PicWords: Render a Picture by Packing Keywords

Zhenzhen Hu, Si Liu, *Member, IEEE*, Jianguo Jiang, Richang Hong, *Member, IEEE*, Meng Wang, *Member, IEEE*, and Shuicheng Yan, *Senior Member, IEEE*

***Abstract*—In this paper, we propose a novel text-art system: input a source picture and some keywords introducing the information about the picture, and the output is the so-called PicWords in the form of the source picture composed of the introduction keywords. Different from traditional text-graphics which are created by highly skilled artists and involve a huge amount of tedious manual work, PicWords is an automatic non-photorealistic rendering (NPR) packing system. Given a source picture, we first generate its silhouette, which is a binary image containing a Yang part and a Yin part. Yang part is for keywords placing while the Yin part can be ignored. Next, the Yang part is further over-segmented into small patches, each of which serves as a container for one keyword. To make sure that more important keywords are put into more salient and larger image patches, we rank both the patches and keywords and construct a correspondence between the patch list and keyword list. Then, mean value coordinates method is used for the keyword-patch warping. Finally, certain post-processing techniques are adopted to improve the aesthetics of PicWords. Extensive experimental results well demonstrate the effectiveness of the proposed PicWords system.**

***Index Terms*—Calligram, keywords, non-photorealistic rendering, picture, PicWords.**

I. Introduction

WHEN people see a beautiful picture, say, a photo of Audrey Hepburn, they may want to know more information about her. Since the picture itself cannot tell such details as the woman's birth date, nationality, experiences, etc., they may have to resort to other sources of information, for example, texts from the Internet. On the other hand, if people read an introduction text about Audrey Hepburn, they may also long to see what she looks like directly. This kind of dilemma is very common in our daily life. So, can we design a system to generate an image, which contains Audrey's face, and at the same time, seamlessly embeds a brief summarization of her biography? To this end, we design an automatic system called PicWords, shown in Fig. 1. It is composed of Audrey's face as well as some keywords to briefly introduce her. In the PicWords, two modalities (picture and keywords) are seamlessly fused together to better represent the human/object of interest. The two modalities complement each other and both contribute to more exquisite art effect: the image is more intuitive and the text is more content-rich. To the best of our knowledge, PicWords is the first attempt to design a text-art to combine an image and its keywords in multimedia research area.

Fig. 1. (a) the source picture: The image of famous movie star Audrey Hepburn, (b) the keywords, (c) the target image: PicWords. For better viewing of all figures in this paper, please see original zoomed-in color pdf file.

Our work is inspired by seeing many advertisement posters which contain actor portraits and the introduction words. These posters are very attractive in real lives. However, they are often hand-made. Therefore, we would like to explore how to automatically generate this kind of posters. PicWords can be applied in many scenarios. For example, PicWords can generate a portrait for a user in any social networking website, where one's personal photo and one's blog (or twitter, tags etc) can be combined together as a PicWords to better describe the user. Currently, some websites[1] have integrated some kinds of word/tag cloud system. But only text cloud can't give people an intuitive feeling about what the person looks like. PicWords can also be used to design posters and advertisements.

PicWords can be considered as a kind of non-photorealistic rendering (NPR) [1]. More specifically, it belongs to *NPR Packing*, depicting an image by automatically arranging a collection of small pictorial elements [2]. Artists and art lovers have always been fascinated by the interplay between a whole and its parts. The earliest NPR Packing art can be traced back to Roman mosaic where small squares of colored glass conspire to form a detailed scene. If the small cooperating elements are limited to only words or letters, it has another terminology: *calligram*. Calligrams enjoy a rich tradition and a wide variety of styles limited only by the artists' imagination. Obviously, PicWords is a kind of calligram.

Manuscript received January 20, 2013; revised June 30, 2013 and September 24, 2013; accepted December 31, 2013. Date of publication February 11, 2014; date of current version May 13, 2014. This work was supported in part by State Key Development Program of Basic Research of China 2013CB336500; in part by the NSFC under Grant nos. 61272393, 61322201, the Program for New Century Excellent Talents in University under grant NCET-12-0836, and the Open Project Program of the National Laboratory of Pattern Recognition (NLPR); and in part by the Singapore National Research Foundation under its International Research Centre @Singapore Funding Initiative and administered by the IDM Programme Office. The work was performed when Z. Hu was visiting National University of Singapore. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Weisi Lin.

Z. Hu, J. Jiang, R. Hong, and M. Wang are with the Hefei University of Technology, Hefei 230009, China.

S. Liu (corresponding author) and S. Yan are with the National University of Singapore, Singapore (e-mail: dcslius@nus.edu.sg).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TMM.2014.2305635

[1]http://www.weibo.com/

The most fascinating part of PicWords is that human visual system (HVS) is attracted to the two modalities (first picture and then keywords) sequentially during the human perception process. At the first glance, human will be attracted to PicWords' overall contour, which corresponds to the **picture** modality. In Fig. 1, general people will immediately recognize that it is a picture of Audrey Hepburn. But if people take a closer look at the PicWords, they will gradually notice its elements, i.e., the **keywords**. For example, in Fig. 1, people will then pay more attention to the keywords, including "film", "tiffany", "Hollywood", etc. In sum, the viewer is caught in a dynamic tension between attending to the image and to the keywords that make it up.

As aforementioned, PicWords is a natural combination of both picture and keywords. The whole generation process contains a picture-only module, a keywords-only module, a cross-modality picture & keywords module and the final post-processing module. The outputs of the picture module are the ranked image patches, which serve as containers for the outputs of the keywords module, i.e., ranked keywords. In the following picture & keywords module, the two modalities are fused together by warping each word to its corresponding patch. Finally, post-processing techniques are adopted to further refine the PicWords.

The rest of the paper is organized as follows. In Section II, we briefly introduce the related work. Then, the whole system framework is illustrated in Section III. Next, in Section IV, more detailed step by step introduction of the system is presented. The experimental results are shown later in Section V and finally we conclude the paper in Section VI.

II. Related Work

These days, increasing efforts have be devoted into non-photorealistic rendering (NPR) [3] and especially its sub-category named NPR Packing. It addresses the arrangement of a multitude of small tiles to form artistic representation to enhance multimedia presentations and has multiple applications such as [4] and [5]. According to different kinds of cooperating elements styles, NPR packing can be further classified into mosaicking and calligrams [3]. Mosaicking (a.k.a tiling or stippling) aims to reappear the image using a medium, packing image regions with a multitude of atomic rendering elements. Besides mosaicking, some artists have also found the charm of the combination of text words and images and developed the second art style called calligram. Calligram is a rendering of a large target image by arranging a collection of small text/words, often in an array, each chosen specifically to fit a particular block of the target image. There are also other works related to the topic of improving the visual aesthetics of multimedia contents such as [6] and [7].

1) Mosaicking: There are many representative works of mosaicking. Hausner *et al.* [8] simulated the appearance of Roman mosaics. It is the first work to address irregular tile shapes through an energy minimization scheme for shape packing. In their approach, Lloyd's method was applied to Voronoi diagrams via the Manhattan metric, giving an arrangement of oriented rectangular mosaic tiles. Kim and Pellacini [9] presented Jigsaw image mosaics (JIM) approach using an active contour based optimization scheme to minimize the energy function that traded off among various measures of the packing's quality. Orchard and Kaplan [10] described a fast technique for mosaicking images with irregular tiles, also capable of cropping partial regions from the image database to use as tiles. Although great success has been achieved in mosaicking, viewer cannot read the content related with the target image. To the contrary, PicWords is much more informative than mosaicking by seamlessly inserting the keywords as the components into the target image. Therefore, viewers can sense how the image looks like as well as read a brief introduction of the key object inside the image.

2) Calligrams: Computer aided design of text-based art-forms has been explored in a number of different contexts. One well-known example is ASCII art,[2] a technique of composing pictures with printable text characters. In ASCII art textual and numeric characters are only the means to build an image; that is, single characters are not meant to convey meaning but to be perceived as components to form a whole. The work of Xu [11] showed how to generate structure-based ASCII art by analysis of contour structures. Nacenta [12] developed a technique called FatFonts based on Arabic numerals. This enables accurate reading of the numerical data while preserving an overall visual context. The drawback shared by the above mentioned systems is that no relationship exists between the target image and its components (text or Arabic number). Also, comparing with ASCII art and its variants, PicWords can convey more information via packing keywords.

Xu and Kaplan [2] developed a solution of packing letter forms, a specific case of irregular tiling. They divided up a target region into pieces and warped a letter into each piece. But their method can only process letters, and thus cannot convey enough information. However, PicWords can contain much richer information. Stylization through text packing was later considered by Maharik and Sheffer [13]. They presented an algorithm for creating digital micrography images, created from minuscule text. Their main focus is to stitch the words together to resemble a source image, and their words are usually too small to be recognized. To the contrary, PicWords can make sure that most keywords are recognizable.

III. System Overview

In this section, we give an overview of the whole PicWords system. As shown in Fig. 2, the whole system contains four modules: picture, keywords, picture & keywords and post-processing.

The fist part is *picture* module. Given a source image, we first generate its silhouette image. In the silhouette image, background and trivial details are filtered and only the important patches are kept. At the same time, the original color image is segmented into several small patches with the state-of-the-art super-pixel segmentation algorithm [14]. Since we are only interested in filling the foreground area, only patches covered by the silhouette image are kept. After that, all the remaining patches are ranked according to how much they are suitable as a keyword container. Usually, central and long patches are supposed to rank higher.

[2]http://en.wikipedia.org/wiki/ASCII_art

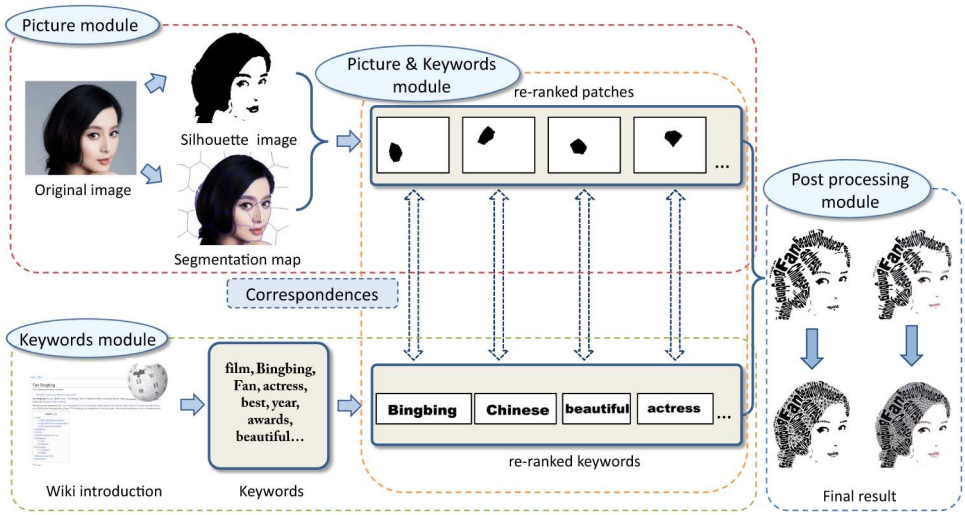

Fig. 2. The flowchart of PicWords contains four modules: picture, keywords, picture & keywords and post-processing modules. The source image is fed into the picture module to generate a segmentation map and a silhouette image, both of which collaboratively generate some patches. The patch list is re-ranked. A keyword list is obtained from the keywords module. Then we construct a correspondence between the patch list and keyword list and fit a keyword into its mapped patch. Finally, in the post-processing module, some colorization and symbol filling techniques are adopted to improve the visual effects of PicWords.

Fig. 3. Silhouette image generation process. (a) original image, (b) filtering background, (c) segmented foreground, (d) binary map, (e) final silhouette image after smoothing.

In the *keywords* module, we first extract the relevant introduction from Internet (as Wikipedia, Twitter, etc.). After filtering stop words, the remaining keywords are assigned a weight according to the keywords' frequencies.

In the *picture & keywords* module, each word in the keywords list corresponds to its counterpart in the patch list. Higher weighted keywords should fit into more salient, lager patches because longer keywords in central area can better capture the viewer's attention.

In the *post-processing* module, several kinds of strategies can collaboratively help generate more colorful and meaningful PicWords.

IV. Technique Details of the PicWords System

In this section, we will discuss the methodology of picture module, keywords module, picture & keywords module and post-processing module sequentially.

A. Picture Module: Patch Generation and Ranking

1) Silhouette Image Generation: The first step in the picture module is to split the original image into a Yang part and a Yin part. The Yang part usually corresponds to object contour or important image content, and thus should be kept for keywords placing. To the contrary, the Yin part usually corresponds to background or unimportant image details, and thus can be ignored.

The schematic diagram is shown in Fig. 3. Fig. 3(a) is the source image. We first use mean shift algorithm [15], [16] to segment the image into small superpixels (patches). Each superpixel is considered as a semantic consistent unit. We assume that the biggest meaningless patch near the picture boundary corresponds to background and is thrown away, shown in Fig. 3(b). The foreground is also segmented out simultaneously, and the results are shown in Fig. 3(c). We can see that the spatially nearby and visually similar pixels, such as the cheek area, are clustered together in the segmentation map. Next, we convert all foreground superpixels to greyscale image, which is further thresholded into binary image. More concretely, the average luminance value of each patch is calculated, and only bright patches are kept. The threshold setting can be fully automatic or adjusted according to users' requirement. Fig. 3(d) is the generated binary map. Finally, the binary map is refined and smoothed by applying a Gaussian filtering to remove tiny holes and blurs. The final silhouette is shown in Fig. 3(e). It resembles the original image, but in a more abstract style.

2) Patch Generation: Based on superpixel segmentation method [14], each color image is segmented into several small units as shown in the top panel of Fig. 4. Comparing with mean-shift segmentation method, the adopted superpixel method generally generates the near-rectangular patches, which are more suitable for inserting keywords. Since we target at filling the foreground area with keywords, only the patches covered by the silhouette image are kept.

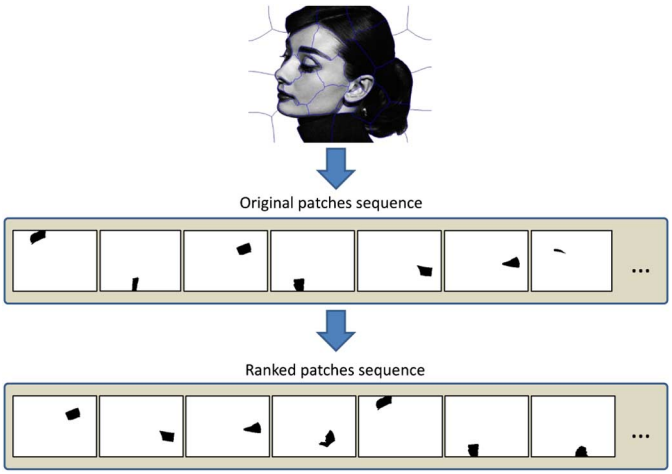

Fig. 4. Top panel: the image is segmented via method in [14]. Middle panel: the obtained patches. Bottom panel: the re-ranked patch list. Larger and center patches are ranked higher in the list.

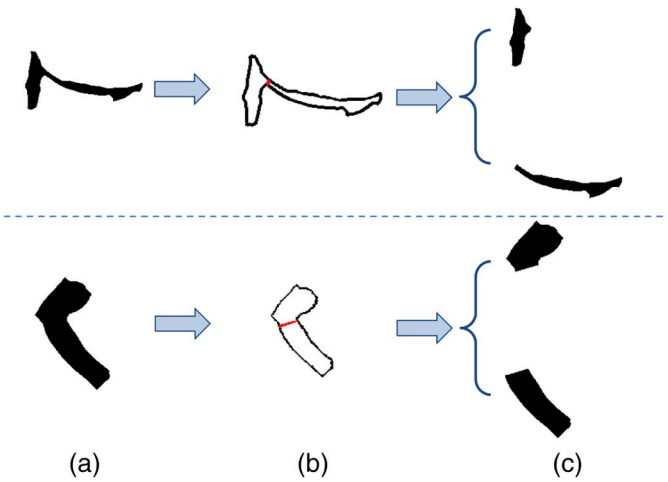

Fig. 5. Two examples of concave patch decomposition, and the cut is marked in red. The original concave patch is cut into two near-convex ones. (a) A concave patch. (b) The cut point (red) on the shape. (c) Decomposition.

3) Convex Decomposition: Most warping methods [17] can work well only when the source and target polygons are near-convex, otherwise produce unacceptable distortion. Therefore, we estimate the concave index of each super pixel by the method introduced of Ren and Yuan [18]. If the concave index of one patch is bigger than a threshold, it is fed into the convex decomposition procedure. As shown in Fig. 5, the original concave patch is split into two near-convex patches [18], which can greatly facilitate the later keywords warping process.

4) Patch Ranking: The results of super pixel based image segmentation are N patches denoted as $\{P_1, P_2, \cdots, P_N\}$. However, these patches are unequally important. For example, larger patches near the center are intuitively more important. We evaluate three criteria for each patch P_i: first, patch area S_i, counted by pixel numbers within the region; second, patch location D_i, indicating the distance between the patch to the center point of image; and third, patch length L_i, calculated by the major axis of the region's external ellipse. Based on the defined three criteria, the weight of a patch is presented as:

$$w_i = w_1 S_i + w_2 L_i - w_3 D_i, \tag{1}$$

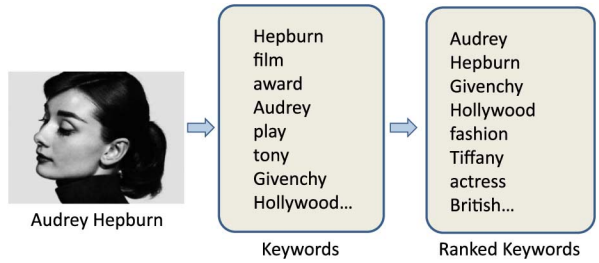

Fig. 6. Keyword ranking results. From the results, we can see that longer and more important keywords, such as "Audrey" and "Hepburn" are ranked higher than some trivial keywords "tony" or "play".

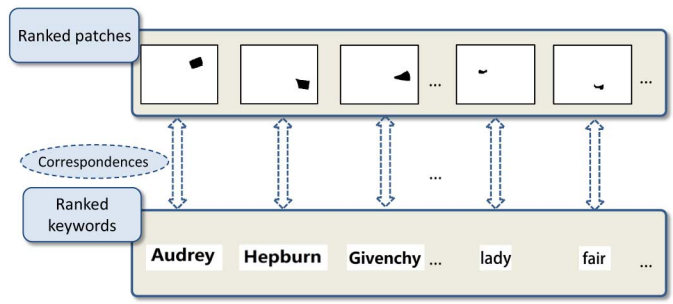

Fig. 7. The correspondence between patch and keyword. Longer and more important keywords are placed in more salient and larger patches.

where w_i is the weight coefficient of each element. In practice, we use $w_1 = 0.3$, $w_2 = 0.4$ and $w_3 = 0.3$. A representative patch ranking result is shown in the bottom panel of Fig. 4. We can see that central and larger patches are usually ranked higher.

B. Keywords Module: Keywords Collection and Ranking

There are mainly two parts in the keywords module. One is keywords collection and the other is keywords ranking.

1) Keywords Collection: In this paper, the keywords are crawled from Internet (such as Wikipedia, Twitter, Weibo, etc.) and processed by the algorithm of [19]. Several text preprocessing techniques such as lowercasing and removing peculiars are conducted. Stop words are also removed.

2) Keywords Ranking: The collected keywords are initially ranked by their frequencies. The more frequently a word appears, the higher it is ranked. We then re-rank these keywords according to their lengths. Comparing with a shorter word "film", a longer word "Hollywood" needs a larger patch to fit in. We present some keywords ranking results in Fig. 6.

C. Picture & Keywords Module: Correspondence and Warping

1) Patch vs. Keywords Correspondence: The goal of this step is to select appropriate container patch for each keyword. The obtained ranked image patches and ranked keywords are mapped sequentially from high weight to low weight shown in Fig. 7 such that: 1) the important keywords have a more salient location such as near the center; and 2) the longer keyword should be matched with a larger patch for better packing.

2) Warping Text to Image Patches: Here, the task is to warp a particular keyword into an arbitrary image patch. It is quite challenging because the patch is extremely unconstrained. It can

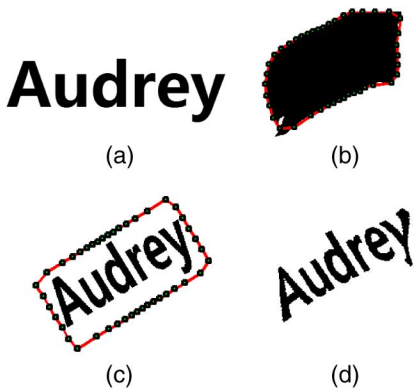

Fig. 8. Given a keyword (a) and its related patch (b), we first adjust the size and orientation of (a) into (c). The sample points of boundary are shown as green points, and (d) is the warping result.

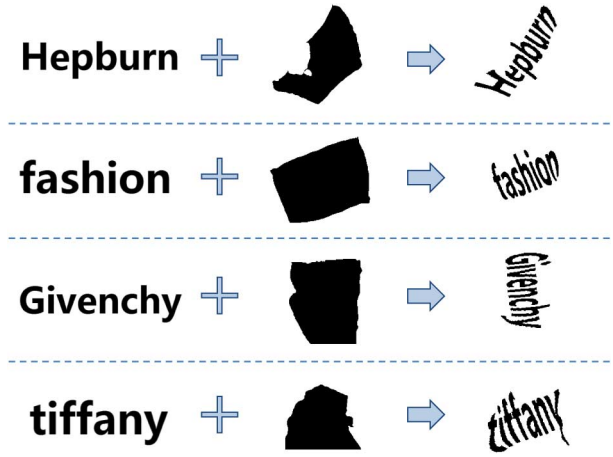

Fig. 9. Four examples of warping keywords according to the reference patch's shape. From the results, we can see that keywords become more artistic after the transformation. And we can still recognize these keywords easily.

be of any size, and of any shape. We illustrate the whole process in Fig. 8. The whole procedure contains three steps.

Size and Principal Direction Alignment: Regular words are usually in upright direction and untilted, while the image patch can be in any direction. Therefore, the first step is to align principal orientation of the word in Fig. 8(a) with the target patch in Fig. 8(b). The aligned word is of the same direction with the patch and shown in Fig. 8(c). Also the size of the word is adjusted according to the size of container.

Correspondence Points Localization: The key idea of most warping methods [17], [20] is that the user must provide an explicit correspondence between the source and target shapes. Other points are mapped according to their relative positions with respect to the correspondence (anchor) points. We would like the computer to derive this correspondence automatically. In our case, we first extract the contour of the word shown as the red boundary in Fig. 8(c). The contour $\mathcal{C}$ is sampled and n points $\{c_1, \cdots, c_n\}$ are generated, emphasized as green points in Fig. 8(c). After that, The outline of the irregularly-shaped image patch is approximated by piecewise-linear paths with vertices and drawn as red line in Fig. 8(b). Then we equally sample the word contour $\mathcal{P}$ and get m points $\{p_1, \cdots, p_m\}$. Usually, m is equal to n. These anchor points are denoted as the green points on the boundary line.

Warping: Based on the correspondences constructed, the word is warped by computing new positions of patch boundary for each of the vertices in those patches.

Hormann and Floater [17] presented the *mean value coordinates* for warping between two arbitrary planar polygons, based on a generalized notion of barycentric coordinates. Barycentric coordinates for triangles provide a convenient way to linearly interpolate data that is given at the corners of a triangle. Given a planar triangle $[v_1, v_2, v_3]$, any point v inside it has three masses w_1, w_2 and w_3. If placed at the corresponding vertices of the triangle, their barycentre will coincide with v:

$$\frac{w_1 v_1 + w_2 v_2 + w_3 v_3}{w_1 + w_2 + w_3} = v \tag{2}$$

So w_1, w_2 and w_3 are defined as the barycentric coordinates of v.

Given an arbitrary planar polygon ψ with vertices $\{v_i\}$, each point $x_i \in \psi$ has mean value coordinates corresponding to three vertices v_{i-1}, v_i and v_{i+1}. For a topologically equivalent target polygon ψ' with vertices v'_i, which ψ and ψ' have the same number of components and vertices for per component, we would like to construct a smooth warp function $f : \psi \to \psi'$ that maps each v_i to v'_i. This warp function f can then be used to deform the source image $I : x_i \in \psi$ into the target image $I' : x'_i \in \psi'$ with new coordinates according to v'_{i-1}, v'_i and v'_{i+1}.

We define a warp from C_i to P_i by the approach introduced above, and apply it to the keywords. We apply the geometric warp of Hormann and Floater [17] inside each convex piece to obtain a warp of the entire phrase polygon. Each point in C_i has the mean value coordinates according to vertices locations. When C_i is warped to the location of P_i, a correspondence between the sample points on C_i and P_i can be generated. Given a correspondence between the sample points on C_i and P_i, we can then map the points in polygon C_i through the correspondence.

Some examples are shown in Fig. 9. Further more, during the PicWords generation process, we do not distinguish key word and phrase. Both of them are dealt as an image, which is then warped to a patch by the mean value coordinates method.

D. Post-Processing Module

Given the example of stitched warped keywords, there is still blank area between keywords. Then we fill these blank space with random selected symbols (such as asterisk) for a better ornamental effect. We choose symbols instead of letters for preventing confusion with keywords.

Current PicWords can only be a binary image. For better visual effects, we also implement two variations of PicWords, i.e., the *Gray PicWords* and the *Colored PicWords*. They use the binary PicWords as a mask to pick the gray/color value of the original image, respectively.

Fig. 10 illustrates the results of different post-processing techniques. Visually, the colored one is better, followed by gray. The binary is the worst. The main drawback of the binary version is that it is difficult for human to segment different keywords.

Fig. 10. The first column is the original image; the second to the fourth column correspond to binary, gray and colored PicWords.

Since we use image segmentation techniques, all the patches are spatially tightly connected and the fitted keywords are also tightly connected. Take the first row of Fig. 10 as example, in the binary case, it is impossible to tell whether it contains {"bestAnne"} or {"best", "Anne"}. With the help of gray value or color value, human can easily tell that "best" and "Anne" are two separate keywords.

V. Experiments

First, we evaluate the effectiveness of the PicWords quantitatively via user study. In practice, we collect pictures including movie stars, brand logos, cartoon figures and find related keywords from Wikipedia. Then we present 25 Picwords images to subjects. Totally, 40 participants (15 females and 25 males who are students and staff members of National University of Singapore) ranged from 22 to 40 years old ($\mu = 27.3$, $\sigma = 3.9$) participated in the user study. Then we show some qualitative exemplar results. Finally we discuss the limitation of the system.

A. Quantitative Results: User Studies

In the user study, we first evaluate the properties of PicWords itself. Then we compare PicWords with other two competitive baselines to show its advantages.

1) Evaluation of Each Component of PicWords: We conduct user study by voting for different versions of PicWords. We evaluate with and without the keywords weighting strategy. All subjects are also asked to vote for different post-processing techniques, i.e., binary, gray and colored PicWords. Considering different weighting and post-processing methods, totally 6 kinds of PicWords versions are compared. Users are asked to give a score from 1 to 10 where 10 is the highest. We calculate the average score of each version and show the results in Fig. 11.

Weight vs. Non-weightPicWords: From Fig. 11, we can see that averagely, the *weight* version is generally better than *non-weight* version, no matter what kinds of post-processing techniques are adopted. If we do not assign weights to the keywords, we cannot make sure that longer keywords shall be fit into longer patches, which will reduce aesthetics of the PicWords.

Colored vs. Gray vs. BinaryPicWords: In both weight and non-weight version, an obvious conclusion can be drawn: the colored version is consistently better than the gray one, while gray is always better than binary PicWords.

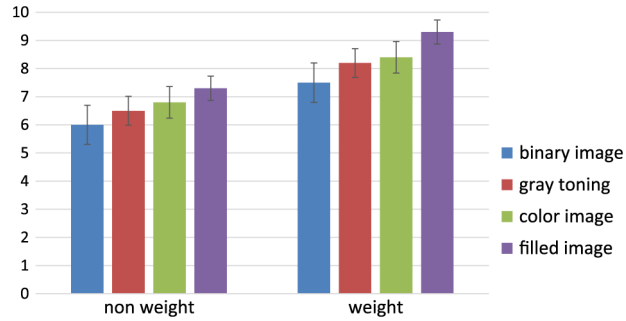

Fig. 11. Evaluation of different versions of PicWords.

Filled vs. Non-filledPicWords: In order to compare the results of the two versions, with and without symbols filling, we fix the other post-processing setting as colored and weight PicWords. Results show that filling is better.

2) Comparison with Baselines: We compare the results of PicWords with two previous works: textorize[3] and word cloud generator.[4] In all settings, the keywords are collected from Wikipedia. We evaluate the three methods in the following six aspects:

- Natural: Is the result image natural or with obvious artifacts?
- Aesthetic: Does the result looks elegant or ugly?
- Discriminative: Can you recognize the people/object inside the image?
- Informative: How much information does the image convey?
- Visual Effect: Is the result vivid?
- Overall Attractiveness: All in all, do you like this system?

We provide the subjects three kinds of results and ask them to choose which result generates the best performance. We count the number of each aspect of each result and show the final results in Fig. 13. PicWords is better than the baselines in all aspects. The two baselines beat each other in different aspects. The textorize method is more aesthetic since it can fit the shape of image. But word cloud is more informative since it embeds many keywords.

B. Qualitative Results

In this section, we first illustrate exemplar results of PicWords compared with other two baselines. Then, we give more PicWords results to show its effectiveness.

1) Comparison with Baselines: The exemplar results of both baselines and our method are shown in Fig. 12. The source image is a portrait of Audrey Hepburn. We can easily conclude that PicWords is more interesting and attractive than word cloud since it reappears both shape and texture of the picture. The advantage is quite obvious since a picture is worth a thousand words. PicWords is also much more exquisite than textorize in two aspects. First, although both methods contain the picture of Audrey, it is much easier to tell the identity of the woman from

[3]http://lapin-bleu.net/software/textorizer/textorizer.html

[4]http://www.wordle.net/

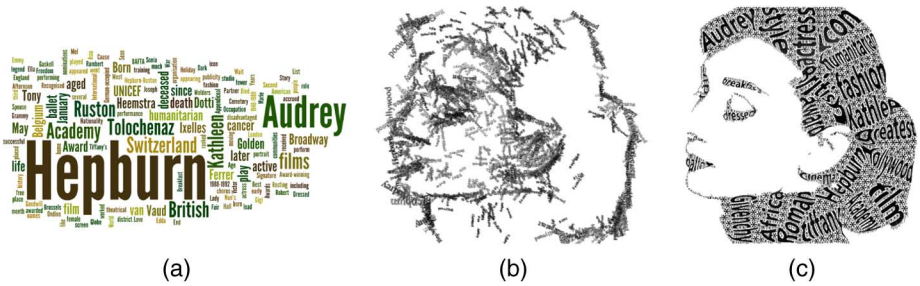

Fig. 12. Comparison between two baselines and PicWords. Left panel: word cloud. From the result, we can easily recognize some keywords, such as "Hepburn", "Audrey", etc. But we cannot imagine what Audrey looks like. Middle panel: Textorize. Text is hardly recognizable and it is difficult to tell that the woman in the picture is Audrey Hepburn. Right panel: The results of PicWords. It has the advantages and correct the shortcomings of both word cloud and textorize. We can easily tell the identity of the person, and we can read the text as "British", "actress", "Tiffany", etc.

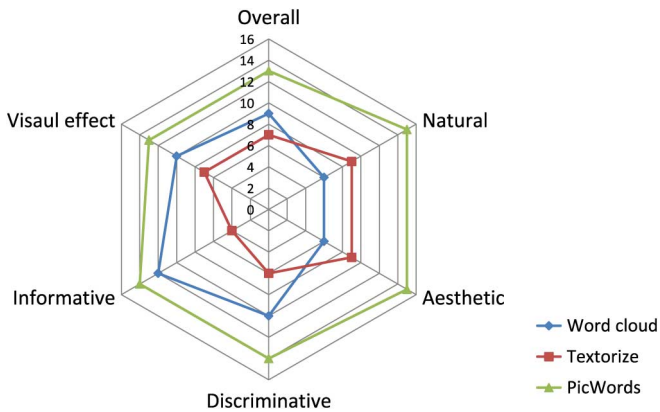

Fig. 13. The radar chart to compare PicWords with word cloud and textorize.

PicWords than textorize. It is because textorize puts too many words inside the facial area, while PicWords can cleverly select the most suitable patches to accommodate the keywords. Second, it is quite difficult to recognize the inserted words of textorize. But it is much easier to read that PicWords contains some keywords such as "British", "film", etc. To sum up, PicWords is much more vivid than words cloud. PicWords is better than textorize in terms of both picture and keywords modalities.

2) More Exemplar Results: In this subsection, we show more results of PicWords on several images with well known movie stars, logos and cartoon figures in Fig. 14. The first column is the source image, the second is the silhouette image, and the third to the fifth columns correspond to binary, gray and colored version of PicWords. The PicWords with symbols filling are put from the sixth column to the eighth column. From the results, we have the following observations. (1) Based on the generated PicWords, we can easily recognize the identity of the person/object of interest. (2) Some details are quite exquisite in the PicWords. For example, the source image of the second row is a photo of Anne Hathaway. We can see that PicWords can well capture the graceful streamline of her long hair. It is even more beautiful than the source image from this perspective.

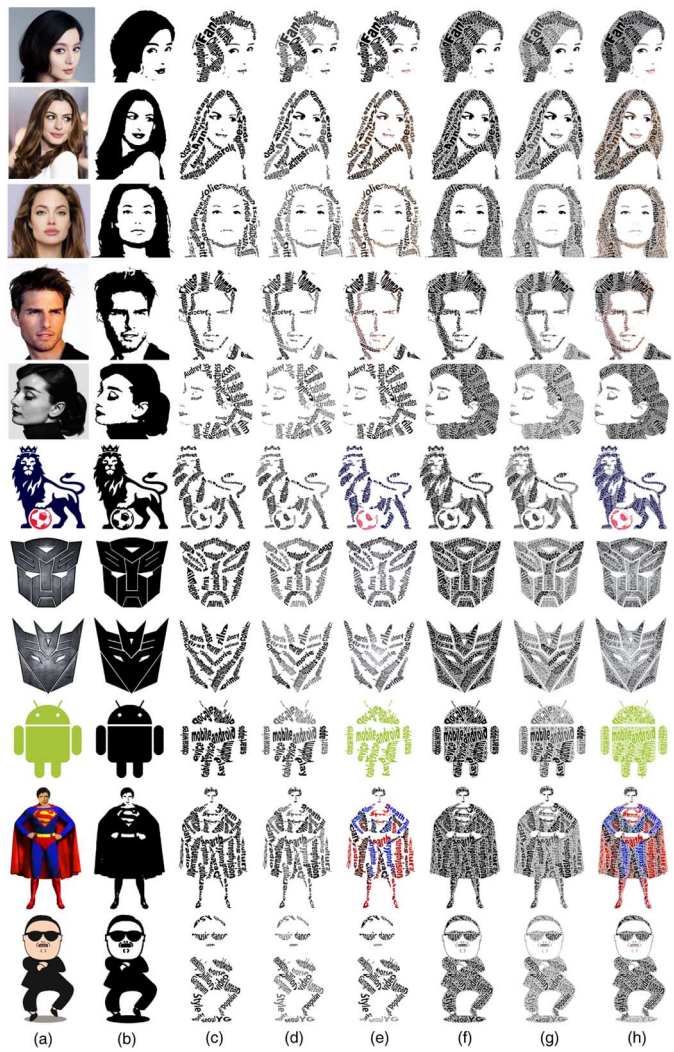

Fig. 14. Examples of PicWords results. (a) are input source pictures and (b) are binary silhouette pictures. From (c) to (e), we present three versions of PicWords: binary, gray and colored. From (f) to (h) are PicWords with symbols filling. For better viewing, please see in $\times 3$ size of original color PDF file.

C. Computational Efficiency & Failure Cases

To demonstrate the efficiency of our framework, it is implemented in MATLAB and tested on an Inter Core 4 computer with 3.40 GHz CPU. Segmentation and warping part are the most time-consuming parts of the whole system. Segmenting a 600*400 picture needs 63 seconds and warping 30 keywords into corresponding patches requires 20 seconds. Other modules of the system take only 1.3 seconds, and the speed would be further improved if the algorithm is implemented in C. In real applications, all the time consuming computing is conducted

in the server end. The response time is expected to be shortened to 10 seconds if implemented on GPU or by parallel computing so that this system can be used to design posters and advertisements.

Of course, the current system is still not perfect and has certain limitations. The success of PicWords comes from the fact that it can keep the region's contour and replace the region's original texture by placing some keywords. Keeping the contour can guarantee PicWords to resemble the original image while region keywords filling can serve as extra information complementary to the image content. But if the original image is textureless, such as Eiffel Tower or Golden Gate Bridge, PicWords will probably fail.

VI. Conclusion and Future Work

We develop an automatic calligram system called PicWords. It can fuse one source picture and keywords seamlessly into one target PicWords. Viewer can sense the picture and read more details from the keywords at the same time. More important keywords have higher weights and are put into more salient and larger regions. PicWords has great market potentials. It can be developed as an app for the social network to generate more vivid and informative user profile photos. In future work, we will apply the technique in broader application areas. For example, it can be used as a new kind of postcard for canteen advertisement. The canteen's picture is used as source image, and its specialty names can be used as keywords to generate a very fancy PicWords.

References

[1] T. Strothotte and S. Schlechtweg, *Non-Photorealistic Computer Graphics: Modeling, Rendering, and Animation*. San Francisco, CA, USA: Morgan Kaufmann, 2002.

[2] J. Xu and C. S. Kaplan, "Calligraphic packing," in *Proc. Graphics Interface*, 2007, pp. 43–50.

[3] J. Kyprianidis, J. Collomosse, T. Wang, and T. Isenberg, "State of the 'art': A taxonomy of artistic stylization techniques for images and video," *IEEE Trans. Visual. Comput. Graph.*, vol. 19, no. 5, pp. 866–885, 2013.

[4] J.-Y. Yeh, M.-C. Hu, W.-H. Cheng, and J.-L. Wu, "Interactive digital scrapbook generation for travel photos based on design principles of typography," in *Proc. 19th ACM Int. Conf. Multimedia*, 2011, pp. 1565–1568.

[5] W.-H. Cheng, C.-W. Wang, and J.-L. Wu, "Video adaptation for small display based on content recomposition," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 17, no. 1, pp. 43–58, 2007.

[6] M. Kuribayashi, K. Fujita, and M. Morii, "Expansion of image displayable area in design QR code and its applications," *Forum Inf. Technol.*, vol. 4, pp. 517–520, 2011.

[7] R. Carroll, A. Agarwala, and M. Agrawala, "Image warps for artistic perspective manipulation," *ACM Trans. Graph.*, vol. 29, no. 4, pp. 127–135, 2010.

[8] A. Hausner, "Simulating decorative mosaics," in *Proc. 28th Annu. Conf. Computer Graphics and Interactive Techniques*, 2001, pp. 573–580.

[9] J. Kim and F. Pellacini, "Jigsaw image mosaics," *ACM Trans. Graph.*, vol. 21, no. 3, pp. 657–664, 2002.

[10] J. Orchard and C. S. Kaplan, "Cut-out image mosaics," in *Proc. 6th Int. Symp. Non-Photorealistic Animation and rendering*, 2008, pp. 79–87.

[11] X. Xu, L. Zhang, and T. T. Wong, "Structure-based ascii art," *ACM Trans. Graph.*, vol. 29, no. 4, pp. 52:1–52:9, 2010.

[12] M. Nacenta, U. Hinrichs, and S. Carpendale, "Fatfonts: Combining the symbolic, and visual aspects of numbers," in *Proc. Int. Working Conf. Advanced Visual Interfaces*, 2012, pp. 407–414.

[13] R. Maharik, M. Bessmeltsev, A. Sheffer, A. Shamir, and N. Carr, "Digital micrography," in *ACM Trans. Graph. (Proc. SIGGRAPH 2011)*, 2011, pp. 100:1–100:12.

[14] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik, "Contour detection and hierarchical image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 5, pp. 898–916, 2011.

[15] D. Comaniciu and P. Meer, "Mean shift: A robust approach toward feature space analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 5, pp. 603–619, 2002.

[16] C. M. Christoudias, B. Georgescu, and P. Meer, "Synergism in low level vision," in *Proc. 16th Int. Conf. Pattern Recognition*, 2002, vol. 4, pp. 150–155.

[17] K. Hormann and M. S. Floater, "Mean value coordinates for arbitrary planar polygons," *ACM Trans. Graph.*, vol. 25, no. 4, pp. 1424–1441, 2006.

[18] Z. Ren, J. Yuan, C. Li, and W. Liu, "Minimum near-convex decomposition for robust shape representation," in *Proc. 2011 IEEE Int. Conf. Computer Vision (ICCV)*, 2011, pp. 303–310.

[19] M. Molch, 2012 [Online]. Available: http://www.find-keyword.com/

[20] J. Gomes, *Warping and Morphing of Graphical Objects*. San Francisco, CA, USA: Morgan Kaufmann, 1999, vol. 1.

Zhenzhen Hu is currently a Ph.D. candidate at HUFT-TI DSP United Research Lab of Hefei University of Technology, Hefei, China. She received her received the B.Sc. and M.Sc. Degrees from the School of Computer and Information Science, Hefei University of Technology (HFUT) in 2008 and 2011. Her research interests include computer vision and multimedia.

Si Liu is currently a research fellow at Learning and Vision Group of National University of Singapore. She received her Ph.D. degree from National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences in 2012. Her research interests include computer vision and multimedia.

Jianguo Jiang is a professor of School of Computer and Information Science, Hefei University of Technology (HFUT). He is head of the TI-HFUT DSP Laboratory in Engineering Research Center of Safety Critical Industrial Measurement and Control Technology, Ministry of Education. His research interests include image processing and multi-agent system.

Richang Hong received the Ph.D. degree from the University of Science and Technology of China, Hefei, China, in 2008. He worked as a Research Fellow in the School of Computing, National University of Singapore, as a Research Fellow from September 2008 to December 2010. He is now a Professor in Hefei University of Technology, Hefei, China. He has coauthored more than 60 publications in the areas of his research interests, which include multimedia question answering, video content analysis, and pattern recognition. Dr. Hong is a member of the Association for Computing Machinery. He was the recipient of the Best Paper Award in the ACM Multimedia 2010.

Meng Wang is a professor in the Hefei University of Technology, China. He received the B.E. degree and Ph.D. degree in the Special Class for the Gifted Young and the Department of Electronic Engineering and Information Science from the University of Science and Technology of China (USTC), Hefei, China, respectively. He previously worked as an associate researcher at Microsoft Research Asia, and then a core member in a startup in Silicon Valley. After that, he worked in the National University of Singapore as a senior research fellow. His current research interests include multimedia content analysis, search, mining, recommendation, and large-scale computing. He has authored more than 150 book chapters, journal and conference papers in these areas. He received the best paper awards successively from the 17th and 18th ACM International Conference on Multimedia, the best paper award from the 16th International Multimedia Modeling Conference, the best paper award from the 4th International Conference on Internet Multimedia Computing and Service, and the best demo award from the 20th ACM International Conference on Multimedia.

Shuicheng Yan is currently an Associate Professor in the Department of Electrical and Computer Engineering at National University of Singapore, and the founding lead of the Learning and Vision Research Group. Prof. Yan's research areas include computer vision, multimedia and machine learning, and he has authored/co-authored over 300 technical papers over a wide range of research topics, with Google Scholar citation 8,100 times and H-index-40. He is an associate editor of IEEE Transactions on Circuits and Systems for Video Technology (IEEE TCSVT) and ACM Transactions on Intelligent Systems and Technology (ACM TIST), and has been serving as the guest editor of the special issues for TMM and CVIU. He received the Best Paper Awards from ACM MM'12 (demo), PCM'11, ACM MM'10, ICME'10 and ICIMCS'09, the winner prizes of the classification task in PASCAL VOC 2010-2012, the winner prize of the segmentation task in PASCAL VOC 2012, the honorable mention prize of the detection task in PASCAL VOC'10, 2010 TCSVT Best Associate Editor (BAE) Award, 2010 Young Faculty Research Award, 2011 Singapore Young Scientist Award, 2012 NUS Young Researcher Award, and the co-author of the best student paper awards of PREMIA'09, PREMIA'11 and PREMIA'12.