Snap & Play: Auto-Generated Personalized Find-the-Difference Game

SI LIU*, QIANG CHEN*, and SHUICHENG YAN, National University of Singapore
CHANGSHENG XU, Institute of Automation, Chinese Academy of Science,
China-Singapore Institute of Digital Media
HANQING LU, Institute of Automation, Chinese Academy of Science

In this article, by taking a popular game, the Find-the-Difference (FiDi) game, as a concrete example, we explore how state-of-the-art image processing techniques can assist in developing a personalized, automatic, and dynamic game. Unlike the traditional FiDi game, where image pairs (source image and target image) with five different patches are manually produced by professional game developers, the proposed Personalized FiDi (P-FiDi) electronic game can be played in a fully automatic Snap & Play mode. *Snap* means that players first take photos with their digital cameras. The newly captured photos are used as source images and fed into the P-FiDi system to autogenerate the counterpart target images for users to *play*. Four steps are adopted to autogenerate target images: enhancing the visual quality of source images, extracting some changeable patches from the source image, selecting the most suitable combination of changeable patches and difference styles for the image, and generating the differences on the target image with state-of-the-art image processing techniques. In addition, the P-FiDi game can be easily redesigned for the im-game advertising. Extensive experiments show that the P-FiDi electronic game is satisfying in terms of player experience, seamless advertisement, and technical feasibility.

Categories and Subject Descriptors: H.5.1 [**Information Interfaces and Presentation**]: Multimedia Information Systems; I.2.1 [**Artificial Intelligence**]: Applications and Expert Systems

General Terms: Design, Algorithms, Performance

Additional Key Words and Phrases: Automatic personalized system, image analysis, human–computer interaction, image editing, image modeling

ACM Reference Format:
Si Liu, Qiang Chen, Shuicheng Yan, Changsheng Xu, and Hanqing Lu. 2014. Snap & Play: Auto-generated personalized find-the-difference game. ACM Trans. Intell. Syst. Technol. 5, 4, Article 65 (December 2014), 18 pages.
DOI: http://dx.doi.org/10.1145/2668109

1. INTRODUCTION

The Entertainment Software Association[1] has reported that more than 200 million hours are spent each day playing electronic games (e.g., computer, video games) in the United States. The huge market for electronic games motivates us to think about

[1]http://www.theesa.com/facts/.

*Indicates equal contribution.

Authors' addresses: S. Liu, Institute of Information Engineering, Chinese Academy of Sciences, National University of Singapore; email: fifthzombiesi@gmail.com; Q. Chen, National University of Singapore; email: chenqiang@nus.edu.sg; C. Xu, Institute of Automation, Chinese Academy of Science, China-Singapore Institute of Digital Media; email: csxu@nlpr.ia.ac.cn; H. Lu, Institute of Automation, Chinese Academy of Science; email: luhq@nlpr.ia.ac.cn.

DOI: http://dx.doi.org/10.1145/2668109

Fig. 1. The interface for the basic version of P-FiDi. For better viewing, please see the original color.jpg file for all images.

one problem: *Can state-of-the-art image processing techniques help develop automatic personalized games*?

In this article, by taking a popular game, the Find-the-Difference (FiDi) game, as a concrete example, we explore the merits brought by the increasingly mature image processing techniques for game design. The FiDi game, as a typical representative of a puzzle/logic game,[2] is very popular. Statistics show that in QQ Game,[3] the biggest online game society in China, on average about 180,000 players were simultaneously online playing the FiDi game at 8:00 PM from March 1 to March 31, 2013. By contrast, the online number is 20,387 for Texas hold 'Em Poker, which is one of the most popular forms of poker.[4]

A typical interface of FiDi is shown in Figure 1. Two images are displayed in a pair. The left image is called the source image, while the right one is called the target image. The source image and target image are almost the same except for several (generally five) small different patches generated by Photoshop or other techniques. The players are required to identify all differences within the limited time.

The success of the traditional FiDi game is largely due to the *naturalness* of the target image. Generally, obvious artifacts can be easily identified, which reduces the challenge of the game. In spite of its great success, the FiDi game is imperfect in the following aspects. First, the target images are manually generated by professional image processing experts using professional image processing software, such as Photoshop. This is very time-consuming and tedious. It even becomes infeasible for a very large-scale image set. Second, the images are provided by game developers, which may not well suit players' taste. Finally, the number of available image pairs is limited and the differences for each image pair are fixed, and thus experienced players can remember all the differences, which may easily lead to cheating in gaming process.

We propose to intelligently apply a series of image processing techniques to improve FiDi, producing the Personalized Find-the-Difference (P-FiDi) game, which can automatically generate the target images. Together with P-FiDi, we also propose a new game mode called *Snap & Play*. Player first *snap* something by their mobile phone,

[2]http://en.wikipedia.org/wiki/Spot_the_difference.
[3]http://qqgame.qq.com.
[4]http://en.wikipedia.org/wiki/Texas_hold_'em.

and then the instantly captured photo is used as the source image in the P-FiDi game. The autogenerated target image paired with the source image together serve as game materials to *play*. P-FiDi has a similar interface and game rules as FiDi, shown in Figure 1.

P-FiDi should inherit the aforementioned good characteristics of the traditional FiDi game. **Naturalness** is the most important principle when designing the P-FiDi game. Differences should naturally and seamlessly fuse with other patches in the image. Also, the imperfectness of the traditional FiDi game should be tackled via state-of-the-art image processing techniques. The P-FiDi game has several unique characteristics: (1) to alleviate the image editing experts from tedious manual editing, the P-FiDi system is **automatic**. The whole autogeneration process includes automatic generation of changeable patches, automatic patch & change style selection, and target image autogeneration. We automatically generate the changeable patches by state-of-the-art image segmentation [Comaniciu and Meer 2002], face detection [Viola and Jones 2004], and text detection techniques [Epshtein et al. 2010]. The automatic patch & change style selection is achieved via a binary quadratic programming optimization process. Finally, the target image is autogenerated by making use of many automatic image editing algorithms, such as Poisson editing [Pérez et al. 2003]. (2) P-FiDi is a **personalized** system, where users can play with their own photos. In the traditional FiDi game, players have to play with images provided by the game developers. Personalization can arouse players' interest and affinity for the game. (3) Finally, to avoid experienced players cheating and possibly getting bored in the traditional FiDi game, the P-FiDi system is **dynamic**. The same source image may produce different target images when played at different times. We make P-FiDi dynamic in several ways. For example, we randomly choose between color transfer and color harmony methods to change one patch's color.

To sum up, many image processing, computer vision and computer graphics techniques are intelligently combined to enhance automation, personalization, and variability of the proposed P-FiDi game. Besides the unique Snap & Play gaming mode, we also extend P-FiDi for the seamless advertisement embedding scenario.

An earlier version of this earlier is published in Liu et al. [2011]. The main differences between these two versions are summarized as follows. First, an extension toward in-game advertisement is added, which provides the game the extra possibility to make profit. Second, qualitative results for each change style as well as a combination of all change styles of the P-FiDi system are presented in the Experiments section. Third, qualitative results to compare our patch & change style selection strategy and a random baseline are shown in the Experiments section. Finally, more user studies about the characteristics of P-FiDi and its extension are shown.

2. RELATED WORK

In this section, we briefly introduce the background of the FiDi game and some image processing techniques used in our P-FiDi system. Finally, we review the recent development of in-game advertising.

FiDi Game: The FiDi game belongs to one important computer game category, that is, puzzle/logic games. FiDi can help players develop observation capabilities and relaxed mood while simultaneously browsing many high-quality photos. FiDi games are often found in children's puzzle books and in newspapers. Recently, FiDi has feature in computer, online, and mobile games.[5] As far as we know, our system is the first to automatically generate personalized "find-the-difference" image pairs, which shows the uniqueness of the work.

[5]http://www.theesa.com/facts/.

Image Processing Techniques: To automatically generate target images in P-FiDi, several state-of-the-art techniques are utilized. There is a lot of literature on enhancing the visual experience [Bhattacahrya et al. 2010; Yeh et al. 2010; Reinhard et al. 2001]. Some work [Cohen-Or et al. 2006; Reinhard et al. 2001] focuses on how to change the color of the foreground to match the color of the background. Poisson editing [Pérez et al. 2003] serves as a commonly used technique to insert items into images. Due to acceptable accuracies in real applications, face detection [Viola and Jones 2004] and text detection [Epshtein et al. 2010] are often adopted to assist in parsing the higher-level semantics in images. Scene recognition [Lazebnik et al. 2006] can help globally understand the images' content. With these techniques serving as the cornerstones of the whole P-FiDi system, we propose a patch & change style selection process, which is the key step of our system. This step is formulated as a binary quadratic programming problem, which can be effectively solved by an off-the-shelf optimization package.

In-Game Advertising: As the computer-game-playing population expands, in-game advertisements are expanding as well. Massive Incorporated,[6] a creator of dynamic game advertisements, estimates the in-game advertising market could grow to 1 billion globally by 2014. P-FiDi can be extended in a straightforward way to in-game advertisements. P-FiDi-based advertisements are nonintrusive and interesting compared with their counterparts [Mei et al. 2008] in web image advertisement. Besides in-game advertisement, the game can be designed as a carrier for other purposes.

3. SYSTEM OVERVIEW

3.1. P-FiDi Interface and Rules

A typical interface is shown in Figure 1. Two images, one source image and one target image, lie in the central area. The source image is captured by one's mobile device or downloaded from a user's social network account. The target image is generated by the P-FiDi system. Players should identify all five different patches by clicking (or touching, for iPad-like devices) within the fixed time. Clicking either image is acceptable for the game. After each correct identification, the boundary of the corresponding region is marked by an ellipse. A scoreboard is shown in the lower left corner of the interface. The player gets several scores for each correct identification. Above the scoreboard, the remaining time is displayed. After identifying all the different patches within the given time, the next image pair is automatically loaded. In the top left corner, three optional difficulty levels (i.e., easy, middle, and difficult) are shown. Players can select a level to start playing. When stuck, one may seek help in two ways. As shown in the left part, the magnifier can help point out one different patch, and the clock can extend the remaining time. Players have a limited chance to ask for help. To prevent players from cheating by randomly clicking the screen, the remaining time elapses double faster if the player clicks (or touches) the wrong position. Finally, a high score list pops up after the end of the game.

3.2. P-FiDi Technical Flow

The technical flowchart of the P-FiDi system is illustrated in Figure 2. First, one player takes a picture as shown in Figure 2(a). Then, if necessary, the source image may be aesthetically enhanced for a better visual experience, as shown in Figure 2(b). After that, the photo is segmented into small changeable patches, typically 30 to 50 patches, as shown in Figure 2(c). Moreover, face detection, facial component alignment, and text detection are used to generate some semantic patches if available. They all serve as the changeable patches in the following steps. According to the gaming rules, up

[6]http://www.massiveincorporated.com/.

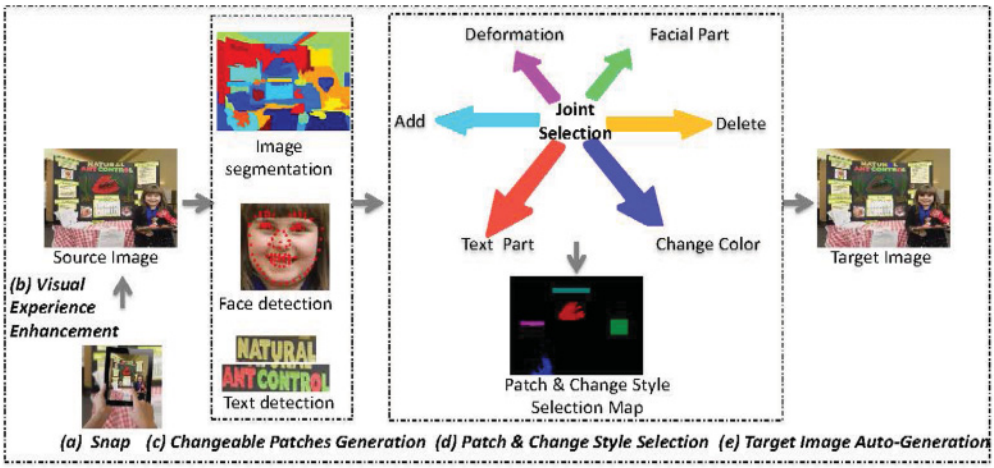

Fig. 2. The framework for the P-FiDi game. The instantly captured image (a) is used as the source image. It is visually enhanced whenever necessary (b). Three kinds of changeable patches (oversegmented patches, facial patches, and text patches) are extracted from the source image (c). Then, we intelligently decide the most suitable changeable patches and their corresponding change styles by a binary optimization process (d). Finally, the target image is generated by applying several image processing techniques (e). Source of image http://www.flickr.com/photos/111945105@N07/11598322025/.

to M (5 in this work) patches are selected to change and their corresponding change styles are determined simultaneously. In Figure 2(d), the optimal patch & change style combination is selected. The result is represented by a map, where different change styles are colored differently. Finally, the patches undergo the selected changes and the counterpart target image is generated in Figure 2(e).

4. TECHNIQUE DETAILS

The P-FiDi system contains four key components: visual experience enhancement, changeable patches generation, patch & change style selection, and target image auto-generation. In this section, we will introduce each component in detail and discuss our difficulty level setting strategy.

4.1. Visual Experience Enhancement

A key factor to foster players' adhesion is the pleasant gaming experience. However, most players are not professional photographers, so we perform photo-quality aesthetic enhancement when necessary. We mainly focus on two kinds of photos: (1) images containing single foreground objects and (2) scene images containing no definite foreground, such as landscapes or seascapes [Bhattacahrya et al. 2010].

Image with a Single Foreground Object: To judge whether the image contains a single object, we first segment the image into several patches (e.g., total n patches) using the mean-shift algorithm [Comaniciu and Meer 2002]. We also calculate the saliency value of each pixel by the spectral residual approach [Hou and Zhang 2007]. The saliency value of a patch is calculated by averaging all pixels' saliency values within it. Then, all the patches are decreasingly ranked by their saliency values, denoted as $[p_1, p_2, \ldots, p_n]$. If the saliency ratio between the top-ranked and second-ranked patch is sufficiently high, that is, $p_1/p_2 \geq h$, then the top-ranked patch is considered as the only foreground object inside the image. h is set as 1.25 in the P-FiDi system. For the images with a single object, three methods are adopted to highlight the foreground. The first method is the rule of thirds, a well-known photograph composition rule [Grill and

Scanlon 1990; Liu et al. 2010]. The idea is to place main objects at roughly one third of the horizontal or vertical dimension of a photo. The second method is to increase the brightness around the main object, so that the generated target image has stage effects. The last method is to blur the background, which indirectly highlights the foreground object.

Scene Images Without Definite Foreground: First, we evaluate the quality of the scene image to decide whether the visual experience enhancement step is required by the Personalized Photograph Ranking (PPR) method [Yeh et al. 2010]. For the relatively lower-quality photos, we use color transfer [Reinhard et al. 2001] to propagate the color style from professional photos to the user's photos. To this end, we collect a professional photo dataset containing 5,000 images. To construct the database, we first define several scene-related keywords, such as highway, street, and bedroom. Then the images favored by more people are crawled from Flickr. Given a user-provided scene photo, we first select a semantically similar image from the professional photo dataset based on GIST feature [Oliva and Torralba 2001] similarities. The color style of the selected professional image is transferred to the user's photo.

4.2. Changeable Patches Generation

Three kinds of changeable patches, that is, oversegmented patches, facial patches, and text patches, are considered in the P-FiDi game. The mean-shift algorithm [Comaniciu and Meer 2002] is adopted to produce several oversegmented patches that are composed of several visually similar and spatially adjacent pixels. Besides the oversegmented patches, two other kinds of semantic patches, that is, face and text, are also considered. With face detection and facial component alignment algorithms [Viola and Jones 2004], we obtain the locations of several semantic facial patches, including two eyes, a nose, and a mouth, for near-frontal faces. Text patches [Epshtein et al. 2010] can express rich information, and therefore are considered as the third kind of candidate changeable patches. To sum up, we have totally N changeable patches, including n segmentation patches and some facial and/or text regions if available.

4.3. Patch & Change Style Selection

After obtaining several candidate changeable patches, we determine *where to change* and *how to change*, which is a patch & style selection process. In this subsection, we first introduce the formulation of the selection process and then introduce the definition of a very important component of the formulation, that is, patch & style fitness matrix.

4.3.1. Formulation of Patch & Change Style Joint Selection. The optimal combination of changeable patches and change styles is denoted by a binary $N \times K$ *Patch & Style Joint Selection Matrix* S, where K (6 in our system) is the number of candidate change styles, such as changing color, adding an item. The nonzero element S_{ij} indicates that the i^{th} patch is selected to undergo the j^{th} change style. Each row represents a patch and each column corresponds to a change type. For the convenience of presentation, the row vectors of matrix S are concatenated into an indicator vector s. The $j^{th}, (K + j)^{th}, (2K + j)^{th}, \ldots, ((N - 1)K + j)^{th}$ elements correspond to the i^{th} patch. Our target is to find an optimal s satisfying the following two objectives:

Objective I: Overall Fitness: The best patch & change style combination means the patches and change styles should suit each other. On the one hand, each kind of patch has its favored change styles. For example, for small patches, changing its color is much easier than inserting a dragonfly into it. On the other hand, each kind of change style has its preferable patches. For example, technically, it is easier to insert a dragonfly into flat, solid blue sky than cloudy sky. To model the fitness between patches

and change styles, we define a patch & style fitness matrix C, which is of the same size S, with element C_{ij} corresponding to the fitness value of patch i undergoing the j^{th} change style. The definition of C shall be given in Section 4.3.2. A larger value in C indicates stronger fitness. Similarly, we concatenate the row vectors of C as a vector c. To sum up, the first objective of patch & change style selection is to maximize the overall fitness:

$$\max_s c^T s. \tag{1}$$

Objective II: Spatially Even Distribution: Intuitively, distributing the changed patches evenly in the image is important. If two changed patches are too close, it is difficult for players to judge whether they are two smaller different patches or just one large different patch. To describe the patches' spatial distances, we define an $N \times N$ matrix D, with each element describing the pairwise spatial distance between two image patches. We also define a matrix R to sum over s for each patch. In the i^{th} row of R, the $((i-1)\times K+1)^{th}$ to the $(i \times K)^{th}$ elements are of value 1 and others are 0. Then $(Rs)_i$ indicates how many change styles have been selected for the i^{th} patch. Then the objective to maximize the overall spatially dispersed distribution can be expressed as

$$\max_s \frac{1}{2M}(Rs)^T D(Rs). \tag{2}$$

Four constraints are also imposed over s to achieve a reasonable patch & style selection.

Constraint I: Binary Property Constraint: First, s should be binary, indicating whether one patch is selected, that is, $s \in \{0,1\}^{(N\times K)\times 1}$.

Constraint II: Total Change Number Constraint: To ensure the number of changed patches is M, we have $\sum_{i=1}^{N\times K} s_i = M$. In this article, M is set to be 5.

Constraint III: Each Patch with at Most One Change: Each patch can only undergo one kind of change style, namely, $(Rs)_i \le 1, \forall\, i \in \{1,2,\ldots,N\}$.

Constraint IV: Change Style Diversity: The diversity of change style is important for the fun of the game, and the system should generate as many kinds of variations as possible for each image. In our system, we restrict that each change style cannot appear more than twice. For example, at most two patches can change color. So we define a $K \times (N \times K)$ indicator matrix T, where for the j^{th} row, the j^{th}, $(K+j)^{th}$, $(2K+j)^{th}$, $\ldots$, $((N-1)K+j)^{th}$ entries are of value 1 and others are 0. Then Ts is a $K\times 1$ vector and $(Ts)_j$ indicates how many times the j^{th} difference style is adopted. Therefore, an extra constraint is imposed as $(Ts)_j \le 2, \forall\, j \in \{1,2,\ldots,K\}$.

Overall Formulation: In summary, the patch & style selection problem is formulated as follows:

$$\begin{aligned} \max_s \quad & c^T s + \frac{1}{2M} s^T R^T D R s \\ s.t. \quad & (Rs)_i \le 1, \forall\, i \in \{1,2,\ldots,N\}; \\ & (Ts)_j \le 2, \forall\, j \in \{1,2,\ldots,K\}; \\ & \sum_{i=1}^{N\times K} s_i = M; \quad s \in \{0,1\}^{(N\times K)\times 1}. \end{aligned} \tag{3}$$

The optimization problem can be converted to a specific Binary Quadratic Programming problem [Olsson 2007]. Generally, the optimal solution cannot be obtained in

polynomial time. Therefore, many solutions, for example, Mosek[7], relax this problem either using Semi Definite Programming or spectral relaxation [Olsson 2007].

Implementation Details: In practice, the number of variables is $N \times K$ in Equation (3). In our implementation, N is about 30∼50 and K is 6. So the total variable number is about 180∼300. However, we can reduce the variable number by incorporating some priors. First, we delete all segmented patches overlapping with facial or text patches since they should be treated separately. Then, for each change style, only five patches with the highest fitness score (calculated by the method introduced in Section 4.3.2) are kept. Thus, the total computational cost is greatly reduced.

4.3.2. Definition of Patch & Style Fitness Matrix. In this part, we introduce how to define the patch & style fitness matrix. Theoretically, each patch can undergo any kind of change. However, each change style has its preference in patch properties. For example, adding small items into flat patches is more natural and easier to implement than into complicated patches. Later we present four features to measure the fitness between one patch and one specific change style. The first feature is the *patch size*, denoted as f_{sz}. It is calculated as the ratio of patch size divided by the image size. Changes on too small patches may be hard to recognize, while changes on too big patches may be too easy to identify. So median-size patches are generally preferable. The second feature is *patch shape*, denoted as f_{sp}. Square patches are favored since objects in real life are often square. We use an 8-dim edge histogram to describe the shape of a patch. The 8-dim edge histogram is calculated in the following way. First, we compute the pixels' gradient values. Then, we create the patch histograms. Each pixel within the patch casts a vote for an orientation-based histogram channel based on its gradient magnitude. Finally, the gradient strengths are locally normalized. In real implementation, we use ℓ_1 normalization. Then we calculate the entropy of the histogram to measure whether the patch is square. The third feature is the *local contrast* against surrounding regions, denoted as f_{ct}. It is the average visual similarity between the color histogram of the patch and that of all surrounding patches. The last feature is the *color variance*, denoted as f_{cv}. For each color channel, the standard deviation of all pixels within the patch is calculated, and the mean of three color channels is calculated as the final color variance. The fitness value of a patch for change style j is finally defined as

$$c_j = w_{j1} f_{sz} + w_{j2} f_{sp} + w_{j3} f_{ct} + w_{j4} f_{cv} + q, \tag{4}$$

where w_{j1}, w_{j2}, w_{j3}, w_{j4}, and w_{j5} are the weighting coefficients and q is set to be zero during training time. Different change styles have different coefficients. Learning the coefficients is a typical regression problem. To learn these coefficients, we randomly generate 100 source–target image pairs for each change style and ask users to label the qualities of changes (corresponding to the fitness value c in Equation (4)), which is then normalized to [0,1]. Summing up six change styles, we have in total 600 samples. Then we train a regressor to map the proposed features to the fitness values. When generating a target image, q in Equation (4) is a random variable uniformly distributed within $(0, 0.5)$. Different q results in a different patch & change style fitness matrix C, which further causes a different patch & change style selection vector s via Equation (3). In this way, a given source image may pair up with different target images at different times. The *dynamic* setting can effectively prevent players from cheating and keep the freshness of the game.

[7]http://www.mosek.com/.

Fig. 3. Several examples of color change. In this article, each changed patch is marked by rectangle, eclipse, or arrow. For better viewing, please see the original color.jpg file for all images.

4.4. Target Image Autogeneration

Based on the selected patches with their corresponding change styles determined in Section 4.3, in this subsection, we elaborate on autogenerating the counterpart target image with various image editing operations including changing color, deformation, adding items, erasing items, text change, and facial change.

4.4.1. Changing Color. The first change style is changing color. The human perception system is very sensitive to color change. In the P-FiDi system, two methods, namely, color transfer and color harmony, are adopted to change the color of the selected patch. The final adopted method is determined randomly. The randomness further makes the system more *dynamic*. An example is shown in Figure 3.

Color Transfer: The basic idea is to transfer the colors from the surrounding patch to the selected patch in $l\alpha\beta$ color space [Reinhard et al. 2001].

Color Harmony: To make the color change more diverse, we introduce another color change method, color harmony [Cohen-Or et al. 2006]. The RGB color of each pixel is mapped into HSV color space. Then, the colors of the selected patch are harmonized to accommodate the colors of its surrounding patches with respect to certain randomly chosen harmonic template on the hue wheel.

4.4.2. Deformation. The second change style is deformation (also known as changing the texture) of the patches. To change the texture of a selected patch, we first generate a mesh (grid) G over this region. Then we map this mesh onto a predefined template or randomized mesh $\hat{G}$. Every grid in the mesh has one-to-one correspondence with the original mesh G. For each grid, we use color interpolation to generate the new texture. Finally, the obtained new texture for the given region is embedded into the original image using Poisson editing [Pérez et al. 2003]. An example is shown in Figure 4.

4.4.3. Adding. The third change style is adding certain elements into the image. We propose two methods to implement this. In both methods, the adopted inserting technique is Poisson image editing [Pérez et al. 2003], which solves a Poisson partial differential equation with Dirichlet boundary conditions. It tries to preserve the gradient inside the inserted item and keep its color similar with the background.

Insert a small item: The first method is to insert a small item into the selected patch. To this end, we collect 120 categories of small items, including animals, daily tools, and vegetables. We further manually classify all items into nine groups based on two criteria: suitable scene and preferred location. For example, for an indoor image, it is more natural to insert a clock rather than an ambulance. Another example is that

Fig. 4. Some examples of deformation over the selected patches. Source of image: http://www.flickr.com/photos/112639541@N02/11584398433/.

Fig. 5. Some examples of adding items in the selected patches. Source of image: http://www.flickr.com/photos/112645733@N02/11584703903/.

chairs often appear at the bottom part of an image, while clocks are more likely to be in the upper part of an image. Several examples are shown in Figure 5.

Technically, we first estimate the scene class of the source image by the classifiers trained using the Scene 15 dataset [Lazebnik et al. 2006] (we merge the original 15 categories into three categories: indoor, city, and countryside). We also check the location of the selected changeable patch. Then, only the items satisfying both scene and location preferences are kept and form a candidate set. Among the candidate set, the top 10 items most similar to the selected changeable patch are selected. Then only one patch among the top 10 set is randomly selected, which serves as another strategy to enhance the *dynamic* characteristic of P-FiDi system.

Copy a patch to its neighborhood: The second method of adding items is to copy the selected patch to its neighborhood. We scan every patch around the selected patch and choose the flattest one as the destination patch. Then we use alpha matting [Levin et al. 2006] to implement the patch copy. Since many real-world scenes are characterized by repeatability, copying adjacent patches can achieve surprisingly good results sometimes.

4.4.4. Erasing Items. The fourth change style is to remove and then refill the selected patch. Two inpainting methods are implemented. One is based on examples [Criminisi et al. 2003], and the other is based on solving Partial Differential Equations (PDEs) [Chan and Shen 2000]. The basic idea of the example-based method is to copy the surrounding patch to fill the hole. The results are more natural but it has a higher computation cost. The PDE-based method generates smooth interpolation from the

Fig. 6. Some examples of erasing items/patches. Source of the image: http://www.flickr.com/photos/112645733@N02/11584711343/.

Fig. 7. Some examples of text change. Source of image: http://www.flickr.com/photos/112639541@N02/11584410953/.

outline into the hole with a lower computation cost, but may not be natural enough sometimes. We design a method to determine which method to use for a selected patch. If the color variance of the surrounding region is larger than a preset threshold, we use the example method; otherwise we use the PDE-based method. An example is shown in Figure 6.

4.4.5. Text Change. The fifth change style is related to text. We apply a text detector [Epshtein et al. 2010] to obtain the text region in the source image. Then, based on the detected text parts, we project the pixels toward the bottom border of the text bounding box. Based on the projected histogram, we can segment the text into separate letters. After obtaining the letters, three types of changes can be conducted: changing a letter's color, copying a certain letter to its surrounding position, and switching the letters' order. An example is shown in Figure 7. Given a source image, we make the P-Fidi system more *dynamic* by randomly selecting one among the three text change methods.

4.4.6. Facial Change. For the last change style, we first use a state-of-the-art face detector [Viola and Jones 2004] to localize the face region. Then a template-matching method is applied to align the face so that we can localize the eyebrow, eye, nose, mouth, and facial contour. Then we can proceed with several changes, such as enlarging (shrinking) the eyes or mouth or lengthening (shortening) the eyebrow. Even more, we can add some items to the face such as a mole or ink spot. These changes are subtle and difficult to recognize even they are on the face. Figure 8 shows an example.

4.5. Difficulty Level Setting

Game enjoyment can generally benefit from the intelligent difficulty level setting [Jin et al. 2013]. We observe that the difficulty level is very relevant to the selected change

Fig. 8. Some examples of facial changes. Source of image: http://www.flickr.com/photos/111945105@N07/11598354015/.

styles. We called in 10 persons to play the beta-version P-FiDi game and recorded the average rank h of each change style. h is originally in the range of $[1, 6]$ (higher rank means more difficult) and then linearly normalized within $[0, 1]$. From the results, we observe that the ranks for different styles are quite different. For example, the average rank of adding items is 2.37 and the average rank for deformation is 3.35, which shows that compared with deformation change, adding items is easier to identify. To control the difficulty level of a given image, we restrict that the average difficulty of all selected patches is within a certain range. Mathematically, we repeat the vector h for N times (N is the number of patches in the image) and obtain the long $(N \times K) \times 1$ vector ζ. To constrain the difficulty level of an image, we have

$$\theta_1 < s^T \zeta \leq \theta_2. \tag{5}$$

For easy, middle, and difficult levels, (θ_1, θ_1) is set as $(0, 1.2]$, $(1.2, 2.3]$, and $(2.3, 6)$, respectively.

4.6. Extensions Toward In-Game Advertising

In addition to entertainment, P-FiDi can also be a good carrier for advertisement. One desirable property of in-game advertisement is to be nonintrusive; that is, the players should almost not (if not completely not) perceive that there are advertisements. In this article, two solutions are provided for in-game advertising, namely, inserting commercial logos as small items and utilizing commercial posters as source images.

For the first solution, the commercial logos are inserted as small items into the source images. On the surface, it seems impossible to balance the benefits between players and advertisers. From the perspective of players, they expect to be not intruded upon, and thus logos cannot be too obvious. However, from the perspective of advertisers, logos must be obvious enough to be noticed by the players for better advertising results. P-FiDi perfectly resolves this situation. Logos are inserted by the techniques introduced in Section 4.4.3.

The second solution is to use advertisement posters as source images. The posters are often elaborately designed and contain many superstars to attract audiences' attention, which makes them ideal as source images. By playing the P-FiDi game, players check the posters carefully and are more likely to remember the brand. In this way, the effect of advertising is achieved.

5. EXPERIMENTS

In this section, we quantitatively and qualitatively evaluate the P-FiDi game. We first list some examples of each step output of the P-FiDi system to demonstrate its validity. The quantitative experiments are in the form of user studies. Four hundred images from two contributors (also participants of user studies) are used as the personalized photo

Fig. 9. Examples illustrating selected changeable patches by two methods. Left: the proposed selection strategy; Right: random selection strategy. Source of image: http://www.flickr.com/photos/112639541@N02/11585000716/.

albums. These daily photos record many memorable life moments, such as children's birthdays, friends' graduation ceremonies, friends' gatherings, and spring outings. In total, 30 friends (20 females and 10 males whose ages range from 10 to 50) of the two contributors are invited to participate in the user studies. We also download 30 advertisement posters from Flickr.com to evaluate the effectiveness of P-FiDi for in-gaming advertisements. The same groups of people are invited for the user study.

5.1. Qualitative Evaluations: Results Demonstration

In this subsection, we first evaluate the key components of P-FiDi step by step. Then, we quantitatively evaluate how P-FiDi performs when individual and multiple change styles are performed. Finally, the effectiveness of P-FiDi for in-game advertising is evaluated.

5.1.1. Patch & Change Style Selection. To validate the effectiveness of the proposed patch & change style selection strategy, we compare our selection method (Figure 9(a)) with a random baseline (Figure 9(b)). This baseline first randomly selects five patches, and then for each patch, one change style is randomly selected to be implemented. The results of two source images are shown in the top and bottom rows of Figure 9, respectively. For the upper image, the random baseline chooses nearby patches, for example, the sequential four patches (green bounding boxes). It is quite confusing whether the patches are four small differences or only one big difference. However, the selection strategy selects spatially evenly distributed patches.

5.1.2. Target Image Generation. Some results of the generated natural target images with desirable change styles are illustrated in Figure 10. More results, along with the front-end of P-FiDi[8] (based on Windows system), are available at http://sites.google.com/site/pfidifidi/personlized-fidi.

Figure 10 shows some results of P-FiDi on daily photos. The first row shows two indoor source–target image pairs. Note that the erasing technique is very effective in the indoor scene; for example, the elements in the bookcases are changed naturally. Outdoor results are shown in the second and third rows. In the second row, the text is detected and pasted to a nearby place. A bee and a butterfly are inserted into the sky (the upper region of the image), which demonstrates the importance of scene classification and location preference. The last row shows some results on several close-up images, that is, some daily necessities on a table and two flowers. The close-up images show that our algorithm can produce quite detailed changes. In all target images, the changed patches are spatially evenly distributed, which shows that the patch & style selection strategy works well.

[8]It is a simplified version for ease of distribution without the server end and the snap & play mode, and the users may play with the stored image pairs autogenerated by the full version of P-FiDi.

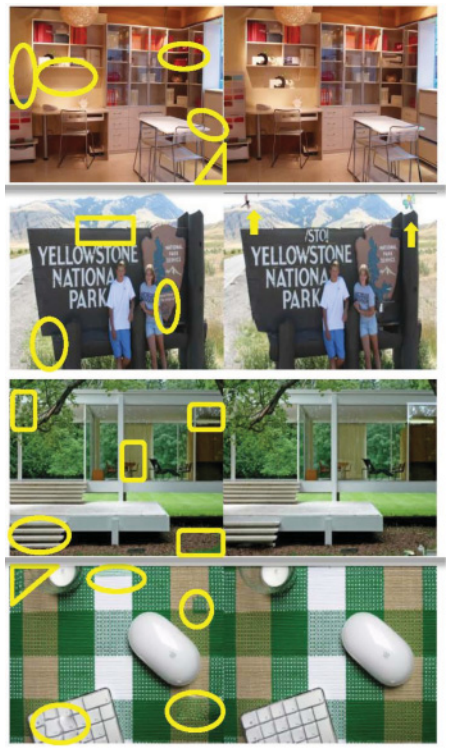

Fig. 10. Some results of P-FiDi on indoor, outdoor, and close-up images. Source of image: http://www.flickr.com/photos/111945105@N07/11598569613/, http://www.flickr.com/photos/111945105@N07/11599145396/, http://www.flickr.com/photos/111945105@N07/11599145106, http://www.flickr.com/photos/111945105@N07/11598361235/.

5.2. Quantitative Evaluations: User Studies

We systematically compare P-FiDi with traditional FiDi games from five aspects. Then the characteristics of P-FiDi are specially studied. Finally, P-FiDi's advertisement version is evaluated.

5.2.1. P-FiDi Versus Traditional FiDi. As the game interface design is not the focus of this work, we focus on evaluating the quality of the generated source–target image pairs. To generate the source–target image pairs of P-FiDi, we randomly select 15 images pairs from the personalized photo album and use them as source images. Their corresponding target images are automatically generated by P-FiDi, so the 15 are also sent to the 30 players. For comparison, we randomly download 15 source–target image pairs from the Tencent QQ website (the largest online gaming platform in China), denoted as QQ-FiDi, and another popular FiDi website,[9] denoted as Web-FiDi, respectively. The source–target image pairs from these two popular FiDi game providers are used as baselines and sent to the same 30 players. The order of P-FiDi, QQ-FiDi, and Web-FiDi is shuffled randomly for each player. Note that in the three versions of FiDi games, the same game interface is used. After players have played all the 45 image pairs (15 pairs for each FiDi version), they are asked to evaluate them from five aspects:

[9]http://spotthedifference.com/explorer.asp.

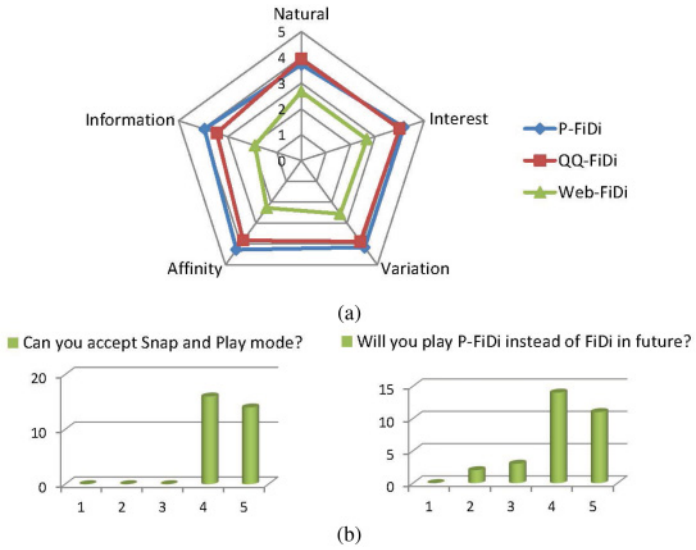

Fig. 11. (a) The user experience comparison between P-FiDi and two baselines from five aspects. (b) Evaluation of the characteristics of P-FiDi.

—Naturalness: Measure whether the generated target images have obvious forged artifacts.
—Interestingness: Assess whether playing the game is fun and helps players relax.
—Variation richness: Evaluate whether the change styles are diverse.
—Content affinity: Measure whether the game gives the players an familiar feeling.
—Information: Measure whether players can gain some information by playing the game.

All the evaluations are categorized into five levels of $\{1, 2, 3, 4, 5\}$, indicating "very bad," "bad," "average," "good," and "very good," respectively. The average statistics from the 30 participants are shown in Figure 11(a). From the results, P-FiDi overall outperforms the two baselines, and the superiority is obvious in terms of information and content affinity. Possible explanations include (1) players are friends of the photo contributors, and thus they may feel these source–target image pairs are very familiar and attractive, and (2) players can obtain much useful information by playing with commercial posters; for example, the players shall know when will the movie "Life of Pi" was released by playing with its movie poster. The naturalness of the P-FiDi system is comparable with, though slightly worse than, QQ-FiDi. It is reasonable since the target images from QQ-FiDi are all manually designed by experts.

The users are then required to give a final satisfactory comparison between P-FiDi, QQ-FiDi, and Web-FiDi by considering multiple criteria. The users are asked to give the comparison results using $\gg$, $>$, $=$, which mean "much better," "better," and "comparable." To quantify the results, we convert the results into ratings. We assign a score of 1 to the worst scheme, and the other schemes are assigned a score of 3, 2, or 1 if they are much better than, better than, or comparable to this one, respectively. Thus, for each comparison, there are 20 ratings. Since there will be disagreements among the evaluators, we perform a two-way analysis of variance (ANOVA) test [King and Minium 2003] to statistically analyze the comparison. The results are shown in Table I. We can see that P-FiDi has a higher satisfaction than both baselines in terms of mean scores. The p-values show that the difference of the different methods is significant and the difference of users is insignificant.

Table I. Two-Way ANOVA Test Results

			Factor of Different Schemes		Factor of Users	
	P-Fidi	QQ-FiDi/ Web-FiDi	F-Statistic	p-Value	F-Statistic	p-Value
P-Fidi vs QQ-FiDi	**2.10±0.45**	1.50±0.31	20.52	5.6×10^{-4}	0.30	0.9995
P-Fidi vs Web-FiDi	**2.70±0.31**	1.02±0.12	62.54	1.3×10^{-9}	0.50	0.9999

The left side of each column illustrates the mean and standard deviation of the three FiDi games. The right side illustrates the ANOVA test results in terms of F-statistic and p-value.

Fig. 12. The average playing time (seconds) for each estimated level of P-FiDi.

5.2.2. Characteristics of P-FiDi. After players finish the user study in Section 5.2.1, we tell them which one among the three versions is the P-FiDi game and then ask the players two more questions about P-FiDi:

(1) Can you accept the unique enjoyment of the Snap & Play game mode?
(2) Will you play the personalized and dynamic P-FiDi instead of traditional FiDi games in the future?

The statistics of the results are shown in Figure 11(b). Again, the scores are within $\{1, 2, \ldots, 5\}$, and a larger value means more preference. From the figure, we can conclude that players can accept the Snap & Play game mode, where photos instantly captured can be used for playing. Most participants are willing to play with P-FiDi instead of traditional FiDi games.

The results on difficulty level setting are shown in Figure 12. We can see that the average gaming times for the estimated "easy," "middle," and "difficult" levels are 16.5s, 23.1s and 31.9s, respectively. It well demonstrates the effectiveness of the difficulty level stetting strategy since a more difficult level needs a longer playing time.

5.2.3. P-FiDi Towards In-Game Advertising. In this article, we will sequentially evaluate the effectiveness of two P-FiDi-related advertising styles: inserting logos or using advertisement posters as source images.

For logo-related advertisement, a naive baseline is direct showing the subjects' images containing logos. We denote the baseline as browsing. We modify the aforementioned advertisement style by incorporating the original images (source images) and the logo-inserted images (target images) into the P-FiDi game. We denote our game-based advertisement style as P-FiDi. Next, we compare the advertisement effects between the browsing and P-FiDi styles. For P-FiDi, we randomly select 10 source images and 10 logos. The logos are inserted into the source images to generate 10 target images. Thus, the 10 source–target image pairs are used as game materials for P-FiDi. To compare the advertising effects of browsing and P-FiDi, we divide all subjects into group A and group B, with 15 members each. Subjects in group A are asked to browse the 10 target images freely. Subjects in group B are required to play with P-FiDi. Each member is assigned 10 source–target image pairs with logo inserted. For both groups,

Table II. The Results on In-Game Advertising

	Recall Rate		Play Time(s)	
Evaluation	Browse	Play	Browse	Play
Insert a Logo	0.40	**0.89**	5.6	**6.2**
poster content 1	0.66	**0.81**	4.79	**21.76**
poster content 2	0.32	**0.73**		

the average staying time for each image is recorded. After all subjects in both groups finish browsing or P-FiDi gaming, they are asked to answer a multiple-choice question, which is designed as follows. We show all players 20 logos (including the 10 appeared logos and 10 unseen logos) and ask players to select 10 logos that they have played with or browsed. Then, recall rate is measured as the percentage of correct selections. The first row of Table II shows that the recall rate of the P-FiDi group is 0.89, which significantly outperforms the 0.40 from the browsing group. We also report the time spent to play the P-FiDi game and browse, and the results show that the members playing P-FiDi spent a much longer time to check the images. In all, we can conclude that P-FiDi can significantly improve the advertisement effects by attracting players for longer time and leaving deeper impressions.

To validate the effect of using advertisement posters as source images, we again randomly divide all subjects into groups A and B. Members of group A play the P-FiDi game with 10 randomly assigned source–target image pairs, and members of group B are assigned the same 10 source images, namely, advertisement posters, for browsing. After they finish, two tests are performed. In the first test, we investigate users' rough understanding of the posters' contents with a multiple-choice question. We provide 20 brand names and ask players to identify whose posters have been played with or browsed. The recall rates are calculated. The results are shown in the second row of Table II. We can conclude that members playing the P-FiDi game obtain a higher recall of 0.81 than 0.66 for browsing users. The second test is to evaluate users' deep understanding about posters' details. In total, 10 single-choice questions (one question for one poster) are asked; for example What is the color of the iPhone in the Apple poster? The results in the third row of Table II show that the subjects who played P-FiDi obtain a better recall rate of 0.73 than browsing subjects of 0.32. These two results show that P-FiDi forces the players to understand both the overall and detailed content in the poster. Again, the average staying time for one image is recorded, and we find that compared with browsing, P-FiDi players spend more time on the posters, which can bring in potential business.

5.3. Discussions

Technical Feasibility: Currently, almost all computational cost is on the server end. The server end of P-FiDi takes about 10 seconds on average to generate a target image of size 500×375 pixels based on unoptimized C-code on an XeonX5450 workstation with 3GHz CPU and 16GB memory.

Failure Cases: It is predictable that some unsatisfactory cases may exist. For example, an item of sheep may be added onto a river, which is against common sense, but fortunately, mostly these unsatisfactory cases only lower the difficulty level of the game and only slightly affect the naturalness of the target image, but do not result in any fatal consequence. Sometimes, these occasional results may even bring fun to the players.

6. FUTURE WORK

In this article, we explored how state-of-the-art image processing techniques can be intelligently combined to develop an automatic, personalized P-FiDi game. The new

electronic game is characterized with (1) no requirement of experts to design gaming image pairs, which is labor-intensive and tedious; (2) the new player experience of "I am the owner of the gamer"; (3) the intelligence in dynamic and personalized aspects; and (4) a unique visual experience. Beyond being a new game, P-FiDi can be slightly redesigned to be an excellent advertisement carrier. The game design methodologies proposed in this work are expected to be generalized to other related popular games for enhancing players' gaming experience.

ACKNOWLEDGMENT

This work was supported by the Singapore National Research Foundation under its International Research Centre @Singapore Funding Initiative and administered by the IDM Programme Office, National Natural Science Foundation of China (No. 61332012), National Basic Research Program of China (2013CB329305), National High-tech R&D Program of China (2014BAK11B03), and 100 Talents Programme of The Chinese Academy of Sciences.

REFERENCES

Subhabrata Bhattacahrya, Rahul Sukthankar, and Mubarak Shah. 2010. A framework for photo-quality assessment and enhancement based on visual aesthetics. In *MM*. 271–280.

Tony Chan and Jianhong Shen. 2000. Non-texture inpainting by curvature-driven diffusions. In *JVCIR* 12 (2000), 436–449.

Daniel Cohen-Or, Olga Sorkine, Ran Gal, Tommer Leyvand, and Ying-Qing Xu. 2006. Color harmonization. *TOG* 25, 3 (2006), 624–630.

Dorin Comaniciu and Peter Meer. 2002. Mean shift: A robust approach toward feature space analysis. *IEEE TPAMI* 24 (2002), 603–619.

Antonio Criminisi, Patrick Perez, and Kentaro Toyama. 2003. Object removal by exemplar-based inpainting. In *CVPR*. 721–728.

Boris Epshtein, Eyal Ofek, and Yonatan Wexler. 2010. Detecting text in natural scenes with stroke width transform. In *CVPR*. 2963–2970.

Tom Grill and Mark Scanlon. 1990. *Photographic Composition*. Amphoto Books.

Xiaodi Hou and Liqing Zhang. 2007. Saliency detection: A spectral residual approach. In *CVPR*. 1–8.

Jung-Hwan Jin, Hyun Joon Shin, and Jung-Ju Choi. 2013. SPOID: A system to produce spot-the-difference puzzle images with difficulty. *Visual Computer* 29 (2013), 1–9.

Bruce King and Edward Minium. 2003. *Statistical Reasoning in Psychology and Education*. John Wiley.

Svetlana Lazebnik, Cordelia Schmid, and Jean Ponce. 2006. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *CVPR*. 2169–2178.

Anat Levin, Dani Lischinski, and Yair Weiss. 2006. A closed form solution to natural image matting. In *CVPR*. 61–68.

Ligang Liu, Renjie Chen, Lior Wolf, and Daniel Cohen-Or. 2010. Optimizing photo composition. In *Computer Graphics Forum*, Vol. 29. 469–478.

Si Liu, Qiang Chen, Jian Dong, Shuicheng Yan, Changsheng Xu, and Hanqing Lu. 2011. Snap & play: Auto-generate personalized find-the-difference mobile game. In *MM*. 993–996.

Tao Mei, Xian-Sheng Hua, and Shipeng Li. 2008. Contextual in-image advertising. In *MM*. 439–448.

Aude Oliva and Antonio Torralba. 2001. Modeling the shape of the scene: A holistic representation of the spatial envelope. *IJCV* 42, 3 (2001), 145–175.

Carl Olsson. 2007. Solving large scale binary quadratic problems: Spectral methods vs. semidefinite programming. In *CVPR*. 1–8.

Patrick Pérez, Michel Gangnet, and Andrew Blake. 2003. Poisson image editing. In *SIGGRAPH*. 313–318.

Erik Reinhard, Michael Ashikhmin, Bruce Gooch, and Peter Shirley. 2001. Color transfer between images. *IEEE Comput. Graph. Appl.* 21, 5 (2001), 34–41.

Paul Viola and Michael J. Jones. 2004. Robust real-time face detection. *IJCV* 57, 2 (2004), 137–154.

Che-Hua Yeh, Yuan-Chen Ho, Brian A. Barsky, and Ming Ouhyoung. 2010. Personalized photograph ranking and selection system. In *MM*. 211–220.

Received April 2013; revised September 2013; accepted November 2013