Shape matching using keypoints extracted from both the foreground and the background of binary images

Houssem Chatbri[1], Kenny Davila[2], Keisuke Kameyama[3] and Richard Zanibbi[4]
[1] Graduate School of Systems and Information Engineering, Department of Computer Science, University of Tsukuba, Japan
[24] Department of Computer Science Rochester Institute of Technology Rochester, NY, USA
[3] Faculty of Engineering, Information and Systems, University of Tsukuba, Japan
e-mails: [13]{chatbri, kame}@adapt.cs.tsukuba.ac.jp [24]{kxd7282, rlaz}@rit.edu

***Abstract*— We introduce a descriptor for shape feature extraction and matching using keypoints that are extracted from both the foreground and the background of binary images. First, distance transform (DT) is applied on the image after contour detection. Then, connected components (CCs) of pixels having the same intensity are extracted. Keypoints correspond to centers of mass of CCs. A keypoint filtering mechanism is applied by estimating the spatial stability of keypoints when successive iterations of image blurring and binarization are applied. Finally, features are extracted for each keypoint using a round layout which radius is set depending on the keypoint's location. We evaluate our descriptor using datasets of silhouette images, handwritten math expressions, and logos. Experimental results show that our descriptor is competitive compared with state-of-the-art methods, and that keypoint filtering is effective in reducing the number of keypoints without compromising matching performances.**

***Keywords*— Shape matching, local descriptors, keypoints, binary images, distance transform.**

I. Introduction

Shape matching is a vibrant area of research on image analysis due to the numerous applications it allows [1]. Particularly, when dealing with binary images where color and texture information are absent (e.g. silhouette images, scanned documents, sketches, etc.), shape is the only available feature to be used for image representation and matching [2].

Numerous methods have been presented for shape feature extraction in binary images [3][4]. Usually, images are subjected to contour detection or skeletonization before using a shape descriptor in order to remove redundant information and reduce processing time [5]. Moreover, some methods select certain *keypoints* and use them to extract features [6][7][8]. In these cases, keypoints are selected based on their saliency or by using uniform sampling from the shape contours.

In this work, we introduce a shape descriptor for binary images based on the extraction of keypoints. The proposed descriptor applies distance transform (DT) on the image after contour detection. This generates a grayscale image where the intensity of each pixel indicates its distance to the nearest foreground pixel. Then, local maxima on the DT image are extracted and they result in connected components having the same grayscale intensity. The centers of mass of these connected components correspond to keypoints. Afterwards, keypoint filtering is performed to detect stable keypoints and filter out keypoints caused by noise and contour perturbations. Finally, a keypoint-dependent round layout is used to extract features for each keypoint.

The keypoints extracted using DT are in locus of symmetry between foreground pixels. We anticipate the significance of such keypoints in shape matching due to the importance of symmetry as a cue for recognition in human perception [9] and as a characteristic of patterns that has been used for image retrieval [10].

The proposed descriptor is evaluated using silhouette images of the Kimia 216 dataset [11], handwritten mathematical expressions of Zanibbi and Yu's dataset [12], and logo images of the Tobacco 800 dataset [13]. Comparison with existing methods shows that our descriptor is competitive and that keypoint filtering reduces the number of keypoints without compromising performances.

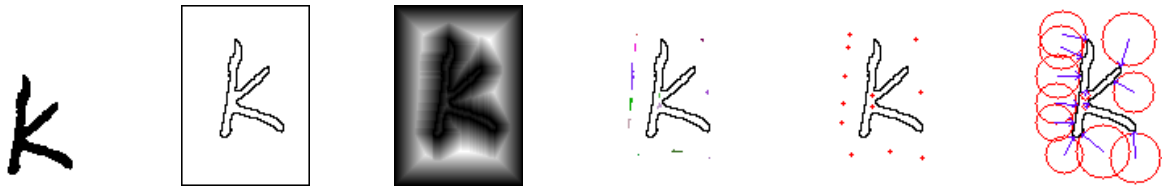
The remainder of this paper is organized as follows: Sec. II reviews key methods of shape matching. We present our descriptor in Sec. III and evaluate it in Sec. IV. Concluding remarks and future work are presented in Sec. V.

II. Related work

Research on shape matching has led to a large depository of methods [3] where shape descriptors can be classified into several categories including methods using global and local features [4], contour-based and skeleton based methods [5], and methods using keypoints [6][7][8].

Global methods extract features using the coarse information of the shape, and hence do not convey much information about the local details. Such methods include shape signatures [14], Fourier descriptors [15], and angular partitioning [16]. Global methods are robust against noise but on the detriment of representing fine details. On the other hand, other methods integrate local neighborhoods of the shape points, which makes them capable of capturing fine details of the shape. Such methods include curvature scale space (CSS) [6], shape contexts [7], and variations of binary local patterns [17].

Contours and skeletons have been used as an intermediate representation before feature extraction. Contours are more robust against noise than skeletons, as skeletons tend to generate noisy branches and artifacts in presence of shape border perturbations [5]. On the other hand, skeletons are more suitable in applications that require the segmentation of the



(a) (b) (c) (d) (e) (f)

Fig. 1. Keypoint extraction steps: (a) Original binary image. (b) Image after normalization. (c) DT image. (d) Local maxima connected components. (e) Keypoints ($k = 11$). (f) Keypoint vectors ($\alpha = 1$): Circle radii correspond to the keypoint distance from the nearest contour point, and arrows show the orientation of the vector delimited by the keypoint and its nearest contour point.

original object into constituent parts for graph-based feature representation [18].

Other methods sample a number of keypoints from the shape contour. Keypoint sampling is done based on their saliency or by uniform sampling of contour points. CSS uses scale space filtering [19] to extract inflection points from closed contours [6]. Then, the contour deformation and merging of inflection points caused by scale space filtering are used for feature extraction. High curvature points of the contour have also been used as keypoints [20]. On the other hand, shape contexts perform uniform keypoint sampling from the shape contours without special consideration about the keypoints curvature or location [7][8].

Keypoints extracted using scale-space filtering in the well-known SIFT descriptor have been very successful when applied on intensity images [21]. However, it has been shown that SIFT keypoints are suboptimal compared to keypoints that are uniformly sampled from the shape contours when using complex binary images of Maya hieroglyphs [22]. This result is due to the absence of local changes of intensity in binary images that hinders scale-space filtering from detecting distinctive keypoints and attributing them characteristic scales. In contrast, our keypoints are extracted using DT and scale-space filtering is used as a way to detect stable keypoints by monitoring their location change during the image filtering.

III. The Proposed Descriptor

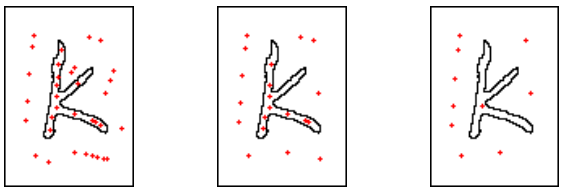
Our descriptor extracts keypoints using DT (Sec. III-A). Then, it applies keypoint filtering in order to filter out unstable keypoints (Sec. III-B). Finally, features are extracted using keypoint-dependent round layouts (Sec. III-C).

A. Keypoint extraction

Keypoints are extracted as follows: First, the original image (Fig. 1(a)) is normalized by applying contour detection, then detecting the bounding box of the contour and using it to generate a larger image with the object shifted towards the center (Fig. 1(b)). The width W_N and height H_N of the normalized image are as follows:

$$W_N = W_{BB} \times 1.5 + 20, \quad H_N = H_{BB} \times 1.5 + 20 \qquad (1)$$

where W_{BB} and H_{BB} are the dimensions of the object's bounding box. A frame of 1-pixel-width is added to the normalized image in order to avoid extracting local maxima from the borders.



(a) $k = 5$ (b) $k = 9$ (c) $k = 15$

Fig. 2. Effect of the parameter k on the number of keypoints.

Next, distance transform (DT) is applied to generate a grayscale image where the intensity of each background pixel corresponds to its L_1 distance from the nearest foreground pixel (Fig. 1(c)). Then, local maxima are detected on the DT image using a $k \times k$ square window. This generates connected components with the same pixel intensity (Fig. 1(d)). Finally, keypoints are extracted as centers of mass of the connected components (Fig. 1(e)).

Local maxima detection is done using a $k \times k$ square window located at each DT image pixel. The parameter k affects the number of extracted local maxima. The larger k gets, the fewer keypoints are detected (Fig. 2).

B. Keypoint filtering

The extracted keypoints may be not be all necessary as some of them might be caused by noise and contour perturbations. We observe that keypoints which maintain stable locations under local image distortion are more distinctive than keypoints that move when image local distortion is applied.

In this step, we implement a keypoint filtering mechanism using scale space filtering [19]. The proposed mechanism produces successive blurred images using the Gaussian filter which spread function is defined as follows:

$$g(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-(x+y)^2/2\sigma^2} \qquad (2)$$

where σ is the smoothing parameter that controls the scale, and x and y are pixel coordinates. Then, the filtered images are binarized using Otsu's algorithm [23]. This mechanism is used to produce N increasingly distorted images that are used for keypoint extraction (Fig. 3).

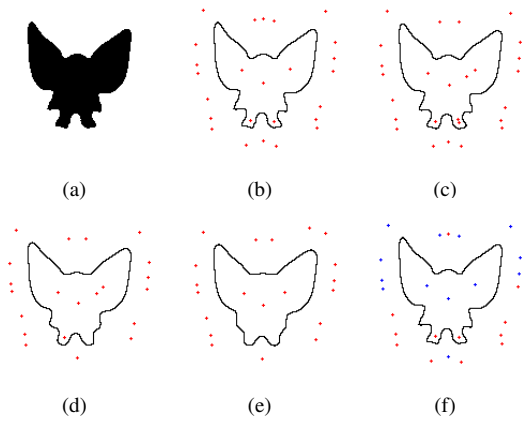


Fig. 3. Keypoint filtering: (a) Original image. (b) Keypoints for $\sigma = 0.5$. (c) Keypoints for $\sigma = 1.5$. (d) Keypoints for $\sigma = 3.5$. (e) Keypoints for $\sigma = 4.5$. (f) Stable Keypoints highlighted in blue for $k = 11$, $\theta = 2$, and $\delta = 3$.

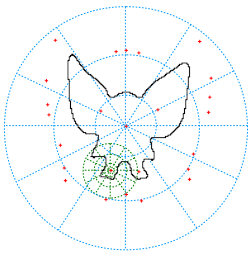


Fig. 4. Feature extraction layout. A layout is defined for each keypoint and the layout's radius is proportional to the distance between the keypoint and its nearest contour point.

Afterwards, a *measure of stability* θ is assigned to keypoints extracted from the original image. θ is equal to the number of blurring iterations during which the keypoint remains in a stable location. A keypoint is considered stable at *iteration* i if its location p_i does not move beyond a neighborhood $\delta \times \delta$ from his previous location p_{i-1} at *iteration* $i-1$. Fig. 3(f) shows keypoints that are stable for at least $\frac{4}{10}$ of $N = 5$ iterations and for $\delta = 3$.

C. Feature represenation and matching

The last step is to generate a feature vector to each keypoint k. For this purpose, we use a round layout which radius r_k is proportional to the distance between the keypoint k and its closest contour point (Fig. 4):

$$r_k = \alpha \times \min_{j \leq N_C} |kp_j| \tag{3}$$

where α is a constant, p_j is a contour point of index j, and N_C is the total number of contour points.

Then, a histogram h_k is extracted by calculating the distribution of contour points in distance and angle sections. The distance between two histograms is expressed by the $\mathcal{X}^2$ statistic:

$$\mathcal{X}^2(h_1, h_2) = \frac{1}{2} \sum_{k=0}^{K-1} \frac{[h_1(k) - h_2(k)]^2}{h_1(k) + h_2(k)} \tag{4}$$

where K is the number of bins in a keypoint histogram.

The dissimilarity d between two images I_1 and I_2 is estimated by the cumulative minimum distance between the images' keypoint histograms:

$$d(I_1, I_2) = \frac{1}{N_1} \sum_{i=0}^{N_1-1} \min_{0 \leq j < N_2} \{\mathcal{X}^2(h_i^1, h_j^2)\} \tag{5}$$

where N_1 and N_2 are the number of keypoints in I_1 and I_2. $d(I_1, I_2)$ is asymmetric. Therefore, we express the distance between two images I_1 and I_2 as follows:

$$D(I_1, I_2) = \frac{d(I_1, I_2) + d(I_2, I_1)}{2} \tag{6}$$

$D \in [0, 1]$. The smaller $D(I_1, I_2)$ is, the more similar I_1 and I_2 are.

The feature vector is translation-invariant due to using the object's bounding box for image normalization, and scale-invariant due to using keypoint-dependent feature extraction layouts. Rotation-invariance can be insured by using the orientation of the vector delimited by the keypoint and its nearest contour point as a reference orientation (Fig. 1(f)).

IV. Experimental results

We evaluate our descriptor using Kimia's dataset of silhouette images [11], Zanibbi and Yu's dataset of handwritten mathematical expressions [12], and the logo set from Tobacco 800 dataset [13]. We downscale Zanibbi and Yu's and Tobacco logo images, which have larger sizes than silhouette images, by a factor of 2 to reduce the number of keypoints for the sake of efficient experimental performance. Fig. 5 and Table 1 show samples from the datasets and information about the images. Silhouette images are neat comparing to handwritten mathematical expressions and logo images. Handwritten mathematical expressions contain significant handwriting fluctuations and component displacement and alteration. Logo images are taken from scanned documents and they are the noisiest compared to the two other datasets. Silhouette images contain single component objects, while handwritten mathematical expressions and logo images contain multi-component objects.

The goal of the experiments is to evaluate the descriptor's matching performances and the role of its parameters. The effect of varying the parameters k, δ and θ is investigated, and the performance of the descriptor is evaluated using the *precision at n* metric, denoted $P@n$, that is defined as follows:

$$P@n = \frac{|\{n \ retrieved \ images\} \cap \{relevant \ images\}|}{|\{n \ retrievd \ images\}|} \tag{7}$$

The larger $P@n$ is, the better matching performances are.

A desirable descriptor should extract a reduced number of keypoints, yet remain distinctive. For this purpose, we also evaluate the compactness of the descriptor using a *compression* metric that is equal to the number of extracted keypoints relative to the number of contour points. The smaller *compression* is, the more compact the descriptor is. Both $P@n$ and *compression* are expressed in percentages.

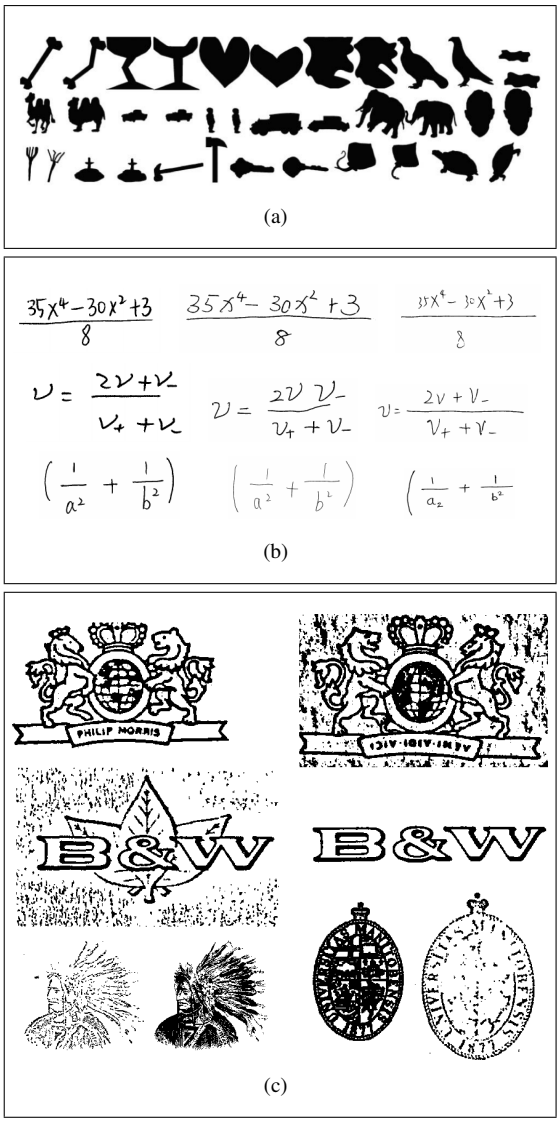


Fig. 5. Samples of the datasets: (a) shows images from Kimia's dataset [11], (b) shows images from Zanibbi and Yu's dataset [12], and (c) shows image from Tobacco 800 logo dataset [13].

Table 1. Information about the datasets images.

Dataset	# images	# classes	# instance per class	# contour points
Kimia	216	18	12	494
Zanibbi and Yu	200	20	10	2275
Tobacco	412	35	[1, 68]	2300

The algorithm's setting are as follows: The number of iterations used for keypoint filtering is $N = 5$. The number of histogram sections for the feature descriptor is 5 distance sections and 12 angle sections. We set the number of retrieved images n as query-dependent and equivalent to the number of the query's class instances. The constant for setting the keypoint-dependent feature layout radius is set heuristically $\alpha = 1.5$ in order to insure that the feature extraction layout's bins of one unit distance can reach a sufficient number of pixels, which are the ones located at a close distance to the keypoint relative to the distance to its closest contour point. Image blurring is done using a Gaussian filter of scale $\sigma_{min} = 0.5$ that is incremented with $\Delta\sigma = 0.5$.

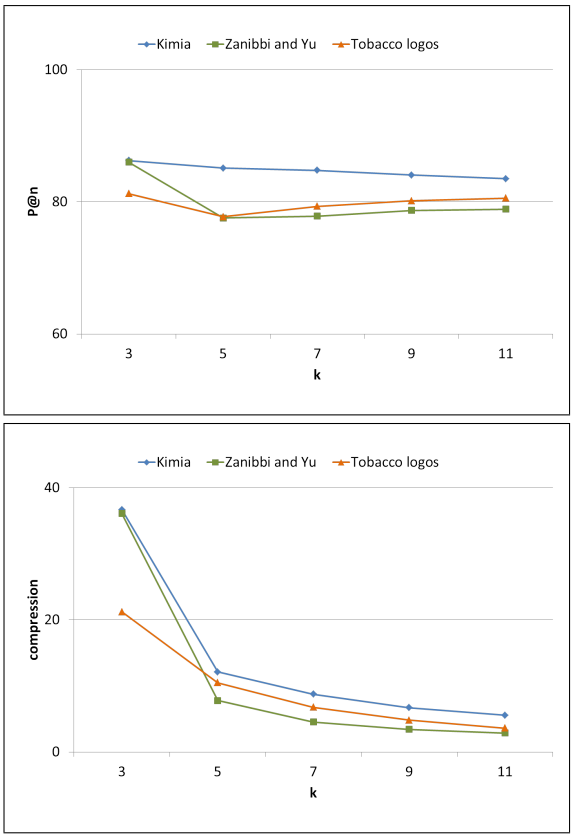


Fig. 6. Effect of varying the parameter k on $P@n$ and $compression$ (the y-axis interval is cropped for better visualization).

A. Evaluation of the parameter k for setting the size of the local maxima detection window

The parameter k defines the size of the local maxima detection window during keypoint extraction. We evaluate the role of this parameter on matching performances and descriptor compactness. During this experiment, the keypoint filtering step is omitted and all the extracted keypoints are used regardless of their stability. Fig. 6 shows curves of $P@n$ and $compression$ as functions of k. For all datasets, the best matching performances correspond to $k = 3$. $compression$ rates for $k = 3$ were 36.67% for silhouette images, 36.1% for handwritten mathematical expressions, and 21.22% for logo images, which correspond to 181, 821, and 488 keypoints on average respectively.

According to the results of this experiment, we set $k = 3$ empirically and use it in the following experiments.

B. Evaluation of the parameter δ for setting the maximum keypoint location shift

The parameter δ defines the maximum location shift allowed for a stable keypoint to move during keypoint filtering. In this experiment, we investigate the effect of varying this parameter on retrieval performances and descriptor compactness.

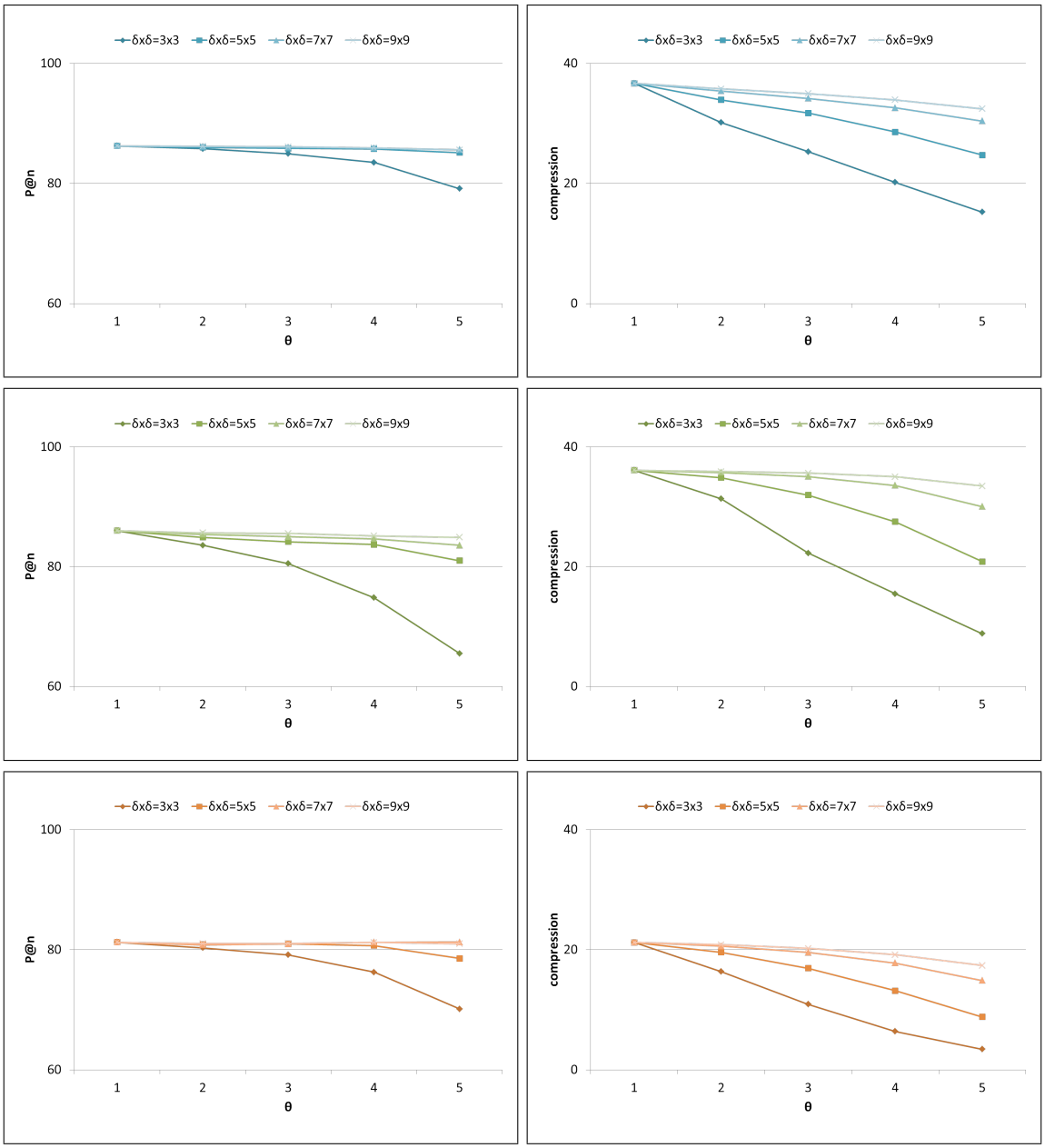


Fig. 7. Effect of varying the parameters δ and θ on $P@n$ and $compression$ (the y-axis interval is cropped for better visualization). The first row corresponds to Kimia's dataset, the second row corresponds to Zanibbi and Yu's dataset, and the third row corresponds to Tobacco logo dataset.

Fig. 7 shows curves of $P@n$ and $compression$ as functions of the stability threshold θ, and for $\delta \in \{3, 5, 7, 9\}$. For all datasets, using a strict $\delta = 3$ leads to the most significant compression. Up to $\theta = 3$, matching performances remain roughly equal for different values of δ. Then, for $\theta \geq 4$, a decrease in performances is observed, noticeably for $\delta = 3$.

The best trade-off between keypoints minimization and matching performance preservation corresponds to $\delta = 5$ and $\theta = 3$. In this case, matching performances correspond to $P@n = 85.84\%$ for Kimia's dataset, 84.15% for Zanibbi and Yu's dataset, and 80.99% for Tobacco 800 logo dataset, while compactness performances are $compression = 31.74\%$ for Kimia's dataset, 31.95% for Zanibbi and Yu's dataset, and 16.92% for Tobacco 800 logo dataset, that is using 157, 727, and 389 keypoints on average respectively.

C. Comparative evaluation

The proposed descriptor is compared with existing methods using the three datasets. In case of Kimia's dataset, we calculate the retrieval performance metric reported in several published papers, that is the number of relevant retrieved images for each of the top 6 ranks and the percentage calculated by summing these numbers. Comparison is done with Support Regions Descriptor (SRD) [17], Shapes Context (SC) [7], and Path Similarity Skeleton Graph Matching (PSSG) [18]. As for the other datasets, we use the $P@n$ performance metric and

Table 2. Retrieval results using Kimia's dataset.

Algorithm	1^{st}	2^{nd}	3^{rd}	4^{th}	5^{th}	6^{th}	Total
SRD [17]	216	199	188	181	176	167	86.96%
SC [7]	214	209	205	197	191	178	92.12%
Our descriptor	216	208	204	193	196	193	93.36%
PSSG [18]	216	216	215	216	213	210	99.22%

Table 3. $P@n$ values using Zanibbi and Yu's dataset and Tobacco 800 logo dataset.

Algorithm	Zanibbi and Yu	Tobacco 800 logos
SRD [17]	47.6%	82.55%
Our descriptor	86.0%	81.25%

we use SRD for comparison. SC and PSSG are omitted here because they need special considerations regarding keypoint sampling and matching of multi-component images.

Tables 2 and 3 show the best performances achieved by our descriptor ($k = 3$ and $\theta = 1$) and performances of other methods. Our descriptor yields competitive performances on Kimia's dataset, outperforming shape context and support regions descriptor (SRD). Our descriptor significantly outperforms SRD in case of Zanibbi and Yu's dataset, and it is slightly outperformed by SRD in case of Tobacco 800 logos.

The lowest performances of our descriptor were observed when using the Tobacco 800 logo dataset. This is explained by the significant noise and intra-class variations in this dataset (Fig. 5) and given that no preprocessing for noise reduction has been applied. Here, the noise pixels and connected components affect the bounding box's dimensions and location during the normalization step, which causes shifting in the locations of keypoints and generates false ones (i.e. keypoints that are caused by noise).

V. Conclusions and future work

In this paper, we introduce a descriptor for shape matching using keypoints that are selected from both foreground and background pixels. The descriptor is translation-invariant due to using the object's bounding box for image normalization, and scale-invariant by using keypoint-dependent feature extraction layouts. Rotation-invariance can be insured by using the orientation of the vector delimited by the keypoint and its nearest contour point as a reference orientation.

We evaluate our descriptor using three datasets of silhouette images, handwritten mathematical expressions, and logo images. Experimental results and comparison with other methods indicate that our descriptor has competitive matching performances. Moreover, the keypoint filtering step is effective in reducing the number of keypoints without compromising matching performances.

We identify several directions for improving and extending our research: Different approaches for keypoint filtering and feature representation can be tried. Using different image normalization strategies that focus on connected components instead of the whole image can be useful for partial image matching. In this case, specific preprocessing should be envisaged to overcome incorrect components disconnectedness or merging, which can cause variations in the keypoints' locations and feature vectors. Related to this, the lower performance when using the Tobacco 800 logo dataset reveals that specific considerations should be done when handling noisy images. We aim to make the setting of the number of image distortion iterations N and the selection of the keypoint stability parameter θ automatic, and evaluate the generality of our descriptor using large scale datasets. Color images can be used for such an evaluation by applying edge detection prior to using DT.

References

[1] M. Breuß *et al.*, *Innovations for shape analysis: models and algorithms*. Springer Science & Business Media, 2013.

[2] S. Liang and Z. Sun, "Sketch retrieval and relevance feedback with biased SVM classification," *Pattern Recognition Letters*, 2008.

[3] M. Yang, K. Kpalma, and J. Ronsin, "A survey of shape feature extraction techniques," *Pattern recognition*, pp. 43–90, 2008.

[4] D. Zhang and G. Lu, "Review of shape representation and description techniques," *Pattern recognition*, vol. 37, no. 1, pp. 1–19, 2004.

[5] H. Chatbri, K. Kameyama, and P. Kwan, "A comparative study using contours and skeletons as shape representations for binary image matching," *Pattern Recognition Letters*, 2015.

[6] F. Mokhtarian, S. Abbasi, and J. Kittler, "Robust and efficient shape indexing through curvature scale space," in *British Machine and Vision Conference (BMVC)*, vol. 96, 1996.

[7] S. Belongie, J. Malik, and J. Puzicha, "Shape matching and object recognition using shape contexts," *IEEE PAMI*, 2002.

[8] E. Roman-Rangel and S. Marchand-Maillet, "Hoosc128: A more robust local shape descriptor," in *Pattern Recognition*, vol. 8495 of *Lecture Notes in Computer Science*, pp. 172–181, Springer, 2014.

[9] C. W. Tyler, *Human symmetry perception and its computational analysis*. Psychology Press, 2002.

[10] S. Lee, "Symmetry-driven shape description for image retrieval," *Image and Vision Computing*, vol. 31, no. 4, pp. 357–363, 2013.

[11] T. Sebastian, P. Klein, and B. Kimia, "Recognition of shapes by editing shock graphs," in *International Conference on Computer Vision*, vol. 1, pp. 755–755, IEEE Computer Society, 2001.

[12] R. Zanibbi and L. Yu, "Math spotting: Retrieving math in technical documents using handwritten query images," in *International Conference on Document Analysis and Recognition (ICDAR)*, 2011.

[13] G. Zhu and D. Doermann, "Automatic document logo detection," in *International Conference on Document Analysis and Recognition (ICDAR)*, pp. 864–868, 2007.

[14] E. R. Davies, *Machine vision: theory, algorithms, practicalities*. Elsevier, 2004.

[15] D. Zhang and G. Lu, "A comparative study of fourier descriptors for shape representation and retrieval," in *Asian Conferernce on Computer Vision (ACCV)*, pp. 646–651, 2002.

[16] A. Chalechale, G. Naghdy, and A. Mertins, "Sketch-based image matching using angular partitioning," *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, 2005.

[17] H. Chatbri, K. Kameyama, and P. Kwan, "Sketch-based image retrieval by size-adaptive and noise-robust feature description," in *International Conference on Digital Image Computing: Techniques and Applications (DICTA)*, pp. 1–8, IEEE, 2013.

[18] X. Bai and L. J. Latecki, "Path similarity skeleton graph matching," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 30, no. 7, pp. 1282–1292, 2008.

[19] A. P. Witkin, "Scale-space filtering: A new approach to multi-scale description," in *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, vol. 9, pp. 150–153, IEEE, 1984.

[20] G. V. Pedrosa *et al.*, "Image feature descriptor based on shape salience points," *Neurocomputing*, 2013.

[21] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, 2004.

[22] E. Roman-Rangel and S. Marchand-Maillet, "Shape-based detection of maya hieroglyphs using weighted bag representations," *Pattern Recognition*, vol. 48, no. 4, pp. 1161–1173, 2015.

[23] N. Otsu, "A threshold selection method from gray-level histograms," *Automatica*, vol. 11, no. 285-296, pp. 23–27, 1975.