac{1}{r_{l-1}}\sum_{t=0}^{r_{l-1}} T(x_0 + t\cos\alpha_j, y_0 + t\sin\alpha_j) \quad (5)$$

Then, Rafi computes the correlation RasCorr between the vectors $R_A[x,y]$ and R_T at the best matching rotation angle:

$$\text{RasCorr}_{A,T}(x,y) = \underset{j=0}{\overset{m-1}{\text{MAX}}}\left[Corr\left(R_A[x,y], \text{cshift}_j(R_T)\right)\right] \quad (6)$$

where "cshift$_j$" denotes circular shifting *j* positions of the argument vector. A first grade pixel (x,y) is upgraded to the second grade if $\text{RasCorr}_{A,T}(x,y) \ge t_2$, for some threshold t_2. The probable rotation angle of pixel (x,y) is α_j where *j* is the argument that maximizes RasCorr.

It seems possible to estimate the local rotation angle using some rotation-discriminating feature, as in [1, 4, 13]. Using this idea, the second filter (Rafi) can be completely eliminated. However, rotation-discriminating feature may not be applicable in some situations (as at the very center of symmetrical shapes like "H", "O", etc.). Rafi solves this problem.

C. Third step – Template matching filter (Tefi)

This step filters the second grade candidate pixels using a conventional template matching with correlation coefficient as metric. This task is fast because Cifi and Rafi computed the probable scale and angle for each candidate pixel. Tefi computes the correlation coefficient using a threshold t_3, to evaluate how well each candidate pixel of the second grade matches the template.

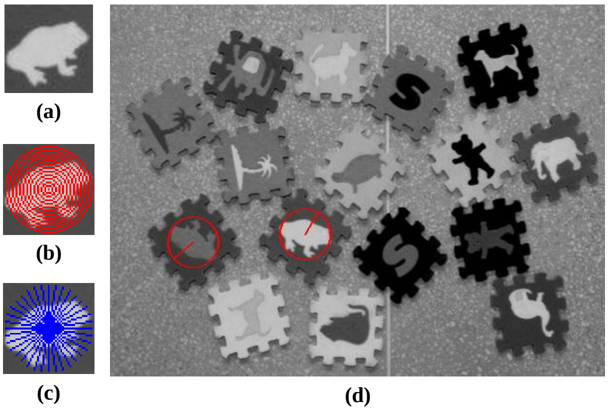


Figure 1. Detection of the template "frog" by Ciratefi. (a) Query template *T*. (b) Circular projections. (c) Radial projections. (d) The circles indicate the matching positions (the radii represent the scales and the pointers represent the angles).

III. COLOR CIRATEFI

The goal of Color Ciratefi is: given a pair of color images *A* and *T*, detect all the instances of *T* that appear in *A*. The instances of *T* in *A* can be affected by different geometric and photometric transformations such as scale, rotation, minor viewpoint variations, blur and illumination.

To deal with illumination changes we propose a similarity measure that uses CIELAB color space. Color-Ciratefi consists of the same three cascaded filters as Ciratefi, using the proposed similarity measure instead of the correlation coefficient.

*A. CIE L*a*b* and the similarity measure*

The CIE L*a*b* (CIELAB) color space was designed to be perceptually uniform [2], that is, a small perturbation to a color value produces a change of about the same perceptual importance across the range of all colors. Moreover, CIELAB isolates the lightness L* from the chromaticity a*b*. So, this color space is especially suited to evaluate the similarity of two image patches, invariant to brightness and contrast changes. In CIELAB space, the lightness L* varies from 0 to 100. The range of chromaticity components a*b* depends on the original color space of the image. If the original color space is RGB, one can assume the range -100 to +100.

The Euclidean distance or some other perceptual distance is typically used as the similarity measure in perceptual color spaces as L*a*b* [14, 15]. We proposed a similarity measure that uses a weighted composition of chromaticity and lightness components to evaluate the perceptual similarity between two color feature vectors, robust to brightness and contrast changes.

Let $\boldsymbol{x} = \{x_1, x_2, \ldots, x_n\}$ and $\boldsymbol{y} = \{y_1, y_2, \ldots, y_n\}$ be two vectors of colors. Each component x_i or y_i is composed by a set of tristimulus values L*, a* and b* and are denoted, respectively, by x_{iL}, x_{ia}, x_{ib} and y_{iL}, y_{ia}, y_{ib}. For similarity of chromaticity (S_C) the Euclidean distance of components a* and b* is used:

$$S_C(\boldsymbol{x}, \boldsymbol{y}) = 1 - \frac{\sum_{i=1}^{n} \sqrt{(x_{ia} - y_{ia})^2 + (x_{ib} - y_{ib})^2}}{200 \cdot \sqrt{2} \cdot n} \qquad (7)$$

The distance is subtracted of one to obtain the similarity measure.

To measure the similarity of intensity (S_I), the correlation coefficient was employed, since it is invariant to brightness and contrast changes:

$$S_I(\boldsymbol{x}, \boldsymbol{y}) = \frac{\sum_{i=1}^{n} (x_{iL} - \bar{\boldsymbol{x}}_L)(y_{iL} - \bar{\boldsymbol{y}}_L)}{\sqrt{\sum_{i=1}^{n} (x_{iL} - \bar{\boldsymbol{x}}_L)^2} \sqrt{\sum_{i=1}^{n} (y_{iL} - \bar{\boldsymbol{y}}_L)^2}} \qquad (8)$$

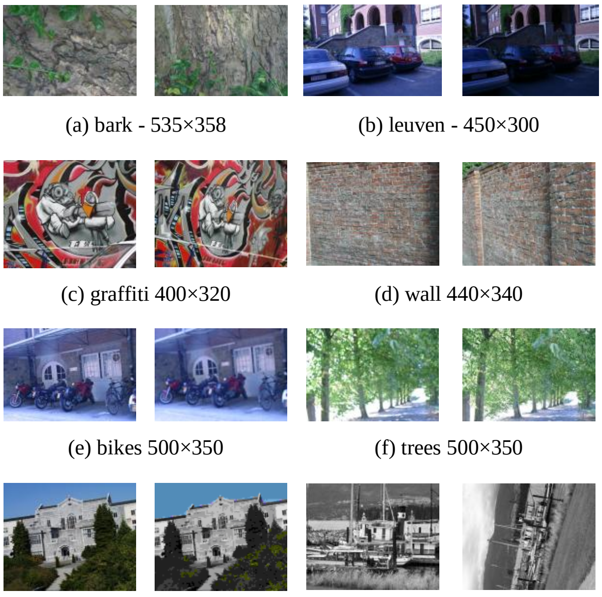
where $\bar{\boldsymbol{x}}_L$ and $\bar{\boldsymbol{y}}_L$ are, respectively, the mean lightness of $\boldsymbol{x}$ and $\boldsymbol{y}$. The proposed similarity measure (Sim) is a weighted geometric mean of S_C and S_I:

$$\mathrm{Sim}(\boldsymbol{x}, \boldsymbol{y}) = [S_C(\boldsymbol{x}, \boldsymbol{y})]^{\alpha} \cdot [S_I(\boldsymbol{x}, \boldsymbol{y})]^{\beta} \qquad (9)$$

where α and β are the weights attributed to chromaticity and intensity similarities (usually, $\alpha+\beta=1$). For example, if $\alpha=0$ only the correlation of intensity values is considered. On the other hand, if $\beta=0$ only the chromaticity information is taken into account. We use weighted geometric mean (instead of weighted arithmetic mean) because either complete chromaticity dissimilarity or complete lightness dissimilarity represents complete dissimilarity of the two patches.

IV. EXPERIMENTAL RESULTS

In order to evaluate Color Ciratefi, a set of real-world color images with different geometric and photometric transformations was used [10]. It consists of 42 images divided in 7 subsets, labelled by the authors as *bark*, *leuven*, *graffiti*, *wall*, *bikes*, *trees* and *ubc* (Fig. 2). There is a 8th subset named *boat*, but it was discarded because it contains only grayscale images (Fig. 2h). Each subset is composed of 6 images of the same scene with gradual geometric or photometric transformation: rotation and scale changes (Fig. 2a); illumination changes (Fig. 2b); viewpoint changes (Fig. 2c and 2d); image blur (Fig. 2e and 2f) and JPEG compression (Fig. 2g). This image database has been used to evaluate the performance of many other color descriptors, such as [10, 16].



(a) bark - 535×358

(b) leuven - 450×300

(c) graffiti 400×320

(d) wall 440×340

(e) bikes 500×350

(f) trees 500×350

(g) ubc - 400×320

(h) boat - unused subset

Figure 2. Images used for evaluating Color Ciratefi. The images were reduced to 70% (bark subset images) and 50% (all other images) of their original sizes.

In the experiments, we extracted randomly 20 templates with 60×60 pixels from the first image of each subset, and searched for them in the 6 images of the same subset. In total, we did 840 templates matchings (20×6×7). For evaluation purpose, we compare Color Ciratefi with C-color-SIFT, using the best color descriptor reported in the paper [6].

C-color-SIFT is truly scale-invariant. However, Color Ciratefi needs a pre-specified scale range. We used *n*=9 (0.3 to 1.1) scales in all the Color Ciratefi experiments. We used also the following Color Ciratefi parameters: number of circles *l*=21, number of angles *m*=36. Cifi chose the 2000 best matching pixels as the first grade candidates, Rafi promoted 300 of them as the second grade candidates, and finally Tefi chose the best matching position. We chose the weights α=0.8 and β=0.2 in the similarity measure (eq. 9) to maximize the accuracy, as depicted in Table 1. In all experiments, the performance of algorithms is given in terms of recall: TP/(TP+FN), where TP is True Positive and FN is False Negative.

TABLE 1. EXPERIMENTS VARYING α AND β IN COLOR CIRATEFI, TO CHOOSE THE OPTIMAL PARAMETERS.

Parameters α and β	Possible Matches	TP	FN	Recall
0.9 ; 0.1	840	627	213	0.746
0.8 ; 0.2	**840**	**631**	**209**	**0.751**
0.7 ; 0.3	840	599	241	0.713

TABLE 2. ACCURACY EVALUATION – COLOR CIRATEFI × C-COLOR-SIFT.

Image Subset	Possible Matches	Color Ciratefi			C-color-SIFT		
		TP	FN	Recall	TP	FN	Recall
Bark	120	87	33	**0.73**	83	37	**0.69**
Leuven	120	94	26	**0.78**	76	44	**0.63**
Graffiti	120	74	46	**0.62**	48	72	**0.40**
Wall	120	61	59	**0.51**	94	26	**0.78**
Bikes	120	108	12	**0.90**	76	44	**0.63**
Trees	120	109	11	**0.91**	94	26	**0.78**
UBC	120	98	22	**0.82**	58	62	**0.48**
Total	840	631	209	**0.75**	529	311	**0.63**

The comparison between color Ciratefi and C-color-SIFT is depicted in Table 2. In average, Color Ciratefi is more precise than C-color-SIFT. For images with changes in scale/rotation (Bark), illumination (Leuven), affected by blur (Bikes and Trees) and JPEG compression (UBC), Color Ciratefi outperforms C-color-SIFT.

In blurred images (bikes and trees), the weak performance of C-color-SIFT can be attributed to the small amount of keypoints extracted by SIFT descriptor. In the case of JPEG compression (UBC), a large amount of inconsistent features arises from the artifacts introduced by the JPEG compression, leading to erroneous SIFT keypoint matchings. The only subset where C-color-SIFT is more accurate than Color Ciratefi is in the Wall subset. Actually, neither Color Ciratefi nor C-color-SIFT is designed to deal with viewpoint changes.

C-color-SIFT is faster than Color Ciratefi in conventional computer. It took, in average, 5s to detect a template while color Ciratefi took 28s, both using a 2.8GHz Pentium-4. However, Ciratefi can be accelerated thousand of times via hardware FPGA implementation, achieving more than real-time performance [11]. Meanwhile, the SIFT can only be accelerated tens of times and the hardware implementation is far more intricate [12].

V. CONCLUSIONS

In this paper we have presented a robust color-based template matching named Color Ciratefi. We have proposed a new similarity metric for color images. We compared the performance of Color Ciratefi with C-color-SIFT algorithm using a dataset of natural images containing different geometric and photometric transformations. Surprisingly, the experiments have shown that Color Ciratefi is more accurate than C-color-SIFT, although the latter uses a powerful set of color invariants.

ACKNOWLEDGMENTS

One of the authors (SAA) acknowledges the financial support from Universidade Nove de Julho.

REFERENCES

[1] D. M. Tsai and Y. H. Tsai, “Rotation-invariant pattern matching with color ring-projection,” Pattern Recognition, vol. 35, n. 1, pp. 131-141, January 2002.

[2] T. Moerland and F. Jurie, Learned color constancy from local correspondences, In: Proc. of ICME, pp. 820-823, July 2005.

[3] J. M. Geusebroek, R. van den Boomgaard, A. W. M. Smeulders and H. Geerts, “Color invariance,” IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 23, n. 12, pp.1338-1350, December 2001.

[4] D. G. Lowe, Object recognition from local scale-invariant features, In: Proc. of ICCV, pp. 1150–1157, September 1999.

[5] A. E. Abdel Hakim and A. A. Farag, CSIFT: A SIFT descriptor with color invariant characteristics, In: Proc. of CVPR06, pp. 1978-1983, 2006

[6] G. J. Burghouts and J. M. Geusebroek, “Performance evaluation of local colour invariants, Computer Vision and Image Understanding, v. 113, n.1, pp. 48-62, January 2009.

[7] K. E. A. van de Sande, T. Gevers and C. G. M. Snoek, Evaluation of color descriptors for object and scene recognition. In: Proc. of CVPR, pp. 1-8, June 2008.

[8] C. Ancuti and P. Bekaert, SIFT-CCH: Increasing the SIFT distinctness by color co-occurrence histograms. In: 5th International Symposium on Image and Signal Processing and Analysis, pp. 130-135, September 2007.

[9] H. Y. Kim and S. A. de Araújo, “Grayscale template-matching invariant to rotation, scale, translation, brightness and contrast,” LNCS, vol. 4872, p. 100-113, December 2007.

[10] K. Mikolajczyk and C. Schmid, “A performance evaluation of local descriptors,” PAMI, vol. 27, n. 10, pp. 1615-1630, October 2005.

[11] H. P. A. Nobre and H. Y. Kim. Automatic VHDL generation for solving rotation and scale-invariant template matching in FPGA. In: V Southern Programmable Logic Conference, São Carlos, Brazil, pp. 229-231, April 2009.

[12] V. Bonato, E. Marques and G. A. Constantinides, “A Parallel Hardware Architecture for Scale and Rotation Invariant Feature Detection,” IEEE T. Circuits and Systems for Video Tech., vol. 18, n. 12, pp. 1703-1712, December 2008.

[13] H. Y. Kim, “Rotation-Discriminating Template Matching Based on Fourier Coefficients of Radial Projections with Robustness to Scaling and Partial Occlusion,” Pattern Recognition, vol. 43, n. 3, pp. 859-872, March 2010.

[14] B. Bascle, O. Bernier and V. Lemaire, “Illumination-invariant color image correction,” LNCS, vol. 4153, pp. 359-368, August 2006.

[15] A. Gijsenij, T. Gevers and M. P. Lucassen. “A perceptual comparison of distance measures for color constancy algorithms,” LNCS, vol. 5302, pp.208-221, October 2008

[16] Y. Cui, N. Hasler, T. Thormählen and H. P. Seidel, “Scale invariant feature transform with irregular orientation histogram binning,” LNCS, vol. 5627, pp. 258-267, July 2009.