VisFM: Visual Analysis of Image Feature Matchings

Chenhui Li[1,2] and George Baciu[2]

[1]School of Computer Science and Software Engineering, East China Normal University, China
dawn.chli@gmail.com
[2]Department of Computing, The Hong Kong Polytechnic University, Hong Kong, P.R. China
csgeorge@comp.polyu.edu.hk

Abstract

Feature matching is the most basic and pervasive problem in computer vision and it has become a primary component in big data analytics. Many tools have been developed for extracting and matching features in video streams and image frames. However, one of the most basic tools, that is, a tool for simply visualizing matched features for the comparison and evaluation of computer vision algorithms is not generally available, especially when dealing with a large number of matching lines. We introduce VisFM, an integrated visual analysis system for comprehending and exploring image feature matchings. VisFM presents a matching view with an intuitive line bundling to provide useful insights regarding the quality of matched features. VisFM is capable of showing a summarization of the features and matchings through group view to assist domain experts in observing the feature matching patterns from multiple perspectives. VisFM incorporates a series of interactions for exploring the feature data. We demonstrate the visual efficacy of VisFM by applying it to three scenarios. An informal expert feedback, conducted by our collaborator in computer vision, demonstrates how VisFM can be used for comparing and analysing feature matchings when the goal is to improve an image retrieval algorithm.

Keywords: information visualization, image processing, visual analytics

ACM CCS: Human-centred computing → Visualization, Visual analytics

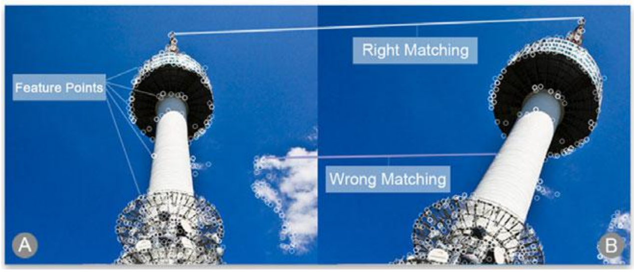
1. Introduction

Image feature matching visualization has attracted considerable interest in the field of computer vision due to increasingly stringent requirements for applications such as image retrieval and pattern recognition. A feature matching is a set of links between pairs of features. Each pair of matched features between two similar images is visually linked by a link (edge). The image features are extracted using standard computer vision methods, such as SIFT [Low04] and SURF [BTVG06]. Feature extraction and matching thus comprise a fundamental problem in the fields of computer vision and image processing. This problem lies at the root of numerous high-level research problems, such as image-based localization [QZH15], duplicate image discovery [WZL13] and object tracking [ZYS09]. A pair of images may contain many features and matchings. Therefore, visualizing these matchings over the source images in a straightforward manner may lead to visual clutter.

The feature matching accuracy can be easily computed using existing matching algorithms; however, the correct and incorrect matchings over two images are still difficult to discern. Figure 1(a) shows an example of a feature matching visualization that suffers from visual clutter. Obviously, most of the correct matchings are difficult to find by simply showing all of the matching lines. A correct matching indicates that for a feature in one image, the corresponding feature can be found in another image, even if that image has been rotated, translated or scaled, as shown in Figure 1(b). Hence, with an appropriate tool designed for the visual analysis of image features and matchings, a better understanding and potential improvement of related computer vision algorithms could be achieved more intuitively and efficiently.

Information visualization researchers have proposed many approaches to visualizing edge clouds, e.g. edge bundling [Hol06]. In the edge bundling method, adjacent edges are visualized as a bundled group to reduce visual clutter. Hence, visualization methods based on edge bundling can be used to represent matching lines between image features. However, few methods have been presented with the specific purpose of visualizing image features and their matchings. This often requires the consideration of feature characteristics such as image colour, image segmentation and matching position. To the best of our knowledge, a comprehensive solution

(a) Feature matchings on a pair of images (A and B).



(b) Examples of correct and incorrect feature matchings.

Figure 1: *Visualizing feature matchings in a straightforward way can lead to visual confusion. White bubbles outline the image features.*

for visualizing image features and their corresponding pair matchings has not been investigated from an information visualization perspective.

In order to address these concerns, we present a visual analysis system, called VisFM. VisFM clearly presents image features and their matchings in order to support improvements in related computer vision approaches, such as image retrieval applications. Based on requirements specified by experts, we summarize the necessary tasks of VisFM. Image feature data are extracted and clustered before further visual analysis can be performed. VisFM presents three main views for the observation of image feature matchings: a matching view, a feature view and a group view. We consider the image background colour in the feature and matching views to facilitate feature understanding and match finding. We also provide user-friendly semi-automatic interactions for highlighting and querying information from the desired region. A case study and feedback from a domain expert demonstrate the effectiveness of VisFM. VisFM can either provide support for computer vision researchers or extend the fields of application of edge bundling techniques.

In summary, this paper proposes a visual analysis system with the following contributions:

(1) an interactive visual analysis system for understanding and exploring feature matching patterns;
(2) a novel layout design for visualizing links of image feature matchings;
(3) an institute group view design for representing the relationships of image segments and
(4) the extension of the scope of edge bundling applications.

2. Related Work

2.1. Image feature analysis

SIFT [Low04] is amongst the most common feature descriptors for characterizing image features. Normally, SIFT features are represented by 128-dimensional feature vectors. The number of matching SIFT feature pairs can be used to assess the similarity between two images. A large number of matching feature pairs between two images indicates a high similarity. Although SIFT is slower than SURF [BTVG06], another image feature descriptor, it can achieve higher accuracy. Thus, in many feature-based applications, such as image-based localization [QZH15], SIFT is adopted as the preferred image feature descriptor.

The similarity between two images can be calculated using the feature matching approach. The simplest and most time-consuming feature matching method is the brute-force method, which involves computing the each similarity between pairs of feature vectors one at a time. FLANN [ML09] uses kd-tree to accelerate the feature matching process. The brute-force method offers higher accuracy than FLANN, but FLANN is faster. Many methods can be used to cluster image key points, such as k-means [KMN*02], spectral clustering [vL07] and salient region matching [QZH15], to improve the feature matching accuracy and speed up the matching process. Spectral clustering is an extension of the k-means algorithm based on a combination of dimensional reduction with the k-means approach. Previous work has mainly focused on how to improve the accuracy and reduce the time cost of feature matching methods; however, no convincing visual analysis tool exists for the interpretation of feature patterns and matching patterns.

2.2. Feature visualization

Image features are typically visualized as points on an image; thus, a scatterplot is the most common visualization method. Many scatterplot-based visualizations have been proposed, as summarized by Ellis and Dix [ED07]. Attempts to render all the points on the screen suffer from the overlapping problem because the number of pixels is relatively much lower than the image data. To solve the point overlap problem, a kernel density estimation (KDE) method [EKS*96] has been used to find point patterns in a visually friendly way. Splatterplots [MG13] were proposed to further overcome the overlap problem by means of KDE and multiple contours when the data set includes multiple groups.

A number of visualization techniques have been developed from a computer vision perspective to improve the human understanding of image features. Vondrick *et al.* [VKP*16] proposed an accurate visualization tool for understanding an object detection system to improve feature space selection. Zeiler and Fergus [ZF14] presented a visualization tool for visualizing and comprehending convolutional image features, which are widely used in convolutional neural network (CNN) based applications [KSH12]. Nevertheless, the visualization of features on an image remains challenging because combinations of points on nearby pixels can lead to colour conflicts, which will affect the ability to observe and compare features.

2.3. Edge bundling

Edge bundling is a visual clustering method in which edges are organized into groups to achieve an uncluttered visualization layout. Many methods of edge clustering have been presented from different perspectives, as summarized in the work of Zhou *et al.* [ZXYQ13]. Holten [Hol06] presented an early work on edge bundling. Later, force-directed edge bundling (FDEB) [HVW09], a physical force-based approach, was developed to reduce edge clutter in large graphs. Following the emergence of FDEB, many edge bundling methods arose, most of which were focused on increasing the speed and effectiveness of bundling. For example, Telea and Ersoy [TE10] proposed an image-based approach (IBEB) for rendering a skeleton of bundled edges. Ersoy *et al.* [EHP*11] then extended the work on IBEB and presented a skeleton-based method of iteratively transforming edges towards the skeleton of the line set.

Hurter *et al.* [HET12] presented a fast image-based edge bundling method for spreading control points while clustering edges. They were the first to implement edge bundling on a GPU. Instead of pursuing speed improvements, Hurter *et al.* [HET13] performed edge bundling for time-varying data. Afterward, Bach *et al.* [BRH*17] presented a confluent drawing method for visualizing a graph by considering network connectivity and information preservation. To further reduce the complexity of bundled edges, a module-based edge bundling method was proposed by Dwyer *et al.* [DMM*14]. A similar cluster-based approach was later considered by Sun *et al.* [SMNR16].

Edge bundling has been frequently addressed in information visualization research; however, current edge bundling methods have not been deployed and evaluated for image feature matching analysis, which requires the consideration of image colour and segmentation information.

3. Task Analysis

Feature matching is frequently implemented in image processing and computer vision. We collaborated with two domain researchers to develop the VisFM system. One of them is a researcher in the image processing field (P), and the other is from the computer vision field (V). From several discussions, we identified the basic requirements for the visual analysis of image feature matchings. The main objective is to use VisFM to support the understanding of feature matchings and the improvement of image retrieval algorithms. We designed a prototype with a bundled matching line layout and received very positive feedback from them. They also offered many insightful suggestions based on the prototype system design. Hence, the development of the VisFM system was an iterative process. Based on the feedback from the domain researchers, we confirmed the following requirements for the system.

R1: **Analysing the feature data for a pair of images**. Both experts wanted the system to incorporate methods of basic feature extraction and matching. Moreover, expert P stipulated a requirement for feature grouping since he wanted to be able to observe the matchings segment by segment. Expert V specified a requirement for a feature density calculation. He claimed that an overview of salient features is necessary, particularly when thousands of image features are extracted from a pair of images.

R2: **Visualizing the image features**. This requirement is aimed at the visualization of feature characteristics, such as location, size and angle. In addition, for the visualization of high-density feature points, a hierarchical representation of the feature points is required to allow features to be observed in either a macroscopic or a microscopic view.

R3: **Revealing feature matchings**. Feature matching visualization without visual clutter is a pervasive requirement in the computer vision field. A convincing representation will help experts or other users to identify the matching lines of interest. This is helpful for interpreting the results of existing algorithms and finding shortcomings in these algorithms.

R4: **Exploring feature matchings from different perspectives**. Given a structure of matching lines, the domain researchers expressed the desire to be able to explore this structure from different perspectives, based on aspects such as feature size, feature angle, feature density and image segments. In addition, the ability to access matching information for two similar segments in two different images will be beneficial for understanding feature matching algorithms. Hence, the visual analysis system must provide the relevant interactions.

R5: **Overcoming the negative influence of image colour**. Both experts affirmed that they had suffered from disharmonious colour representations on images in their previous research work. Compared with the colour assignment in the absence of an image background, the visualization of image features and matchings while overcoming the influence of background colour is more difficult; however, it is a critical task for in visual analysis of clustered feature matchings.

R6: **Comparing the matching results from different approaches**. The domain experts expected to be able to use the system to compare two matching results obtained using two different approaches, one being an initial method and the other one being an improved method. Such a comparison of matching results can assist an expert in answering the question of why mismatches occur.

4. System Overview

We developed a visual analysis system, VisFM (Figure 2), to fulfil the requirements specified by the experts. The VisFM system includes the following components: (1) data analysis methods for initializing the image feature data for the visual analysis system (**R1**), (2) a visualization system with three main views to reveal the image features and their matchings (**R2**, **R3**, **R5**) and (3) a set of interactions for exploring the features and matchings (**R4**, **R6**).

Figure 3 shows the pipeline of the VisFM system. We assume that the input to VisFM is a pair of images since matching data are typically generated between two images. The feature view comprises a heat map and a zoom-in/zoom-out interactive view (**R2**). The heat map is generated according to the estimated feature density. The zoom-in/zoom-out view allows the user to observe the image features at different levels of resolution. Based on the feature data analysis (**R1**), a matching view and a group view are also designed to display the feature matchings (**R3**). We transform the matching lines

Figure 2: *VisFM system interface. (a) A matching view shows both bundled matching data and segments of images; (b) a feature view presents the feature data through a heat map and a zoom-in/zoom-out interactive mode; (c) a group view shows the matching relationships of feature groups between two images; (d) a feature scatterplot view and a circular heat view provide additional explorations of the features and (e) an option panel supports parameter setting and interaction mode switching.*

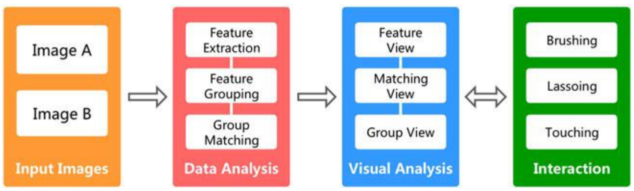


Figure 3: *The pipeline of the VisFM system.*

into bundled edges in the matching view to enable the separation of groups of edges bundled together from other groups. The matching view shows a microscopic view of the matchings, whereas the group view shows the matchings at a macroscopic level. The group view also reveals the correlations between pairs of groups from each image. The ability to explore the feature matchings from different perspectives is considered (**R4**). From the perspectives of feature size and angle, the user is allowed to explore the matchings in a feature scatterplot view and a circular heat view. From the perspective of feature position in an image, interaction with the image itself is supported. From the perspective of feature groups, the group view is designed for exploration of the matchings based on a selected group. Since both the matching view and the feature view contain the images themselves as the background, a contrasting colour assignment (CCA) approach has been proposed to resolve the issue of colour conflict between the visualized items and their background (**R5**). In the VisFM system, the user is allowed to adjust algorithm parameters and to load different image data sets; thus, one can compare results by running two instances of the VisFM system (**R6**). For example, one instance of the system can be used to view the matching results for a pair of original images, while another shows the results for a pair of processed images.

The data analysis component of the system was developed based on Flask, a web framework written in Python. The visualization views and interactions were implemented using the D3 [BOH11] JavaScript library.

5. Feature Data Analysis

In this section, we present the processes executed as part of the feature data analysis. We first introduce the characteristics of features and their extraction from images. We then present the following analytical methods applied to the features and their matchings: density estimation, feature grouping and group matching.

5.1. Feature data

We adopt Lowe's method [Low04] to transform an image into a large number of feature vectors, each of which is invariant to image deformations such as rotation, scaling and translation. Each feature vector consists of a 128-dimensional feature descriptor and three feature characteristics, namely the position on the image, the size and the angle. The similarity between two images is assessed by individually calculating the Euclidean distance between each feature vector pair. High-similarity images contain more matching features. We assume that the input to our system consists of two feature sets, one from each of two images, I_a and I_b. Each feature set can be represented as a set of feature points in a two-dimensional space. We define the feature point sets as $\mathbf{F}_a = \{f_{a0}, f_{a1}, \ldots, f_{an}\}$ and $F_b = \{f_{b0}, f_{b1}, \ldots, f_{bn}\}$, where n is the number of feature matchings. A feature matching that represents a feature point in $\mathbf{F}_a$ is matched with a feature point in $\mathbf{F}_b$. Because we focus on the data analysis of feature matchings, unmatched image features are ignored. Thus, we define the set of matching lines as $\mathbf{M} = \{mat_0, mat_1, \ldots, mat_n\}$, where each mat_i represents a pair consisting of one element in $\mathbf{F}_a$ and one in $\mathbf{F}_b$. We adopt two methods, (1) the brute-force method and (2) FLANN proposed by Muja and Lowe [ML09], to extract matching pairs from the feature vectors. A brute-force matching search is time-consuming; however, the matching accuracy is high. By contrast, FLANN is faster in matching but has a much lower correct matching rate compared with the brute-force method.

5.2. Feature point density

We use the KDE method to calculate the feature density. We adopt the Gaussian function $G(t) = \frac{1}{\sqrt{2\pi}} e^{-\frac{t^2}{2}}$ as a kernel because of its smoothness and parameterization. A detailed discussion of KDE as a fundamental concept in statistics can be found in the work of Silverman [Sil86]. The generated density map D shows the salient feature data in the image, which can be used to define the relative priority of feature regions during visual analysis.

5.3. Feature grouping

The basic intention of feature grouping is to simplify the structure of the feature set by grouping features that belong to the same segment. We adopt a fast superpixel method (SLIC) [ASS*12] to group pixels into visually meaningful segments, thus moving from the pixel level to the region level. We group the features based on the generated segments. Feature grouping is implemented by considering image-region connectivity in order to group features that belong to the

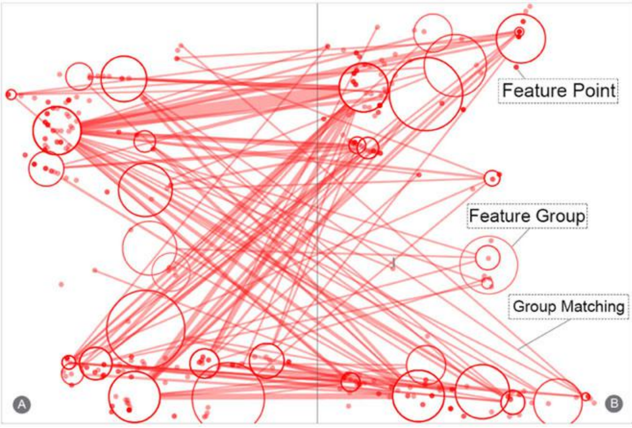


Figure 4: *An example of the feature grouping in an image space. A solid circle indicates a feature point. Each hollow circle indicates a group, where the circle radius represents the size of the group.*

same segment. The feature grouping method is applied to both images in the pair at the same time. Our feature grouping method considers not only spatial proximity but also region connectivity. Considering only spatial proximity, as in k-means [Mac67], leads to perceptually improper groups. We define the feature groups of I_a as $\mathbf{g}_a = [g_{a1}, g_{a2}, \ldots, g_{an}]$ and the feature groups of I_b as $\mathbf{g}_b = [g_{b1}, g_{b2}, \ldots, g_{bn}]$. We discuss the detail of superpixel generation in Section 6.2. Figure 4 shows an example of the feature grouping results in an image space. The position of each group is obtained through the feature point centre calculation.

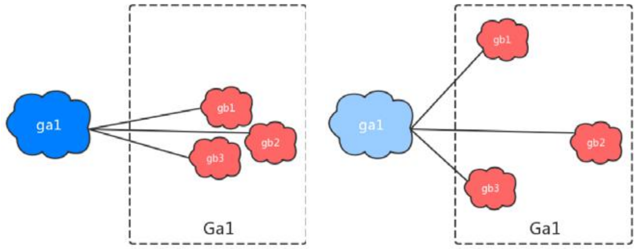
5.4. Group matching

The purpose of the group matching method is to find the matching segments in a pair of images. This method takes the information from both images and the feature matchings themselves into account to support further interaction and exploration of the feature matchings. Through further interaction, we can identify only the groups of features and their matchings that link one segment in image A to a similar segment in image B. The basic idea of group matching is to calculate the number of matching features that belong to two corresponding groups from the two images. We formulate this solution as follows:

$$c_{aibj} = cout\{mat(f_a, f_b) \in \mathbf{M} : f_a \in g_{ai}, f_b \in g_{bj}\}, \qquad (1)$$

where mat means a valid feature matching between two features and $cout$ indicates the set number. There is a link e_{aibj} between g_{ai} and g_{bj} as $c_{aibj} > 0$. The topology of the matchings between groups is denoted by $\mathbf{GG} = (\mathbf{GV}, \mathbf{GE})$, where $\mathbf{GV} = [g_{a1}, g_{a2}, \ldots, g_{an}, g_{b1}, g_{b2}, \ldots, g_{bn}]$ denotes the nodes and $\mathbf{GE} = [e_{a1b1}, e_{a2b1}, \ldots, e_{anbn}]$ indicates the links. Figure 4 shows an example of the group matching result in an image space.

Unlike feature matching, group matching is uncertain because the segment shapes in image A and image B are most often different. Hence, we define the group matching weight (GMW) to indicate the correct matching rate of a group (segment). GMWs can be used to guide further visual design (Section 6.4) for the exploration of feature matchings. If g_{ai} is matching to g_{bj} and g_{ai} is also matching to the neighbours of g_{bj}, we consider g_{ai} has high GMW. A high GMW indicates a high probability of correct feature matching, and a low GMW indicates a potential for incorrect feature matching between groups.



(a) A group in image A with a higher GMW.

(b) A group in image A with a lower GMW.

Figure 5: *A comparison of different GMW. A higher coherence of groups leads to a higher GMW.*

We define the linked groups to g_{ai} as a complete graph $G_{ai} = [V_{ai}, E_{ai}]$, where $V_{ai} = \{g_{bj}, c_{aibj} > 0\}$ indicates the graph nodes and E_{ai} indicates the graph edges. We then transform the problem of GMW calculation to a problem of node coherence calculation as follows:

$$w_{ai} = \frac{1}{N(E_{ai})} \sum_{g_\alpha \in V_{ai}, g_\beta \in V_{ai}} \frac{c_{aib\alpha} + c_{aib\beta}}{2c_{ai}} \frac{d_{\max} - dist(g_\alpha, g_\beta)}{d_{\max}}, \qquad (2)$$

where i and j are the group indices for image A and image B, respectively; c_{ai} is the number of features in group g_{ai}; c_{aibj} is the number of matched features between g_{ai} and g_{bj}; $N(E_{ai})$ indicates the number of edges; $d_{\max}$ indicates the maximum group distance and we set it as the diagonal length of the image; $dist$ is a function for measuring the Euclidean distance between two group centroids in image B. Similarly, we can calculate the GMW of groups in image B and then lead to the final group matching weights $\mathbf{GMW} = [w_{a1}, w_{a2}, \ldots, w_{an}, w_{b1}, w_{b2}, w_{bn}]$. Figure 5 shows a comparison of the different GMW.

6. Visual Design

The objective of the visualization design of VisFM is to assist experts in interpreting feature data to support further analysis. The complete visualization design includes a matching view, a feature view, a group view and two additional feature sub-views for the exploration of matching data. In addition, we apply a CCA to the matching view and the feature view. Moreover, the VisFM system supports multiple user-friendly interactions with the various views.

6.1. Interface

Figure 2 shows the user interface of VisFM. It consists of five interactive components: (1) a matching view that depicts either the bundled matching data or the segments of an image; (2) a feature view that presents a heat map of the feature data and detailed feature characteristics at different levels; (3) a group view that shows the relationships between matching groups in two images; (4) a feature scatterplot view and a circular heat view that assist the exploration of matching data and (5) an option panel that provides functions for parameter setting and interaction mode switching. VisFM also provides a menu panel that provides additional interactions, such as data set loading and bundling mode switching.

6.2. Matching view

The main purpose of the matching view is to separate matching line groups to present an uncluttered layout. There are two sub-views in the matching view. The left sub-view displays image A as a background, whereas the right sub-view displays image B as a background. We first segment the images based on pixel connectivity and then apply a group-based edge bundling (GBEB) approach to the matching lines to achieve an uncluttered layout.

An important basis for the matching view is a superpixel approach for revealing feature matchings segment by segment. As discussed in Section 5, we use SLIC, a fast superpixel-based method, to generate high-quality and visually meaningful segments. The main function of SLIC is to divide the image into uniform superpixels as shown in Figure 6(b) and then to iteratively adjust the pixels in each superpixel. The superpixel size is initialized in advance. During each iteration, each pixel in the image is assigned to its closest superpixel by computing the colour distances between it and its neighbouring superpixel centres. Iteration stops when no further pixel has been assigned to a neighbouring superpixel. Figure 6(c) shows an example of the final superpixels of an image. Generated superpixels support the feature grouping and the complete object selection (Figure 6d). Feature points in the same superpixel (segment) will be grouped together. Complete object selection is helpful for the object matching analysis. A related interaction, brushing, is discussed in Section 6.6.

As mentioned in Section 5, the image features are grouped according to the image segmentation. Our *GBEB* procedure is performed by considering the matched groups in image A and image B. For the features of a group in image A (g_{ai}), we find the matching features in a group of image B (g_{bj}). If the number of matchings with a group in image B is greater than 1, we apply GBEB to separate the corresponding matching lines from the others. We define a set of

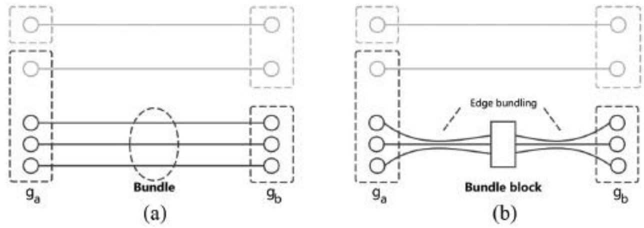


Figure 7: *An example of the group-based edge bundling approach. (a) Matching lines without edge bundling. (b) Matching lines with our edge bundling.*

matching lines between two groups as a bundle. The key idea of GBEB is to group the bundle in a block as shown in Figure 7(b), which is designed to represent the bundle in an organized and controllable form. In our system, the spatial characteristic of the bundle block does not relate to the spatial correspondence of the underlying image. We have interviewed our cooperating experts and they said the influence from the additional spatial correspondence was slight when they were using VisFM. It is easy for them to identify the bundle block as a junction that connects the feature points of two groups.

Bundle block design is inspired by the work of Biset [SMNR16]; however, the design detail and the usage scenario in our system are different. There are three advantages of the bundle block design. First, a rectangle is easier to be selected in a visual exploration than selecting a set of matching lines. Second, the regular arrangement of the matching lines at two sides of the bundle block is convenient for the selection of a single matching line. Third, a bundle block reduces the implementation difficulty of an uncluttered matching line layout.

There are five parts of the GBEB approach. First, we calculate the left feature centre (cen_a) and right feature centre (cen_b) of a bundle. Second, we initialize the bundle block position as $bb = \frac{1}{2}(cen_a + cen_b)$. Third, we apply the force-directed approach [FR91] on all bundle blocks in order to avoid the bundle block overlapping. Force-directed approach separates a bundle block from others as shown in Figure 8(b). Fourth, after updating the bundle block position, linking nodes at two sides of the bundle block are placed one by one as shown in Figure 7(b). Finally, for the two sets of the matching lines separated by a bundle block, we apply the FDEB [HVW09]

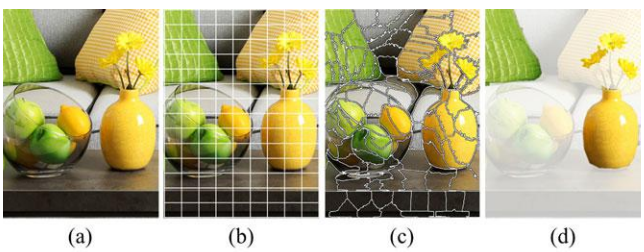


Figure 6: *An example of the superpixels generation and selection. (a) Original image. (b) Initial uniform superpixels. (c) Final superpixels. (d) Complete object selection.*

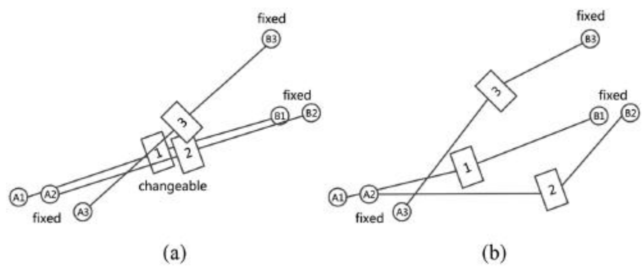
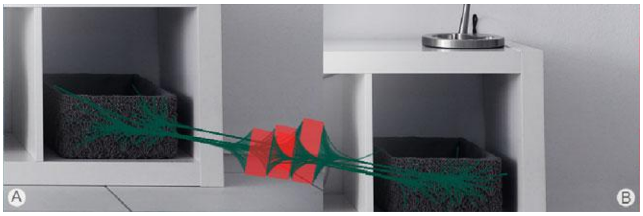
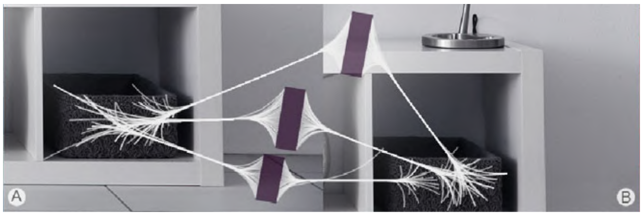


Figure 8: *An example of the bundle block layout adjustment. (a) Bundle blocks without layout adjustment. (b) Bundle blocks with force-directed layout.*

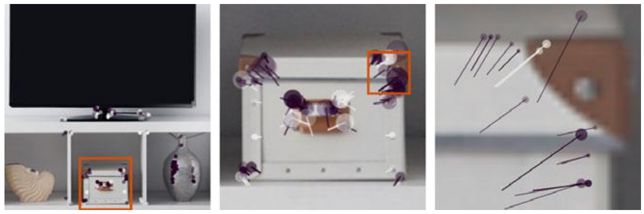


(a) Original bundle blocks layout.



(b) After applying force-directed layout adjustment and CCA.

Figure 9: *An example of the optimized bundle blocks layout and colour assignment.*


(a) Feature representations at different level of detail.

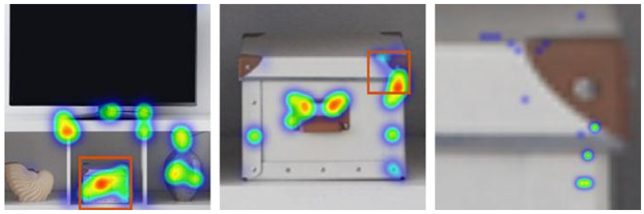

(b) Feature heat maps at different level of detail.

Figure 10: *Examples of the feature view. Feature view is allowed to switch the level of detail.*

technique to implement edge bundling, as shown in Figure 7(b). Figure 9(b) shows an example of the optimized bundle blocks layout.

6.3. Feature view

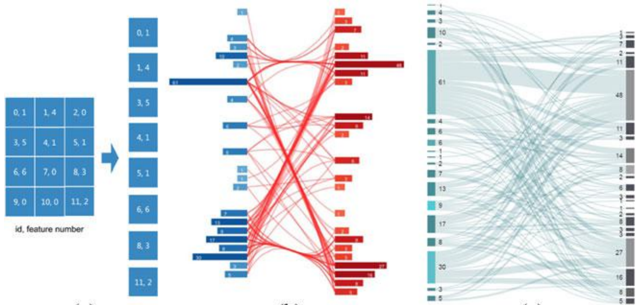
The feature view comprises two view modes: a heat map and a zoom-in/zoom-out mode. The heat map visualization was suggested by expert V, whose area of research is saliency map detection and analysis. He emphasized that 'Saliency indicates the most important and informative parts of a scene; thus, it can improve the efficiency of a visual analysis system of the feature matching'. Therefore, we present a heat map approach for enhancing salient regions. A feature density map is estimated using the method described in Section 5. We use a colour mapping to indicate the different density scales as shown in Figure 10(b). The zoom-in/zoom-out mode, which was inspired by the visualization of geographical data on a map, provides the ability to observe features at different levels to further overcome the problem of feature clutter on the original image. The detailed size and angle of a feature, which is represented by a circle and a line, can be observed in the zoom-in/zoom-out mode. The zoom-in/zoom-out mode also avoids the need to adjust the interface layout for different image sizes and resolutions. Figure 10 is an example of the feature view in different levels of details (left-to-right: from low detail to high detail).

In addition, we designed two additional sub-views (Figure 2d), for visualizing statistical feature characteristics. The feature scatterplot view shows a distribution of the feature angles and sizes in two-dimensional space. It allows the user to effectively select features in a given range of angle or size. The cluster pattern in two-dimensional space can be found through a scatterplot view. The circular heat view represents the heat information of the features in a polar coordinate system. Arc glyphs in the circular heat view indicate the statistical angles of the features. The distance between an arc glyph and the centre of the view indicates the statistical size of the features. A larger distance means a larger feature size. The circular heat view is helpful for image matching analysis in the case of repetitive patterns. The heat information related to the size and angle spaces could effectively represent the similarity between two images. These two sub-views not only represent the feature information in another form but also provide abundant interactions for feature matching selection.

6.4. Group view

The group view is designed for the visualization of matching data at a macroscopic level. The correspondences between feature groups in the two images are visualized in the group view. Finding the matching is difficult when there are a number of feature groups in a pair of images because of the group overlapping problem. Figure 4 is an example of the group overlapping problem in an image space. Therefore, we design an organized form to visualize the group matchings. The group view is separated into three parts. The left part shows the group nodes as summarized from image A, whereas the right part shows the group nodes as summarized from image B. We transform group nodes in an image space to a vertical space in order to avoid the node overlapping in the visualization. Figure 11(a) shows an example of our approach, where a group with no feature is hidden. The length of each node indicates the number of features. The colour depth of the node indicates the GMW. The calculation of the GMW is described in Section 5. If the GMW is high, we assign a deep colour to the node. Otherwise, we assign a light colour to it. A light colour indicates a group pair that includes potentially incorrect feature matchings. The middle part visualizes the relationships between the groups.

In the prototype of group view as shown in Figure 11(b), we have not considered the components of a group, which indicate a number of feature sets that are matching to the groups in another image. The improved group view, using Sankey diagram as shown in Figure 11(c), provides a more intuitive matching result. Sankey diagram was mainly designed for visualizing event changes [SAA*11],



(a) (b) (c)

Figure 11: *Group view design for the group matching visualization. (a) Space transformation. (b) Edge bundling. (c) Sankey diagram.*

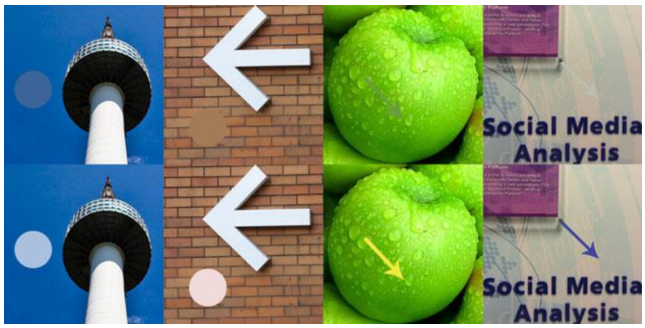


Figure 12: *Contrasting colours can more effectively draw human attention. (top) Points and arrows with non-contrasting colours. (bottom) Points and direction arrow with contrasting colours.*

such as the event convergence and divergence. We adopt Sankey diagram because of its advantage of visualizing many-to-many data. Although edge bundling enables an uncluttered layout for the group matching lines, it is difficult to identify the one-to-one matching detail as shown in Figure 11(b).

6.5. Contrasting colour assignment

A CCA approach is applied in the feature view and the matching view because these views include a hybrid visualization of the image and vectorial element. A visualization of the vectorial elements on an image, such as points and lines, is subject to colour influence for a human observer, thus affecting the user's ability to find patterns in these views. The importance of assigning contrasting colours has been discussed in the works of Lin *et al.* [LRFH13]. Figure 12 illustrates that the visual perception of humans is highly sensitive to the colour contrast against an image background.

We present a CCA model to guarantee that our visualizations will easily draw the user's attention by ensuring their luminance contrast. The CCA model follows the web content accessibility guidelines [CCRV08]. In the CCA model, we formulate the contrast ratio between two colours as follows:

$$Cont_r(c_1, c_2) = \frac{\mathbf{max}(\mathbf{Lumi}(c_1), \mathbf{Lumi}(c_2))}{\mathbf{min}(\mathbf{Lumi}(c_1), \mathbf{Lumi}(c_2))}, \tag{3}$$

where **Lumi** is the illumination calculation for a colour [CCRV08]. For the colour assignment, we prepare 64 colour candidates (can_i, $i \in [1, 64]$) with an average RGB distribution in advance. We formulate the CCA for a feature point as follows:

$$Cont_c(f) = can_i, \, Cont_r(col_b(f), can_i) > \sigma, \tag{4}$$

where col_b is the background colour of a feature point f, $Cont_c(f)$ is the corresponding contrasting colour and σ is a free parameter that defines the termination condition for contrasting colour finding. The contrasting colour finding is performed on 64 colour candidates (can_i). We set σ to 5.0 to avoid excessive time consumption for finding the colour with the maximum contrast ratio. According to the web content guidelines [CCRV08], the maximum contrast colour ratio is 21. We assume that $\sigma = 5.0$ is enough for CCA. Based on the CCA method, we can assign a contrasting colour for each feature circle and angle line in the feature view. For the colour assignment of the bundle block in the matching view, we adopt the same approach. For the colour assignment of the matching lines in a matching view, we assign the contrasting colour by considering all background colours of the matched feature points in a group. All matching lines belonging to the same group matching will share the same colour. We formulate the CCA of a matching line as follows:

$$col_l(g_\alpha, g_\beta) = \underset{can_i, i\in[1,64]}{\arg\max} \sum_{f\in(g_\alpha \cup g_\beta)} Cont_r(col_b(f), can_i), \tag{5}$$

where g_α and g_β indicate a pair of matched groups. Figure 9(b) illustrates the harmony CCA result of the matching lines and the bundle blocks.

6.6. Interactions

We provide three basic interactions, namely brushing, lassoing and touching, in our system to assist users in finding and understanding feature matchings.

Brushing Based on prior works on superpixels and feature grouping, we provide the brushing interaction to highlight only the matching elements corresponding to certain segments and bundle blocks. The user can select segments or bundle blocks by brushing, which will cause all related contents and the corresponding matching lines to be highlighted. Because the superpixel method extracts only small segments, these segments will typically constitute a large shape with an accurate contour. Hence, brushing is a suitable operation for the manual assignment of a complete shape in an image for a further analysis of features and matchings. Our interface checks whether a brush trajectory crosses a target. Figure 13 shows that our interaction can support matching visualization in the desired region through simple brushing.

Lassoing Lassoing is designed for defining an irregular selected area. We employ this interaction in the feature scatterplot view and the feature view. The lassoing operation generates a trajectory based on a number of marker points. We then adopt the work of Moreira and Santos [MS07] to generate a concave hull representing the selected region. All related elements in this concave hull will

(a) A brushing trajectory through segments.

(b) Only the matching lines linked to the selected segments are highlighted.

(c) A brushing trajectory through bundle blocks.

(d) Only the matching lines linked to the selected bundle blocks are highlighted.

Figure 13: *An example of the brushing interaction.*

(a) Lasso selection in the magnified feature view.

(b) Highlighted feature points and matching lines in the matching view.

Figure 14: *Lassoing selection of the feature points in the feature view.*

be highlighted. Figure 16(c) depicts an example of the lassoing interaction in the feature scatterplot view. Figure 14 outlines a lasso interaction example in the feature view.

Touching The touching interaction is implemented only in the feature view, the group view and the circular heat view. If a user touches an element in one of these views, other related elements will be highlighted. For example, if the user touches an arc glyph in the circular heat view, all features and all matching lines in the matching view that link to this element will be selected. In the group view, touching a group block will cause the related bundled matching lines to be highlighted in the matching view as shown in Figure 15.

7. Case Studies

We evaluated the usefulness of the VisFM system in three distinct usage scenarios.

(a) Selected group node.

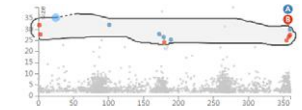
(b) Highlighted bundles in a matching view.

Figure 15: *Touching on a group node in the group view leads to the bundles highlighting in the matching view.*

(a) Selected group node and group matchings in a group view.

(b) Selected group node and group matchings in a group view.

(c) Selected group node and group matchings in a group view.



(d) Selected group node and group matchings in a group view.

Figure 16: *Matching line exploration on a special object of the images through a lassoing interaction.*

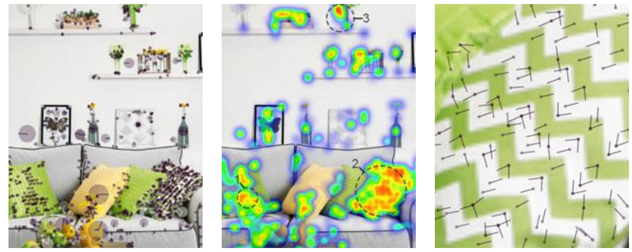
7.1. Visual evaluation of SIFT

To visually verify that the key-point features identified by SIFT are invariant to transformations, such as rotation and scale, we used VisFM to visualize the feature matching result of SIFT. In this usage scenario, we set the matching method to FLANN and the segment size to 84. First, we captured an image A in a real scene and captured an image B with a 165ºforward camera rotation and a bit magnification. We then input these two images to VisFM. From a simple visualization of all matching lines, it was difficult to find the matched and mismatched regions on the image as shown in Figure 16(a).

Through the observation in a feature view, we found that the key features of the input images, such as on the white signs, have large feature size. Thus, we adopt lassoing operation in the feature scatterplot view to select large size features and highlight the related matching lines. Through the filtered matching line visualization, we observed that all features are matched well on the white signs. By using VisFM, we verified that SIFT was effective for matching an image from a set of transformed images.

7.2. High-density feature exploration

The pair of images considered in this case study included thousands of features; thus, the high-density features and matching lines caused severe visual clutter. In this usage scenario, we set the



(a) High-density feature points. (b) A feature heat map. (c) Magnified feature view.

Figure 17: *High-density feature exploration in the feature view.*

(a) (b) (c)

Figure 18: *Matching lines exploration in the matching view. (a) Crowd matching lines. (b) Touching selection in the group view. (c) Highlighted bundles in the matching view.*

matching method to FLANN and the segment size to 64. An image with abundant features requires smaller segment size in order to analyse features in a microcosmic form. Since directly visualizing all feature points was difficult to reveal the significant regions for further analysis, we switched to the heat map option to observe the density of the features as shown in Figure 17(b). In the heat map, we found three high-density regions. We then zoomed in to the second one to see the feature distribution. In the magnified view as shown in Figure 17(c), we found that many feature points are detected in the same position with distinct angles. It explained why the features in the magnified region were of high density. In order to further explore the feature matchings on the selected region, we highlighted the matching bundles related to it through a touching in the group view (Figure 18b). Figure 18(c) shows our more intuitive matching result compared with the unbundled layout as shown in Figure 18(a). The wrong matching result in Figure 18(c) provides a clue that the improvement of FLANN should take multi-angle of the feature into account.

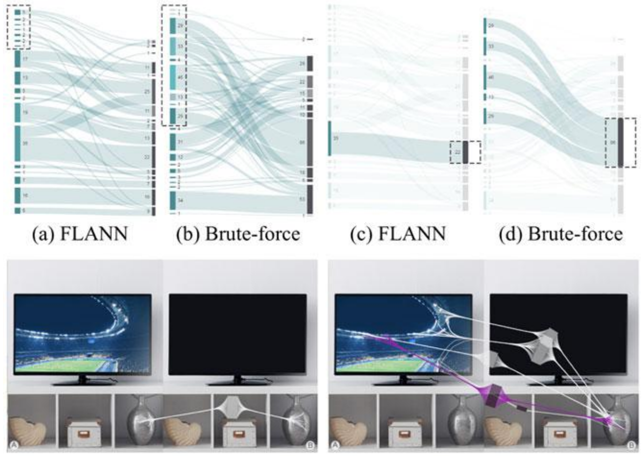
7.3. Visual evaluation of matching approaches

In this usage scenario, we wanted to evaluate the performance of the FLANN algorithm. Figure 19(a) showed an intuitive correct matching result processed by using FLANN matching algorithm between a pair of identical images. However, which region of image B matches the highlighted region in image A was not clear in Figure 19(a). In VisFM, the bundle block brushing selection as shown in Figure 19(b) and the group view as shown in Figure 19(c) showed a very clear matching structure without overlapping.

Hence, we summarized that FLANN matching algorithm performed well on the matching of two identical images.

(a) (b) (c)

Figure 19: *The matching view and the group view clearly indicate the matching result of the image segments. (a) All matching lines. (b) Highlighted bundles. (c) Group matchings.*



(a) FLANN (b) Brute-force (c) FLANN (d) Brute-force

(e) Highlighted bundles through a touch in a group view (FLANN) (f) Highlighted bundles through a touch in a group view (Brute-force)

Figure 20: *Exploration of the matching performance between two algorithms.*

We selected the second pair of images for comparing the matching performance between FLANN and brute-force method. The selected images were highly similar; however, the correct feature matching rate would be low because of the difference in the content displayed on the TV. We found that the content on the TV screen was ignored by using the FLANN method. Figure 20(a) showed that fewer feature matchings were generated related to the top part of the image. Less crossing matching links in Figure 20(a) showed that FLANN achieved a higher matching rate than using brute-force method as shown in Figure 20(b). We then highlighted the bundles through the touching interaction (Figures 20c and d) in the group view and observed the matching result in the matching view. Figure 20(e) represented the correct matching, while Figure 20(f) showed the wrong matching from the TV screen to a bottle. We shared our finding with the related expert and he explained that FLANN was based on the nearest neighbour search thus avoided the wrong matching with large displacement.

8. Informal Expert Feedback

We cooperated with an expert who tested the use of VisFM for understanding and improving an image retrieval algorithm in his field. Image retrieval is the task of finding the most closely matching image from a data set of thousands of images based on an input image. The basic strategy of the expert was to improve the effectiveness of image retrieval based on an image smoothing method. When the images in a data set have been smoothed, the speed of image retrieval will be improved because of the reduction in the number of features; however, the correct matching rate of state-of-the-art matching methods on these smoothed images is simultaneously reduced. Therefore, the expert wanted to use VisFM to observe the change in matching behaviour when images are subjected to a smoothing operation. His goal was to explore the potential improvement of the matching method.

The expert selected two images with abundant features as an example to perform a visual analysis of feature matchings. He chose FLANN as the feature matching method. First, into one instance of the VisFM system, he put two images of the same scene that were captured from different camera poses. He used another instance of the VisFM system loaded with two smoothed images to compare the feature matchings between the original images and the smoothed images. The feature view and the matching view in VisFM provided an overview of the image feature data. The expert noted that the feature view was helpful for assessing the effectiveness of the smoothing methods. Using the VisFM system, he was able to select feature points on the image structure and to observe the corresponding feature matchings. If a smoothing method can preserve the structure of an image, then many key features can still be extracted from the smoothed image.

The expert applied the *L0 Gradient Minimization* (LGM) method [XLXJ11] to smooth the pair of images and compared the corresponding feature matching result with the result obtained on the images when they had been smoothed using the *relative total variation* (RTV) approach [XYXJ12]. At the beginning, the expert preferred RTV because it was an improved smoothing approach of LGM. However, the result in VisFM surprised him that LGM-based smoothing could produce a better feature matching result than RTV-based smoothing as shown in Figure 21. Although the RTV-based method preserved the image structure after smoothing, the expert found that a few sparse incorrect matchings between the two smoothed images still remained as shown in Figure 21(b). These incorrect matchings could still influence correct image matching in an image retrieval application.

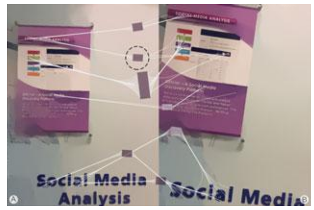
The expert then studied the group relationships in the group view. He found that LGM-based group node had higher GMW value compared with the RTV-based result. He touched a group node with a deep colour and observed almost correct segment matching in the matching view. Later, he magnified the correctly matched features on LGM-based images in the feature view and found some un-smoothed features on the smoothed regions, which might cause the higher correct matching rate of LGM approach.

Based on the observations described above, he designed and proposed an improved feature matching approach that was more suitable for the LGM-based smoothed images. The proposed approach considered the feature group matching results to improve the correct matching rate and speed up the image retrieval process. The expert named the proposed approach structure-based image retrieval.

(a) Group matching result (RTV).

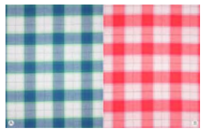
(b) Bundle matching result (RTV).

(c) Group matching result (LGM).

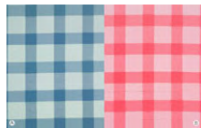


(d) Bundle matching result (LGM).

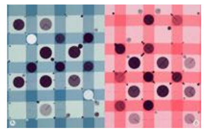
Figure 21: *The matching result comparison of two smoothing approaches. A dotted circle indicates a wrong group matching.*



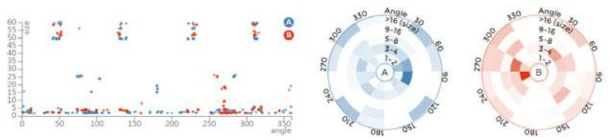
(a) Original textile images.



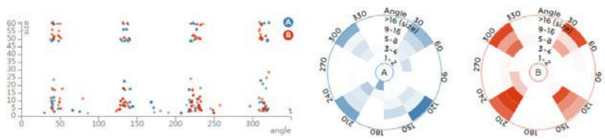
(b) Smoothed textile images.



(c) Feature points of the smoothed textile images.



(d) Feature characteristics of the original images.



(e) Feature characteristics of the smoothed images.

Figure 22: *Feature characteristics and the matching results of the smoothed images.*

Furthermore, the expert applied the VisFM system on a pair of the textile images in order to verify if his approach was suitable for the textile image retrieval application. Image analysis and retrieval methods are frequently applied in textile research; thus, an understanding of feature matching is required for textile images with repetitive patterns. He applied LGM approach on the original textile images and generated the smoothed textile images as shown in Figures 22(a) and (b).

In order to observe the feature difference of two pairs of textile images, he started up two VisFM systems on two displays. Originally, he observed the feature characteristics in a feature view as shown in Figure 22(c). Then, he adopted the brushing selection on the segment in an image in order to observe the matched segments in another image. However, the correct matching rate was not achieved. Later, he found that the circular heat view as shown in Figures 22(d) and (e) was helpful for image matching analysis in the case of repetitive patterns. The feature-angle-based matching selection was convenient for exploring textile feature matchings.

The expert planned to finalize the proposed approach and apply it in a real-world image retrieval application. In addition, the expert expressed that he appreciated the value of the CCA when observing the features and matchings on the smoothed images.

9. Conclusions

In this paper, we present a new visual analysis system, VisFM, for understanding and exploring image feature matchings to satisfy the requirements of domain researchers. We design feasible visualization views to assist users in effectively interpreting and retrieving image features and pair matchings. We also offer a rich set of interactions for exploring image features and matchings and for enabling parameterization of feature matching algorithms. We introduce the edge bundling technique into a new domain and provide a complete system that is simple to implement and use. By using a visual analysis system, a user can gain a more intuitive understanding of feature matching algorithms. Case studies and the expert feedback demonstrate the usefulness and effectiveness of VisFM.

There are still a few limitations of the VisFM system. First, the comparison of results for two pairs of images cannot be performed in a single instance of the VisFM system. Currently, the user must compare the matching results for two pairs of images by means of two instances of the VisFM system on two different displays. A potential improvement would be to integrate statistical information for two pairs of images in the same view. Second, the interactions in the group view should provide a more intuitive means of locating related segments and their matchings. Thus, the work of MapTrix [YDGM17] seems to offer a feasible many-to-many approach for further improving the visualization and interaction performance in the group view.

The approaches adopted in VisFM can be applied to image features detected through other approaches, such as SURF [BTVG06] and HoG [ZYCA06]. Another desirable extension of the VisFM system, as noted by the expert, is to consider dynamic feature matchings, e.g. the analysis of matchings between a single image and a video. Moreover, we plan to consider the group hierarchy in the matching view to further explore the matching lines.

Acknowledgements

The authors wish to thank Dr. Xiujun Zhang and Dr. Yu Han for the detailed research discussions. This work was supported in part by IGRF PolyU 152142/15E, PolyU Departmental General Research Fund 4-ZZFF and Natural Science Foundation of China under Grant Nos. 61532002, 61672237. George Baciu is the corresponding author.

References

[ASS*12] ACHANTA R., SHAJI A., SMITH K., LUCCHI A., FUA P., SUSSTRUNK S.: Slic superpixels compared to state-of-the-art superpixel methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence 34*, 11 (2012), 2274–2282.

[BOH11] BOSTOCK M., OGIEVETSKY V., HEER J.: D^3 data-driven documents. *IEEE Transactions on Visualization and Computer Graphics 17*, 12 (2011), 2301–2309.

[BRH*17] BACH B., RICHE N. H., HURTER C., MARRIOTT K., DWYER T.: Towards unambiguous edge bundling: Investigating confluent drawings for network visualization. *IEEE Transactions on Visualization and Computer Graphics 23*, 1 (2017), 541–550.

[BTVG06] BAY H., TUYTELAARS T., VAN GOOL L.: Surf: Speeded up robust features. In *Proceedings of European Conference on Computer Vision* (2006), Springer, pp. 404–417.

[CCRV08] CALDWELL B., COOPER M., REID L. G., VANDERHEIDEN G.: Web content accessibility guidelines (wcag) 2.0.

[DMM*14] DWYER T., MEARS C., MORGAN K., NIVEN T., MARRIOTT K., WALLACE M.: Improved optimal and approximate power graph compression for clearer visualisation of dense graphs. In *Proceedings of 2014 IEEE Pacific Visualization Symposium (PacificVis)* (2014), IEEE, pp. 105–112.

[ED07] ELLIS G., DIX A.: A taxonomy of clutter reduction for information visualisation. *IEEE Transactions on Visualization and Computer Graphics 13*, 6 (2007), 1216–1223.

[EHP*11] ERSOY O., HURTER C., PAULOVICH F., CANTAREIRO G., TELEA A.: Skeleton-based edge bundling for graph visualization. *IEEE Transactions on Visualization and Computer Graphics 17*, 12 (2011), 2364–2373.

[EKS*96] ESTER M., KRIEGEL H.-P., SANDER J., XU X.: A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of KDD* (1996), vol. 96, pp. 226–231.

[FR91] FRUCHTERMAN T. M., REINGOLD E. M.: Graph drawing by force-directed placement. *Software: Practice and Experience 21*, 11 (1991), 1129–1164.

[HET12] HURTER C., ERSOY O., TELEA A.: Graph bundling by kernel density estimation. *Computer Graphics Forum 31*, 3pt1 (2012), 865–874.

[HET13] HURTER C., ERSOY O., TELEA A.: Smooth bundling of large streaming and sequence graphs. In *Proceedings of 2013 IEEE Pacific Visualization Symposium (PacificVis)* (2013), pp. 41–48.

[Hol06] HOLTEN D.: Hierarchical edge bundles: Visualization of adjacency relations in hierarchical data. *IEEE Transactions on Visualization and Computer Graphics 12*, 5 (2006), 741–748.

[HVW09] HOLTEN D., VAN WIJK J. J.: Force-directed edge bundling for graph visualization. *Computer Graphics Forum 28*, 3 (2009), 983–990.

[KMN*02] KANUNGO T., MOUNT D., NETANYAHU N., PIATKO C., SILVERMAN R., WU A.: An efficient k-means clustering algorithm: Analysis and implementation. *IEEE Transactions on Pattern Analysis and Machine Intelligence 24*, 7 (2002), 881–892.

[KSH12] KRIZHEVSKY A., SUTSKEVER I., HINTON G. E.: Imagenet classification with deep convolutional neural networks. In *Proceedings of the International Conference on Advances in Neural Information Processing Systems* (2012), pp. 1097–1105.

[Low04] LOWE D. G.: Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision 60*, 2 (2004), 91–110.

[LRFH13] LIN S., RITCHIE D., FISHER M., HANRAHAN P.: Probabilistic color-by-numbers: Suggesting pattern colorizations using factor graphs. *ACM Transactions on Graphics 32*, 4 (2013), 37:1–37:12.

[Mac67] MACQUEEN J.: Some methods for classification and analysis of multivariate observations. In *Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics* (Berkeley, CA, USA, 1967), University of California Press, pp. 281–297.

[MG13] MAYORGA A., GLEICHER M.: Splatterplots: Overcoming overdraw in scatter plots. *IEEE Transactions on Visualization and Computer Graphics 19*, 9 (2013), 1526–1538.

[ML09] MUJA M., LOWE D. G.: Fast approximate nearest neighbors with automatic algorithm configuration. In *Proceedings of the International Conference on Computer Vision Theory and Application (VISSAPP'09)* (2009), pp. 331–340.

[MS07] MOREIRA A., SANTOS M. Y.: Concave hull: A k-nearest neighbours approach for the computation of the region occupied by a set of points. In *Proceedings of the International Conference on Computer Graphics Theory and Applications* (2007), pp. 61–68.

[QZH15] QIAN X., ZHAO Y., HAN J.: Image location estimation by salient region matching. *IEEE Transactions on Image Processing 24*, 11 (2015), 4348–4358.

[SAA*11] SEBASTIAN B., ANDRIENKO G., ANDRIENKO N., SCHRECK T., LANDESBERGER T. V.: Interactive analysis of object group changes over time. In *Proceedings of International Workshop on Visual Analytics (EuroVA 2011)* (2011), pp. 41–44.

[Sil86] SILVERMAN B. W.: *Density Estimation for Statistics and Data Analysis*, vol. 26. CRC Press, Boca Raton, FL, 1986.

[SMNR16] SUN M., MI P., NORTH C., RAMAKRISHNAN N.: Biset: Semantic edge bundling with biclusters for sensemaking. *IEEE Transactions on Visualization and Computer Graphics 22*, 1 (2016), 310–319.

[TE10] TELEA A., ERSOY O.: Image-based edge bundles: Simplified visualization of large graphs. *Computer Graphics Forum 29*, 3 (2010), 843–852.

[VKP*16] VONDRICK C., KHOSLA A., PIRSIAVASH H., MALISIEWICZ T., TORRALBA A.: Visualizing object detection features. *International Journal of Computer Vision 119*, 2 (2016), 145–158.

[vL07] VON Luxburg U.: A tutorial on spectral clustering. *Statistics and Computing 17*, 4 (2007), 395–416.

[WZL13] WANG X.-J., ZHANG L., LIU C.: Duplicate discovery on 2 billion internet images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops* (2013), pp. 429–436.

[XLXJ11] XU L., LU C., XU Y., JIA J.: Image smoothing via l0 gradient minimization. *ACM Transactions on Graphics 30*, 6 (2011), 174:1–174:12.

[XYXJ12] XU L., YAN Q., XIA Y., JIA J.: Structure extraction from texture via relative total variation. *ACM Transactions on Graphics 31*, 6 (2012), 139:1–139:10.

[YDGM17] YANG Y., DWYER T., GOODWIN S., MARRIOTT K.: Many-to-many geographically-embedded flow visualisation: An evaluation. *IEEE Transactions on Visualization and Computer Graphics 23*, 1 (2017), 411–420.

[ZF14] ZEILER M. D., FERGUS R.: Visualizing and understanding convolutional networks. In *Proceedings of European Conference on Computer Vision* (2014), Springer, pp. 818–833.

[ZXYQ13] ZHOU H., XU P., YUAN X., QU H.: Edge bundling in information visualization. *Tsinghua Science and Technology 18*, 2 (2013), 145–156.

[ZYCA06] ZHU Q., YEH M.-C., CHENG K.-T., AVIDAN S.: Fast human detection using a cascade of histograms of oriented gradients. In *Proceedings of 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (2006), vol. 2, IEEE, pp. 1491–1498.

[ZYS09] ZHOU H., YUAN Y., SHI C.: Object tracking using sift features and mean shift. *Computer Vision and Image Understanding 113*, 3 (2009), 345–352.