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

Si Liu*, National University of Singapore
Qiang Chen*, National University of Singapore
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
*indicates equal contribution.

In this paper, by taking the popular game, Find-the-Difference (FiDi) game, as a concrete example, we explore how state-of-the-art image processing techniques can assist in developing personalized, automatic, and dynamic game. Unlike traditional FiDi game, where image pairs (source image & target image) with 5 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 P-FiDi system to auto-generate the counterpart target images for users to play. Four steps are adopted to auto-generate 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; generating the differences on the target image with state-of-the-art image processing techniques. In addition, P-FiDi game can be easily re-designed for the im-game advertising. Extensive experiments show that 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 Personlized System, Image Analysis, Human Computer Interaction, Image Editing, Image Modeling

1. INTRODUCTION

The Entertainment Software Association\(^1\) has reported that more than 200 million hours are spent each day playing electronic game (e.g. computer, video games) in USA. The huge market for electronic games motivates us to think about one problem: Can state-of-the-art image processing techniques help develop automatic personalized games?\(^2\)

In this paper, by taking a popular game, Find-the-Difference (FiDi) game, as a concrete example, we explore the merits brought by the increasingly mature image processing techniques for game design. FiDi game, as a typical representative of puzzle/logic game\(^3\), is very popular. Statistics show that in the QQ Game\(^4\), the biggest online game society in China, averagely about 180,000 players are simultaneously online playing FiDi game at 8:00 PM from Mar. 1\(^{st}\) to Mar. 31\(^{st}\) 2013. By contrast, the online number is 20,387 for Texas hold ’em Poker, which is one of the most popular forms of poker\(^5\).

A typical interface of FiDi is shown in Figure 1. Two images are displayed in pair. The left image is called source image, while the right one is called target image. Source image and target image are almost the same except for several (generally 5) small different patches generated by Photoshop or other techniques. The players are required to identify all differences within the limited time.

The success of traditional FiDi game is largely due to 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, FiDi game is imperfect in the following aspects. First, the target images are manually generated by professional image processing experts using professional image processing softwares,
such as Photoshop. It is very time-consuming and tedious. It even becomes infeasible for 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 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, as shown in Figure 2. Player first snap something by one’s mobile phone, the instantly captured photo is used as the source image in P-FiDi game. The auto-generated target image, pairing with the source image, together serve as game materials to play. P-FiDi has similar interface and game rules with FiDi, shown in Figure 1.

P-FiDi should inherit the aforementioned nice characters of traditional FiDi game. Naturalness is the most important principle when designing P-FiDi game. Differences should naturally and seamlessly fuse with other patches in the image. Besides, the imperfectness of traditional FiDi game should be tackled via state-of-the-art image processing techniques. P-FiDi game has several unique characters: 1) to alleviate the image editing experts from the tedious manual editing, P-FiDi system is automatic. The whole auto-generation process includes automatic changeable patches generation, automatic patch & change style selection, and target image auto-generation. 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, target image is auto-generated by making use of many automatic image editing algorithms, such as Poisson editing [Perez et al. 2003]. 2) P-FiDi is a personalized system, where users can play with one’s own photos. However, in the traditional FiDi game, players have to play with images provided by the game developers. Personalization can arouse players’ interest and affinity feeling for the game 3) Finally, to avoid experienced players cheating and the possible boring sense of the game in traditional FiDi game, 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.

Fig. 1: The interface for the basic version of P-FiDi. For better viewing, please see the original color pdf file for all images.
Fig. 2: Snap & Play gaming mode for Personalized FiDi (P-FiDi) game, where the instantly captured photo is used as source image, and the system auto-generates the counterpart target image with 5 different patches.

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 early version of this paper is published in [Liu et al. 2011]. The main differences between these two versions are summarized as follows. Firstly, an extension towards in-game advertisement is added, which provides the game the extra possibilities to make profit. Secondly, qualitative results for each change style as well as combination of all change styles of the P-FiDi system are presented in experiments section. Thirdly, qualitative results to compare our patch & change style selection strategy and a random baseline are shown in 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 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: FiDi game belongs to one important computer games gene, i.e., puzzle/logic game. FiDi can help players to develop the capability of observation, relax mood and simultaneously browse many high-quality photos. FiDi games are often found in children’s puzzle books, and in newspapers. Recently, FiDi features in computer, online and mobile games5. 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.

Image Processing Techniques: To automatically generate target images in P-FiDi, several state-of-the-art techniques are utilized. There are a lot of literatures to enhance the visual experience [Bhattacharya et al. 2010; Yeh et al. 2010; Reinhard et al. 2001]. Some work [Cohen-Or et al. 2006; Reinhard et al. 2001] focus on how to change the color of foreground to match the color of background. Poisson Editing [Pérez et al. 2003] serves as a commonly used technique to insert items into images. Due to the 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 semantic 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 off-the-shelf optimization package.

5http://www.theesa.com/facts/
Fig. 3: The framework for the P-FiDi game. The instantly captured image (a) is used as source image. It is visually enhanced whenever necessary (b). Three kinds of changeable patches (over-segmented 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).

**In-game Advertising:** As the computer game playing population expands, in-game advertisements are expanding as well. Massive Incorporated\(^6\) is a creator of dynamic game advertisements, estimates the in-game advertising market could grow to 1 billion globally by 2014. P-FiDi can be straightforward extended to in-game advertisements. P-FiDi based advertisement are non-intrusive and interesting compared with its counterparts [Mei et al. 2008] in web image advertisement. Besides in-game advertisement, game can be designed as a carrier for other purpose. For example, Andreea Molnar et.al. [Molnar and Kostkova] explore how game can assist education.

3. SYSTEM OVERVIEW

3.1. P-FiDi Interface and Rules

A typical interface is shown in Figure 1. Two images, i.e. one source image and one target image, lie in the central area. The source images is captured by one’s mobile devices, or downloaded from user’s social network account. The target image is generated by P-FiDi system. Players should identify all the 5 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 getting stuck, one may seek for 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 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 is popped up after the end of the game.

\(^6\)http://www.massiveincorporated.com/
Fig. 4: Exemplar results on photo-quality aesthetic enhancement on two different types of images. (a) For each image pair, the original images are displayed in the left while the aesthetically enhanced images are placed in the right. (b) Original image, reference image from the collected professional Flickr image set and aesthetically enhanced images are shown sequentially.

3.2. P-FiDi Technical Flow
The technical flowchart of P-FiDi system is illustrated in Figure 3. First, one player takes a picture as shown in Figure 3(a). Then, if necessary, the source image may be aesthetically enhanced for better visual experience, as shown in Figure 3(b). After that, the photo is segmented into small changeable patches, typically 30-50 patches like Figure 3(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 to \( M \) (5 in this work) patches are selected to change and their corresponding change styles are determined simultaneously. In Figure 3(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 3(e).

4. TECHNIQUE DETAILS
P-FiDi system contains four key components, i.e. 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 player’s adhesion is the pleasant gaming experience. However, most players are not professional photographer, 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 mean-shift algorithm [Comaniciu and Meer 2002]. We also calculate the saliency value of each pixel by 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, i.e., \( p_1/p_2 \geq h \), then the top ranked patch is considered as the only foreground object inside the image. \( h \) is set as 1.25 in 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. Some example result images are shown in Figure 4a(1) and Figure 4a(2). The positions of the main object, i.e., the cow and the house are moved to the one third positions. It is obvious that the new images are better proportioned. The second method is to increase the brightness around the main object, so that the generated target image has stage effects. One example is shown in Figure 4a(3). The bulb becomes brighter and the target image has artistic effects. The last method is to blur the background, which indirectly highlights the foreground object. Figure 4a(4) shows an example.

**Scene Images without Definite Foreground:** First, we evaluate the quality of scene image to decide whether the visual experience enhancement step is required by 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 5000 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 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 user’s photo. One example is shown Figure 4b. The generated new image is more attractive to players.

4.2. Changeable Patches Generation

Three kinds of changeable patches, i.e., over-segmented patches, facial patches and text patches, are considered in P-FiDi game. Mean-shift algorithm [Comaniciu and Meer 2002] is adopted to produce several over-segmented patches which are composed of several visually similar and spatially adjacent pixels. Besides the over-segmented patches, two other kinds of semantic patches, i.e., 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, nose, and 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 formu-
luation of the selection process, and then introduce the definition of a very important component of the formulation, i.e., 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 non-zero 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 style should suit for 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 with $S$, 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. 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:

$$\max_s c^T s. \quad (1)$$

**Objective II: Spatially Even Distribution** Intuitively, distribute 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 ones and others are zeros. 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:

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

Four constraints are also imposed over $s$ to achieve reasonable the 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 paper, $M$ 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, $(Rs)_i \leq 1, \forall i \in \{1, 2, \cdots, N\}$.

**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 one and others are zeros. 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 $(Ts)_j \leq 2, \forall j \in \{1, 2, \cdots, K\}$.
Fig. 5: Several examples of color change. (a) using color transfer method. Some windows colors are changed from black to brown, because the nearby patches are all brown. (b) using color harmony method. The originally blue vest has changed to its contrast color, i.e., yellow. In this paper, each changed patch is marked by rectangular, eclipse or arrow. For better viewing, please see the original color pdf file for all images.

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

\[
\begin{align*}
    \text{max} & \quad c^T s + \frac{1}{2M} s^T R^T D R s \\
    \text{s.t.} & \quad (R_s)_i \leq 1, \forall i \in \{1, 2, \cdots, N\} \\
    & \quad (T_s)_j \leq 2, \forall j \in \{1, 2, \cdots, K\} \\
    & \quad \sum_{i=1}^{N \times K} s_i = M; \quad s \in \{0, 1\}^{(N \times K) \times 1}.
\end{align*}
\]

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, e.g., Mosek\(^7\), relax this problem either using Semi Definite Programming or spectral relaxation [Olsson 2007].

**Implementation Details:** In practical, the number of variables is \(N \times K\) in Eq. (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 5 patches with the highest fitness score (calculated by the method introduced in Sec. 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 patch is more natural and easier to implement than into complicated patches. Below 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 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 \(\ell_1\) 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

\(^7\)http://www.mosek.com/
Fig. 6: Some examples of deformation over the selected patches. (a) The food inside the bowl inflates. (b) The original straight table edge becomes distorted.

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,

\[
c_j = w_{j1}f_{sz} + w_{j2}f_{sp} + w_{j3}f_{ct} + w_{j4}f_{cv} + q,
\]

where \( w_{j1}, w_{j2}, w_{j3}, w_{j4} \) and \( q \) 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 Eq. 4), which is then normalized to \([0, 1]\). Summing up 6 change styles, we have totally 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 different patch & change style fitness matrix \( C \), which further causes different patch & change style selection vector \( s \) via Eq. 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.

4.4. Target Image Auto-generation

Based on the selected patches with their corresponding change styles determined in Section 4.3, in this subsection, we elaborate on auto-generating 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. Human perception system is very sensitive to color change. In 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 to be more dynamic.

**Color Transfer**: The basic idea is to transfer the colors from surrounding patch to the selected patch in \( l^\alpha\beta \) color space [Reinhard et al. 2001]. An example is shown in Figure 5(a).

**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. Figure 5(b) shows one example of color harmony.

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-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]. Two examples are shown in Figure 6.

4.4.3. Adding. The third change style is adding certain stuff into the image. We propose two methods to implement it. 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.

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 (PDE) [Chan and Shen 2000]. The basic idea of example based method is to copy the surrounding patch to fill the hole. The results are more natural but it takes higher computation cost. The PDE-based method generates smooth interpolation from the outline into the hole with 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

---

**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 shown in Figure 7, including animals, daily tools, vegetables, etc. We further manually classifier all items into nine groups based on two criterion, i.e., 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 chairs often appear at the bottom part of an image, while clocks are more likely to be in the upper part of an image.

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 3 categories, i.e., indoor, city and country-side). We also check the location of the selected changeable patch. Then, only the items satisfying both scene and location preference are kept and form a candidate set. Among the candidate set, top 10 items most similar to the selected changeable patch are selected. Then only one patch among top 10 set is randomly selected, which serves as another strategy to enhance the dynamic characteristic of P-FiDi system. One example is shown in Figure 8(a).

**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 patch can achieve surprisingly good results sometimes, such as the cloud in Figure 8(b).

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 (PDE) [Chan and Shen 2000]. The basic idea of example based method is to copy the surrounding patch to fill the hole. The results are more natural but it takes higher computation cost. The PDE-based method generates smooth interpolation from the outline into the hole with 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
Fig. 8: Some examples of adding items in the selected patches. (a) Adding small cartoon items, such as boat and car into outdoor images. (b) Repeating existing patches into the photo: the cloud is copied to its neighborhood.

Fig. 9: Some examples of erasing items/patches. (a) Using PDE-based method: the deleted sign is filled by nearby the rocks on the wall. (b) Using interpolation method: the color inside the rectangular is interpolated by the light blue boundary.

Fig. 10: Some examples of text change. (a) Changing letters’ color. (b) Switching the letters’ order and (c) pasting a word to surrounding patches.

of surrounding region is larger than a pre-set threshold, we use example method; otherwise we use PDE-based method. Some examples are shown in Figure 9.

4.4.5. Text Change. The fifth change style is related with texts. 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 towards 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, i.e., changing letter’s color, copying certain letter to its surrounding position and switching the letters’ order. Some examples are shown in Figure 10. 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 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 several changes such as enlarging (shrinking) the eyes or mouth, lengthening (shortening) the eyebrow. Even more, we can add some items to the face such as mole or ink spot. These changes are subtle and difficult to recognize even it is on the face. Figure 11 shows some examples.

4.5. Difficulty Level Setting

The 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 styles. We call in 10 persons to play the beta-version P-FiDi game and record 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 comparing 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 $\zeta$. 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 level, $(\theta_1, \theta_1)$ is set as $(0, 1.2], (1.2, 2.3]$ and $(2.3, 6)$, respectively.

4.6. Extensions towards 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, i.e., the players should almost not (if not completely) perceive that they are being advertised. In this paper, 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, 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 embarrassing situation. Logos are inserted by the techniques introduced in Section 4.4.3. We show two examples in Figure 12. From the results, we can see that the logos of “Facebook” and “HP” are seamlessly inserted into the images and the overall aesthetic feeling of the images have not been spoiled, which meets the demands of the players not to be interrupted. On the other side, in order to find the differences, players need carefully check the source-target image pairs until finding the inserted logos. In this way, the logos will leave a deep impression on the players.
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 P-FiDi game, players check carefully of the posters, such as the “Coca-Cola” poster in Figure 13, 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 P-FiDi game. We first list some examples of each step output of P-FiDi system to demonstrate its validness. The quantitative experiments are in the form of user studies. 400 images from two contributors (also participants of user studies) are used as the personalized photo albums. These daily photos record many memorable life moments, such as children’s birthdays, friends’ graduation ceremonies, friends gathering, spring outing. In total, 30 friends (20 females and 10 males whose ages vary from 10 to 50) of the two contributors are invited to participate the user-studies. We also download 30 advertisement posters from Flickr.com to evaluate the effectiveness of P-FiDi for in-gaming advertisement. The same groups of people are invited for 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 14(a)) with a random baseline (Figure 14(b)). This baseline first randomly selects 5 patches, then for each patch, one change style is randomly selected to be implemented. The results of two source images are shown in up and bottom rows of Figure 14, respectively. For the upper image, the random baseline chooses nearby patches, for example the sequential 4 patches (green bounding boxes). It is quite confusing whether the patches are four small differences or only one big difference. However, selection strategy selects spatially evenly distributed patches. For the bottom image, random strategy chooses a very large patch (blue bounding box) in the sky area. Identifying this difference is too easily and the challenge of the game is reduced. Random strategy also selects a very small patch (white bounding box), which is almost invisible. In this situation, players’ sense of frustration might be aroused. To the contrary, selection produces appropriate-sized patches. To sum up, the selection procedure can greatly improve gameplay experience.

5.1.2. Target Image Generation. Some results of the generated natural target images with desirable change styles are illustrated in Figure 15, 16, 17. More results along with the front-end of P-FiDi\textsuperscript{8} (based on Windows System) are available at http://sites.google.com/site/pfidifidi/personlized-fidi.

Individual Change Style: Figure 15 shows some results of P-FiDi on daily photos when one single change style is implemented. The color of cup and slide are changed very naturally in the first row. Deformation change is almost unnoticeable, such as the distorted boards and cupboard in the second row. We mainly introduce two methods for adding items. Copying repetitive pattern is more natural, for example, the building part, while adding small items, such as the dragonfly and deer, is usually easy to perceive. The fourth row demonstrates some results of deleting. The house is successfully removed, while the area behind dog area is blurred due to the adopted interpolation method. The next difference style is text change. In real applications, most texts always appear in

\footnotesize\textsuperscript{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 auto-generated by the full version of P-FiDi.
Fig. 15: Examples of P-FiDi on daily images are shown for each change style. We consider six change styles: color, deform, add, delete, text and facial change.

The form of brand name. For example, the color of letter ‘E’ is changed in the “Starbucks Coffee” sign, while the letter sequences of ‘a’ and ‘o’ have be switched in the “Mcdonalds” sign. Facial change is generally difficult. In the last row, the girl’s mouth and the woman’s eyes are enlarged. Overall we can conclude P-FiDi can produce satisfactory results for different change styles.

Combination of Multiple Change Styles: Figure 16 shows some results of P-FiDi on daily photos. The first row shows two indoor source-target image pairs. Note that erasing technique is very effective in indoor scene, for example the stuff in bookcases are changed naturally. Outdoor results are shown in the second and third rows. In the left image of the second row, the text is detected and pasted to nearby place. A bee and a butterfly are inserted into the sky (the upper region
Fig. 16: Some results of P-FiDi on indoor, outdoor and close up images.

of the image), which demonstrates the importance of scene classification and location preference. The last row shows some results on several close-up images, i.e. some daily necessities on a table and two flowers. The success in close-up images shows 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.

**Extensions Towards In-game Advertising:** The results of P-FiDi’s in-game advertising extensions are shown in Figure 17. The first two rows show the results of inserting logos into images. From these results, we can conclude that logos are cleverly hidden into regions of similar color or texture. The last row demonstrates the results of using posters as source images. We believe that by playing P-FiDi game, players pay more attention to every detail in the poster, which is beneficial for advertising effectiveness.

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 Vs. 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 Tencent QQ website (the largest online gaming platform in China), denoted
as QQ-FiDi, and another popular FiDi website\textsuperscript{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:

— Naturalness: Measure whether the generated target images have obvious forged artifacts.
— Interestingness: Assess whether playing the game is fun and help players relaxed.
— Variation richness: Evaluate whether the change styles are diverse.
— Content affinity: Measure whether the game deliver the players an intimacy 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 18a. 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 the friends of the photo contributors, and thus they may feel these source-target image pairs very familiar and attractive, and 2) players can obtain much useful information by playing with commercial posters, e.g., the players shall know when will the movie “Life of Pi” release by playing with its movie poster. The naturalness of 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

\textsuperscript{9}http://spotthedifference.com/explorer.asp
Fig. 18: (a) The user experience comparison between P-FiDi and two baselines from five aspects. (b) Evaluation of the characteristics of P-FiDi.

Table I: Two-way ANOVA test results: 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.

<table>
<thead>
<tr>
<th></th>
<th>P-FiDi</th>
<th>QQ-FiDi/Web-FiDi</th>
<th>the factor of different schemes</th>
<th>the factor of users</th>
</tr>
</thead>
<tbody>
<tr>
<td>P-FiDi vs QQ-FiDi</td>
<td>2.10±0.45</td>
<td>1.50±0.31</td>
<td>20.52</td>
<td>0.30</td>
</tr>
<tr>
<td>P-FiDi vs Web-FiDi</td>
<td>2.70±0.31</td>
<td>1.02±0.12</td>
<td>62.54</td>
<td>0.50</td>
</tr>
</tbody>
</table>

are assigned a score of 3, 2 or 1 if it is 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 method is significant and the difference of users is insignificant.

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 P-FiDi game and then ask the players two more questions about P-FiDi:

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

The statistics of the results are shown in Figure 18(b). Again, the scores are within \{1, 2, ..., 5\}, and 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 19. We can see that the average gaming time 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 setting strategy since more difficult level needs longer playing time.

![Average play time for each difficulty level](image)

Fig. 19: The average playing time (seconds) for each estimated level of P-FiDi.

<table>
<thead>
<tr>
<th>Evaluation</th>
<th>Recall Rate</th>
<th>Play Time(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Insert a Logo</td>
<td></td>
<td></td>
</tr>
<tr>
<td>poster content 1</td>
<td>0.66</td>
<td>4.79</td>
</tr>
<tr>
<td>poster content 2</td>
<td>0.32</td>
<td>21.76</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Evaluation</th>
<th>Recall Rate</th>
<th>Play Time(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Insert a Logo</td>
<td></td>
<td></td>
</tr>
<tr>
<td>poster content 1</td>
<td>0.66</td>
<td>4.79</td>
</tr>
<tr>
<td>poster content 2</td>
<td>0.32</td>
<td>21.76</td>
</tr>
</tbody>
</table>

5.2.3. P-FiDi Towards In-game Advertising. In this paper, 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 above-mentioned 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, 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 multi-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 which 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 P-FiDi group is 0.89, which significantly outperforms the 0.40 from the browsing group. We also report the time spent to play P-FiDi game and browse, and the results show that the members playing P-FiDi spent much longer time to check the images. In all, we can conclude that P-FiDi can significantly improves the advertisement effects by attracting players longer time and leaving deeper impressions.

To validate the effect of using advertisement posters as source images, we again randomly divide all subjects into group A and B. Members of group A play 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 multi-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 P-FiDi game have obtained a higher recall of 0.81 than 0.66 of 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, e.g. what’s the color of iPhone in the Apple poster. The results in the third row of Table II show the subjects who played P-FiDi obtain better recall rate of 0.73 than browsing subjects of 0.32. These two results show that P-FiDi force 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 comparing with browsing, P-FiDi players spend more time on the posters, which can bring in potential business value.

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 averagely to generate a target image of size $500 \times 375$ pixels based on un-optimized 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-down the difficulty level of the game, and slightly affect the naturalness of the target image, yet shall not result in any fatal consequence. Sometimes, these occasional results may even bring fun to the players.

6. Future Work
In this paper, 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) 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 the players’ gaming experience.

7. Acknowledgment
This research is supported by the Singapore National Research Foundation under its International Research Centre @Singapore Funding Initiative and administered by the IDM Programme Office.

REFERENCES
Tony F. Chan and Jianhong Shen. 2000. Non-Texture Inpainting by Curvature-Driven Diffusions. In JVCIR.
B. King and E. Minium. 2003. Statistical reasoning in psychology and education.
Andreea Molnar and Patty Kostkova. Seamless Evaluation Integration into IDS Educational Games. (????).
Che-Hua Yeh, Yuan-Chen Ho, Brian A. Barsky, and Ming Ouhyoung. 2010. Personalized photograph ranking and selection system. In MM. 211–220.