Multimed Tools Appl (2016) 75:16861–16880
DOI 10.1007/s11042-015-2950-5

Extra detail addition based on existing texture for animated news production

Yu-Chi Lai[1] · Chun-Wei Wang[2] · Kuang-Yi Chen[2] · Dobromir Todorov[3] · Chih-Yuan Yao[1]

Received: 10 February 2015 / Revised: 4 September 2015 / Accepted: 14 September 2015 /
Published online: 1 October 2015

Abstract Animated news proposed by Next Media Animation becomes more and more popular because an animation is adapted to tell the story in a piece of news which may miss visual and audial circumstances. In order to fulfill the requirement of creating a 90-second animation within 2 h, artists must quickly set up required news elements by selecting existing 3D objects from their graphics asset database and adding distinguished details such as tattoos, scars, and textural patterns onto selected objects. Therefore, the detail addition process is necessary and must be easy to use, efficient, and robust without any modification to the production pipeline and without addition of extra rendering pass to the rendering pipeline. This work aims at stitching extra details onto existing textures with a cube-based interface using the well-designed texture coordinates. A texture cube of the detail texture is first created for artists to manipulate and adjust stitching properties. Then, the corresponding transformation between the original and detail textures is automatically computed to decompose the. detail texture into triangular patches for composition with the original texture. Finally, a complete detail-added object texture is created for shading. The designed algo-

Electronic supplementary material The online version of this article (doi:10.1007/s11042-015-2950-5) contains supplementary material, which is available to authorized users.

✉ Chih-Yuan Yao
cyuan.yao@gmail.com

Yu-Chi Lai
cheeryuchi@gmail.com

1 National Taiwan University of Science and Technology, No.43, Sec. 4, Keelung Rd., Da'an Dist., Taipei 106, Taiwan (R.O.C.)

2 National Taiwan University of Science and Technology and International Games System, No.43, Sec. 4, Keelung Rd., Da'an Dist., Taipei 106, Taiwan (R.O.C.)

3 National Taiwan University of Science and Technology and Next Media Animation, No.43, Sec. 4, Keelung Rd., Da'an Dist., Taipei 106, Taiwan (R.O.C.)

rithm has been integrated into the Next Media Animation pipeline to accelerate animation production and the results are satisfactory

Keywords Texture mapping · Detail stitching · Animation production · Paramterization

1 Introduction

News in newspapers, TV, and Internet is an important source of information for modern people, but textual or oral description in news requires imagination to comprehend the content. With the advance of technology, hand-held cameras become popular and available for general households, and news events are captured by these cameras by accident. These captured videos can visually and audially enhance the comprehension of the news and deliver "being-there" reality to audiences. The most famous example is when the second plane crashed into the World Trade Center, the entire event was captured by a tourist and broadcasted live to an audience. Although event videos become important material in news, only a few events can be captured by accident. For example, there is no even video capturing the death of the North Korean leader, Jong-Il Kim, in 2011 or the car accident of Tiger Woods in 2009. Furthermore, some events are captured incompletely. For example, the 911 event videos missed the first plane crashing into World Trade Center. This lack of event videos or incompleteness in event videos must be compensated with other means for audiences' comprehension. To make an audience easily comprehend news contents with compensation of the missing visual information gives birth of the concept of animated news created by piecing virtual 3D animations with incomplete news sequences or totally using virtual 3D animations. As one of the largest animation production companies in Asia, Next Media Animation(NMA) uses the computer graphics techniques to completely and comprehensively master the animation production pipeline for generation of high-quality 3D animations in high efficiency for providing as many interactive entertainment and news contents in as short periods as possible. Therefore, NMA uses 3D animation techniques to tell the story in a piece of news for satisfaction of audiences' visual and audial need. Although 3D animation production becomes more and more important in the Hollywood movie industry since Pixar published Luxo Jr., the production of a 2-h animation still requires months of work and a large amount of human resource. This limits the possibility of using a traditional animation pipeline to create animated news because broadcasting a piece of breaking news competes with time, and the headline must be online in minutes. As a result, NMA incorporates a game engine into a traditional animation software for construction of an animated news production pipeline which is capable of transforming a piece of news into a 90-second animation within 2 h. The rendered animation can be broadcasted together with available event videos to provide enough visual and audial circumstances for easy comprehension. During production, modeling and animating is time consuming and can be accelerated with a pre-constructed graphics asset database of commonly used 3D models, characters, scenes, motions, and animation templates. For example, the database should contain models and motions of Tiger Woods and Barack Obama due to their appearance possibilities in different events. However, the database may not totally fulfill the descriptions of objects appearing in the news. For example, a cloth appearing in breaking news has a distinct tiger pattern as shown in Fig. 1b, but the database only has a similar cloth model in Fig. 1a. Another example is that a female figure in the news has a tiger tattoo on her back as shown in Fig. 1e, but the database does not have a model with that tattoo as shown in Fig. 1d. Both examples require addition of extra details onto the

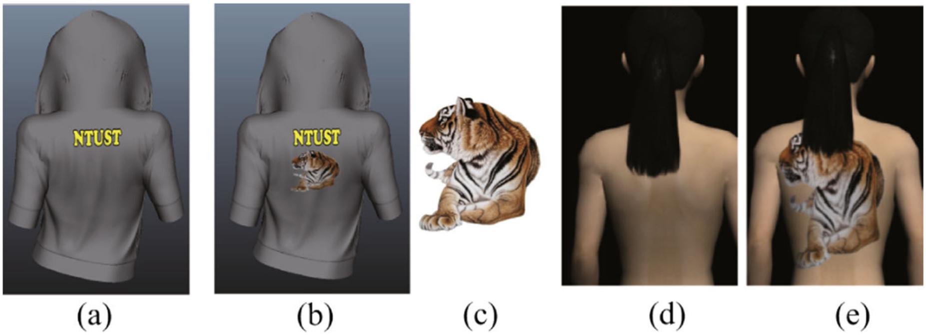

Fig. 1 This shows two motivation examples for detail addition. The cloth described in the news has a distinct tiger pattern on the back below the English letters, but the collected model in the graphics asset database does not have such a pattern. The female figure described in the news has a distinct tiger tattoo on her back, but the database only has a similar female figure without any tattoo on the back. Therefore, to fulfill the characteristic description in the news, artists must add the pattern and tattoo onto the cloth model and female figure respectively. **a** and **d** are the collected cloth and female figure in the graphics asset database. **b** and **e** are the detail-added results. **c** is the detailed texture

existing models. Therefore, how to add details onto existing models without affecting the pipeline and rendering efficiency is important and is also the goal of this paper. For clarity, an object is used to describe all tessellated textural elements in the animation for the rest of the paper.

Texture mapping [2] is an ubiquitous method to enhance object appearance and deliver realism to viewers. Therefore, texturing techniques and tools provided by 3DS Max and Maya can easily add distinguished details onto existing objects, but the process is time-consuming and complex, and to get it right also requires skills and experiences. Traditional texture parameterization algorithms [10, 11] create an extra set of texture coordinates for multi-texturing operations which increase the rendering time, require modification of the rendering pipeline, and require decent parameterization knowledge to do it well. Furthermore, local texture editing tools such as decal composition [8] can easily and intuitively add details with multi-layer texturing and UVW mapping. However, shading requires modification of the rendering pipeline and extra detail information including position, orientation, and etc. Furthermore, when artists manipulate and add new details, the existing texture, motion, and rigging settings are easily broken. As a result, this work explores the fact that all production objects have a set of well-tuned texture coordinates and develops an intuitive, efficient, and robust algorithm to add textural details such as pictorial tattoos and geometrical scars onto the existing model. The system uses a texture cube to directly stitching details onto the existing texture based on the correspondence between the cube and object. Stitching results can avoid modification of the rendering pipeline and provide flexibility for the usage of all existing motions and riggings on the object. Most importantly, the process is intuitive. The implementation has been incorporated into the NMA production pipeline to accelerate detail addition on animated characters and clothes and the rendered animations are perceptually good as shown in supplemental material.

The main contributions of this work lie in:

1. We shortly introduce a fast animation production pipeline which integrates a next generation game engine into a 3D animation software. At the same time, we also present

the challenges of fast production and corresponding solutions to illustrate the need of efficient and robust detail addition.
2. Accordingly, our detail addition algorithm stitches details onto the existing object textures using their original texture coordinates. It is efficient and robust because the stitching process does not require modification of the rendering pipeline and addition of any extra pass to the rendering pipeline.

The remainder of this paper is organized as follows: Section 2 discloses the production pipeline proposed by NMA and illustrates the shortage of existing texturing methods for robust and efficient detail addition. Section 3 describes the details of our detail-adding algorithm. Section 4 demonstrates and discusses the usage of our algorithms. Section 5 illustrates the comparison to the 3DS Max texture alias tool through a user study. Finally, Section 6 concludes with a discussion of limitations and future works.

2 Background

Animated news delivers more information than traditional broadcasting news but it also has several major technical challenges. Therefore, this section first describes the challenges and solutions to create animated news. Then the following gives a short discussion about why existing texture parameterization, 3D painting, and decal algorithms do not fulfill the efficient and intuitive requirement of detail addition for the animated news pipeline.

2.1 Animated news production pipeline

Although animated news can compensate the shortage of traditional broadcasting news, there are several challenges for producing an animation from a piece of breaking news:

1. **Breaking news happens unpredictably**:
   Broadcasting breaking news is time sensitive. When a piece of breaking news happens, the station must broadcast it immediately, and the corresponding animated news must be produced to broadcast as soon as possible. Based on NMA's empirical survey through interviewing different audiences, if a piece of animated news is not put online within 2 h, the meaning and benefit of animated news is lost. However, news happens unpredictably and thus, the production pipeline must always be ready and efficient.
2. **The number of production is large**:
   According to statistics, there are averaging more than 20 pieces of important news per day. In other words, the production of animated news is massive. The pipeline must be efficient and robust.
3. **Involved people are generally famous**:
   Because people involved in a piece of news are generally famous, they are easily recognized by audiences. Therefore, the characters used in animated news must be distinguished and close to the image of the involved people.
4. **The animation quality must be good**:
   If the quality is bad, the acceptance of the animation is low. Therefore, the production is not looking for a cheap solution but requires high-quality results, and this posts a large technical challenge to incorporate real-time techniques into off-line animation production tools.

5. **Information for a piece of breaking news is incomplete and updates frequently**:
   The most important aspect of news is to deliver correct information. When the new information about the piece of news comes in, the information must be updated immediately for animation production. Therefore, its work flow must be highly flexible and efficient.

Currently, animation softwares such as 3DS Max provide good user interfaces to create models, characters, and animations but lack efficiency to generate a 90-s animation within 2 h. Luckily, next generation game engines such as Unreal can compensate this performance limitation and be integrated into a 3D animation software for animated news production. Therefore, there are several critical technical decisions for the NMA pipeline as shown in the right of Fig. 2:

1. **GPU-based rendering**:
   Although global illumination can get photorealistic rendering results, it would take hours to render a frame, and the efficiency is not acceptable. GPU shaders provided by a game engine can provide interactive and realistic rendering results with proper preprocessing steps. NMA sets up the quality goal of rendering as shown in the left of Fig. 2 to generate an animation whose quality is just below global illumination with efficiency close to real-time applications. As a result, the pipeline is constructed by integrating a 3D game engine into an animation software. Real-time GPU shaders are used as main rendering tools. The in-house/tailor-made game engine is deployed to all computers equipped with GPUs to save the rendering cost. Our system is What-You-See-What-You-Get(WYSWYG) and allows artists to examine their animated results immediately.

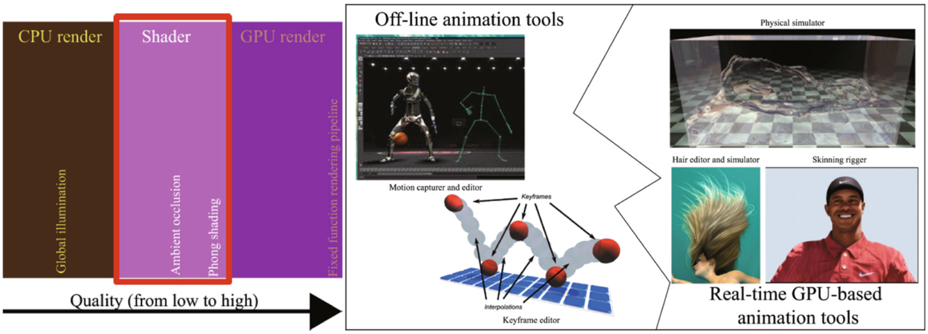

Fig. 2 The left shows the aim of rendering in NMA. The figure shows the spectrum of rendering qualities from high quality (*left*) to low quality (*right*). A CPU software renderer generally uses global illumination algorithms to physically and precisely render the virtual world for really high-quality results but it takes hours to finish the work. A GPU hardware renderer uses a fixed-function local illumination shading pipeline to quickly render the world in real time but its quality is low. The animated news pipeline aims at a good rendering quality using GPU-based shaders with real-time efficiency. The right shows the production pipeline incorporating a game engine with a 3D animation software. The left part shows traditional off-line animation tools such as transformation editors and motion capture units for scene and animation adjustment. The right part shows GPU-based animation tools such as physical simulators, artificial intelligence units, and renderers for fast animating and rendering. The production tasks are set up to take advantage of both platforms' strengths in different aspects for efficiency and flexibility

2. **Motion capture**:
   Although key-framed is the dominant animation technique, tuning a motion is time-consuming and cannot fulfill the efficiency requirement. Motion capture can create a proper motion and animation quickly. Therefore, engineers decide to adopt motion capture as the main animation method. Currently, desired and designed body and face motions can be captured and delivered within 10 minutes. With an in-house motion retargeting tool and a facial expression synthesizer, the captured motion can be quickly applied to all characters and faces.
3. **Graphics asset database**:
   3D models and characters are generally created from scratch for a traditional animation pipeline because the story is generally brand new for each movie. The creation is time-consuming to fulfil the efficiency requirement. Therefore, engineers decide to construct a graphics asset database containing a set of commonly used 3D models, characters, scenes, motions, and animation templates. 3D characters mainly consist of famous people such as Tiger Woods and Barack Obama. During production, the objects and motions are chosen from the database and modified to fulfill the description of news in a short period of time. Later, after production, artists can recycle newly constructed objects into the database. A group of artists are also responsible to extend the database with more objects.
4. **Procedural animation**:
   Although motion capture can help animate characters, the interaction among objects are still needed to generate reasonable animations. Engineers integrate a physical engine into the production pipeline to procedurally animate specific objects for production efficiency and intuitive manipulation. Game artificial intelligence(AI) and finite state machines are also used to steer characters and animate other objects more naturally and efficiently. Motion retargeting and facial expression synthesis are also applied to simplify the animation process.

After these considerations, the animated news production pipeline consists of the following steps as shown in Fig. 3:

1. **Visual development**:
   When a piece of news comes in, the news team starts to sketch out the story in papers. Additionally, the team uses the storyboard to set up the animation procedure and assign jobs to different teams for animation creation.
2. **Pre-production**:
   The capture team reads the storyboard description and captures the designed body and facial motions. The model team selects proper 3D models, characters, scanning facial models, and scenes from the database. Artists modify the selected models to fulfill the news description. If certain news elements are not found, artists directly create those elements from scratch with modeling and texture painting tools provided by the animation software.
3. **Shot Production**:
   The animation team creates the corresponding animation shots using those captured motions and created and selected news elements. The interaction among objects and characters are set up by game AI, procedural simulation, facial expression synthesis, and motion retargeting. Artists also manually clean up motions, set up lighting, and construct shooting paths.

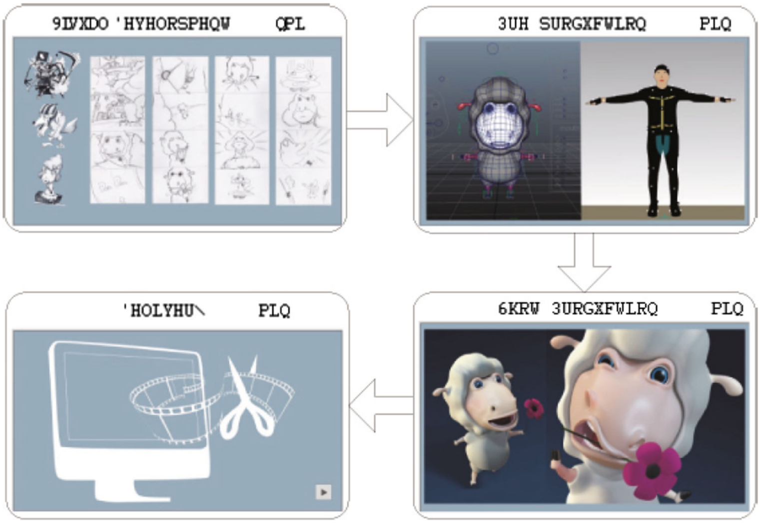

Fig. 3 This illustrates the NMA animated news pipeline: visual development which sketches out the story of the coming-in news in papers, pre-production which captures proper motions and prepares models and animations from the database, shot production which takes all material to generate the animated scenes for the news, and delivery which renders results with the game engine along with postprocessing effects and audio

4. **Delivery**:
   All scenes are rendered with the 3D game engine. Final animations are postprocessed along with voice and caption for broadcasting.

2.2 Limitation of texture mapping algorithms

Since there are more than 3000 characters and models in the graphics asset database of Next Media Animation, an artist picks an existing object from the database and the chosen object is modified to fulfill the news description which is an important trick for fast animation production. Detail addition can be easily done in several different ways: Firstly, texture reparameterization [7, 10–15] either constructs a set of texture coordinates to minimize texturing distortions according to an evaluation function or determine texture parameterization using optimization with a set of constraints. However, all these are too complex and not intuitive for artists to understand and operate them properly. Additionally, when rendering, the newly generated texture coordinates occupy the limited active texture space and must be rendered with multi-texturing techniques which require modification of the original pipeline and addition of extra rendering passes. Secondly, 3D painting [1, 3–5] is intuitive and generally equipped with a good user interface to construct an extra layer to the original texture. These techniques generally require manual dexterity and artistic skills, and the process is too time-consuming to become standard for mass production. Additionally, they require an extra set of texture coordinates for each extra layer and extra cost of UVW mapping at each pixel to increase the number of rendering passes and reduce rendering efficiency. Finally, dynamic texturing [8, 9, 15] uses a decal system adding details

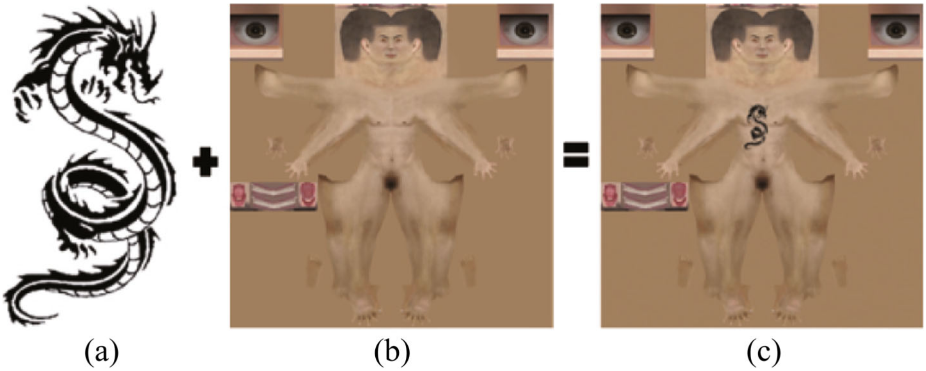

Fig. 4 Our work aims at stitching extra details represented as a detailed color or normal map onto an existing texture by composition in a similar concept of stamping/tattooing. **a** is an exemplar detail texture. **b** is an exemplar original texture. **c** shows the final result after stitching the detail texture onto the original texture

with multi-layer texturing and UVW mapping without understanding complex parametrization mechanism. However, they require supporting decal operations in the pipeline and need extra location, orientation, and texture information for details. Additionally, manipulating and adding these details would easily break existing texture, motion, and rigging settings. A decal system puts an emphasis on adding details in real time, and the precision is an issue for high-quality animated news. As a result, our goal is a simple and robust tool adapted from the stamping concept as shown in Fig. 4 to add extra details onto any kind of existing objects without affecting the current pipeline and adding extra rendering burdens. We achieve the goal by stitching details onto the existing texture by exploring the set of well-designed texture coordinates existing in all database objects for realistic and fantastic rendering.

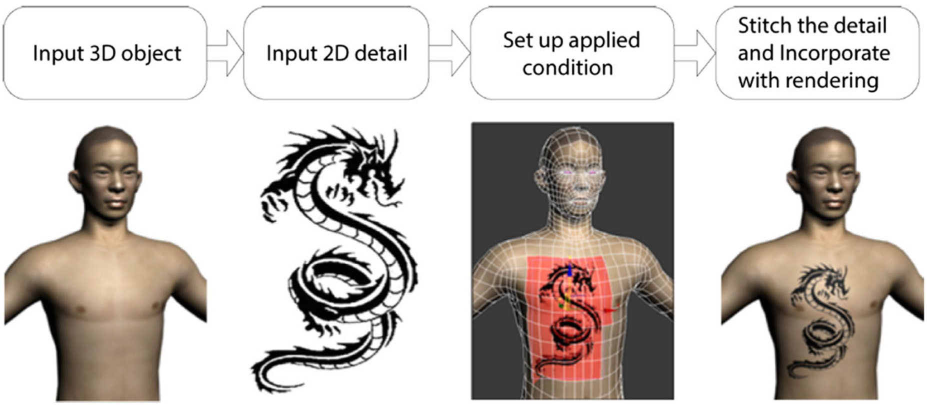

Fig. 5 This shows our work flow from selecting a 3D object to adding details onto the desired region. An artist first loads a 3D model containing a set of well-designed texture coordinates. He/she loads a detail texture and aligns the texture onto the 3D model with our texture cube. Finally, our system computes the correspondence transformation between the detail and original textures for stitching the detail texture onto the original texture

3 Laying extra detail on existing texture

Our proposed algorithm would like to directly modify the values stored in the existing textures instead of adjusting the parameterization space of the original textures. Since generally 3D objects stored in the database have been designed and created carefully, the set of texture coordinates in the objects has been manually adjusted by artists for visually pleasant results. They are even adjusted for rigging and facial synthesis to get the best result when generating animations. Therefore, our work would like to take advantage of this set of well-designed texture coordinates to add more details onto the existing texture without modifying

Fig. 6 This illustrates the steps of stitching the detail texture onto the original texture: detail texture cube construction defining the transformation media, cube alignment determining stitching conditions, affected face identification, bijective texture coordinate computation, stitching patch generation, and patch stitching to composite the two corresponding patches together

the rendering pipeline, without requiring skills to do it efficiently, and without hurting the rendering performance. Our detail-adding algorithm works as shown in Fig. 5. First, an artist selects a 3D object from the graphics asset database to create an element for the animation and the detail-added object must contain a set of well-designed texture coordinates which is a general practice for a production system. Second, the artist chooses or designs the detail texture which can describe the distinguished characteristics in the breaking news. Third, the artist sets up the stitching conditions by manipulating a texture cube created based on the detailed texture to decide the target faces on the object, and the front face of the cube is used to determine the mapping between the detail and original textures (Fig. 6). Finally, our algorithm chops the detailed texture into patches and stitches them onto the original texture according to the correspondence in vertices and texture coordinates of the cube and 3D object in the following steps, and Fig. 7 gives the summary pseudo code of our algorithm.:

1. **Define the detail texture cube**:
   The detail texture is loaded into the system and put onto the front face of a cube whose width and height are the same as the texture and the depth of the cube is set to a predefined value. The cube is used to decide the target faces on the object and its front face is defined with a transformation of $\left(\bar{t}^P, \mathbf{R}^P, \mathbf{S}^P\right)$ for its location, orientation, and scale in each dimension respectively. In other words, it is the projected media for our work to stitch details onto the existing texture. The cube front face called the projection face is designed to determine the mapping between the detail and original textures, and the projection of object vertices to the plane can be computed as

$$\overline{V}^P = \mathbf{M}^P \overline{V}^O \quad (1)$$

   where $\overline{V}^O$ is an object vertex, $\overline{V}^P$ is the projected location on the front face, and $\mathbf{M}^P$ is the transformation from the object to the texture. $\mathbf{M}^P$ is related to

Compute $\mathbf{M}^P = [(\mathbf{S}^P)^{-1}(\mathbf{R}^P)^{-1}(\bar{t}^P)^{-1}]$
For each face $\mathbf{F}_i^{target}$ in $\mathscr{F}^{target}$
    For each vertex $\overline{V}_{ij}^O$ of $\mathbf{F}_i^{target}$
        $\overline{V}_{ij}^P = \mathbf{M}^P \overline{V}_{ij}^O$
For each pixel $\mathbf{P}_i^{target}$
    For each face $\mathbf{F}_j^{target}$ in $\mathscr{F}^{target}$
        if($\mathbf{P}_i^{target}$ inside $\mathbf{F}_j^{target}$)
            $\mathbf{F}_j^{BC}(\overline{V}_{j0}^{BC}, \overline{V}_{j1}^{BC}, \overline{V}_{j2}^{BC})$ = BaryCentric($\mathbf{P}_i^{target}(x, y)$, $\mathbf{F}_j^{target}$)
            $\mathbf{P}_i^{target}(u, v) = \mathbf{F}_{BC}^{target}(\overline{V}_a, \overline{V}_b, \overline{V}_c) * F_j^{target}(u, v)$
            $\mathbf{T}_{value}^{target}(\mathbf{P}_i^{target}(u, v)) = \mathbf{P}_i^{target}(value)$

Fig. 7 This shows the pseudo code of our detail addition algorithm. $\mathbf{M}^P$ is the projection matrix from the world to the front face of the texture cube. $\mathscr{F}_{target}$ is the set of faces on the object inside the texture cube. $\overline{V}^P$ denotes the projection position of a vertex from the face to the texture projection plane. $\mathbf{P}^{target}$ is the set of all pixels on the detail texture. $\mathbf{F}^{BC}\left(\overline{V}_0^{BC}, \cdots, \overline{V}_q^{BC}\right)$ is the barycentric coordinate between the corresponding pixel and the projected $\mathbf{F}^{target}$. $BaryCentric()$ is the computation of the barycentric coordinate from the pixel on the original texture to the target triangle. $\mathbf{T}_{value}^{target}$ is the value of the original texture

$\left(\bar{t}^P, \mathbf{R}^P, \mathbf{S}^P\right)$ as

$$\mathbf{M}^P = [\left(\mathbf{S}^P\right)^{-1}\left(\mathbf{R}^P\right)^{-1}\left(\bar{t}^P\right)^{-1}] =$$
$$\begin{pmatrix} \frac{1}{\mathbf{S}_x^P} & 0 & 0 & 0 \\ 0 & \frac{1}{\mathbf{S}_y^P} & 0 & 0 \\ 0 & & \frac{1}{\mathbf{S}_z^P} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \mathbf{R}_{11}^P & \mathbf{R}_{21}^P & \mathbf{R}_{31}^P & 0 \\ \mathbf{R}_{12}^P & \mathbf{R}_{22}^P & \mathbf{R}_{32}^P & 0 \\ \mathbf{R}_{13}^P & \mathbf{R}_{23}^P & \mathbf{R}_{33}^P & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 0 & 0 & 0 & -\bar{t}_x^P \\ 0 & 0 & 0 & -\bar{t}_y^P \\ 0 & 0 & 0 & -\bar{t}_z^P \\ 0 & 0 & 0 & 1 \end{pmatrix}. \tag{2}$$

2. **Align the cube with the object**:
The bounding cube is used to indicate the scale, orientation, and location of the detail texture. The transformation manipulation tools of the editor are directly attached to the cube for artists to have customized intuitive adjustment. Furthermore, since detail addition is generally the focus of the user, our tool also provides automatically moving, rotating, and scaling with the scene editing camera to easily set stitching conditions and predict the possible stitching result.
3. **Locate the affected faces**:
The texture cube is used to collide with the target object and those faces inside the cube are denoted as a target face set, $\mathscr{F}^{target}$, which is defined as.

$$\mathscr{F}^{target} = \left\{\mathbf{F}_0^{target}, \cdots, \mathbf{F}_{n-1}^{target}\right\} \tag{3}$$

where $\mathbf{F}_i^{target}$ is a face from the attached object and colliding with the texture cube, and it is defined by its three vertices, $\left\{\overline{V}_{i0}^O, \overline{V}_{i1}^O, \overline{V}_{i2}^O\right\}$.
4. **Compute bijective texture coordinates**:
The vertices on the target faces from $\mathscr{F}^{target}$ are then projected onto the front face using (1) with transformation matrix $\mathbf{M}^P$, and it would be deduced by the normal of affected vertices. After projection, all the target vertices are projected onto the target front face and texture coordinates would be transferred into original texture domain. In other words, the projection transformation is used to decide the corresponding stitching patch from the detail texture for each target face.
5. **Generate stitching patch**:
After projection embedding, we would find the barycentric coordinate of front faces, $\mathbf{F}^{front} = (V_a, V_b, V_c)$, to the original faces, and then, we would resample pixels to the original texture. $\mathbf{P}^{target}$ is a pixel set of our detail texture on the front face. For generating composite texture, the mapping between a pixel of the original texture and a pixel of the detail texture is determined via their barycentric coordinate. Each pixel in $\mathbf{P}^{target}$ would be transferred to the original texture using the barycentric coordinate computed previously. The stitching patch can be resampled based on the color of pixels from the detail texture according to the barycentric mapping.
6. **Stitch to form the new texture**:
After the stitching patch is resampled from the detail texture, the detail patch can directly composite with the original texture patch on the target face. After stitching all target patches onto the existing texture, the new detail-added object texture is generated accordingly.

Fig. 8 This is an example of a women. The left is the diffuse map for the character after adding four body paintings or tattoos using our algorithm. The right is the front, back, and zoom-in views of the added details on the character

4 Results

In Next Media Animation, the house-made game engine is integrated with Autodesk 3DS Max 2011 for artists to generate animated news. Thus, our tool is implemented as a plugin using MAXScript and C++ for artists to directly apply details onto the existing models in 3DS Max. Because a texture cube is created as a polygonal mesh in 3DS Max, the scaling, rotation, and translation tools provided by the system can be directly and naturally used. It only takes tens of seconds to finish the detail addition process and the results are good according to the NMA artists. Figure 8 shows the result of using our algorithm to add four body paintings onto the breast, back, abdomen, and buttock of a 3D woman. In the supplemental material, the women body is animated with a walking motion and a running motion. Because the details are directly added to the existing diffuse map, when the skin of the body deforms during the walking motion and running motion, the painting is deformed with the skin. Figure 9 shows the result of adding four tattoos onto a man's front, back, and two arms. Similarly, the man is also animated with a walking motion and a running motion to check the deformation of tattoos. Figure 10 shows the result of stitching roses onto the diffuse map and bump map of a 3D Chinese cheongsam. In the supplemental material, the Chinese

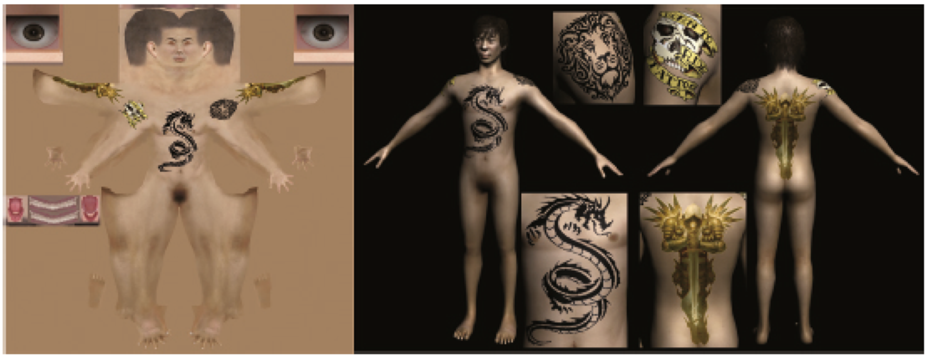

Fig. 9 This is an example of a man. The left is the diffuse map for the character after adding four tattoos using our algorithm. The right is the front, back, and zoom-in views of the added details on the character

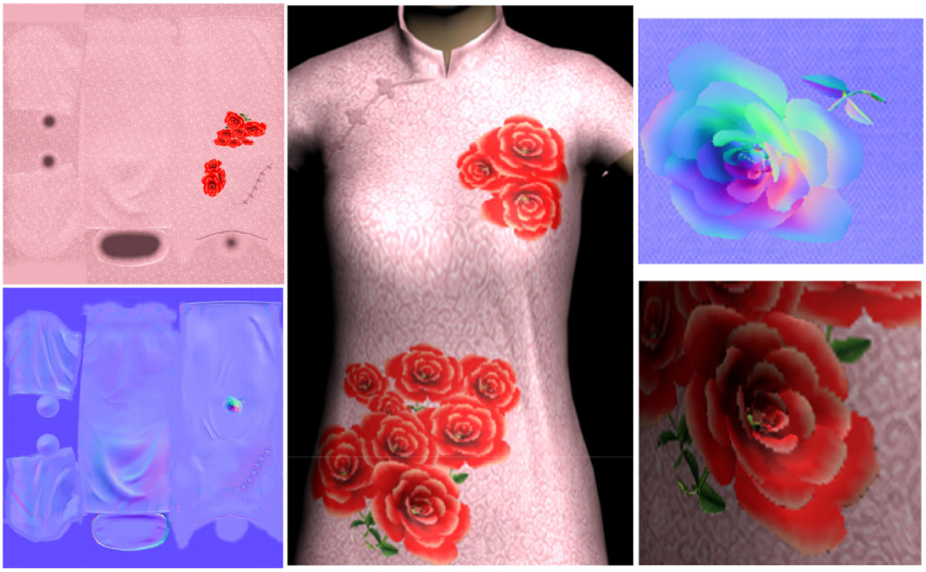

Fig. 10 This shows the result of stitching the color and structure of roses onto a diffuse and bump map. The two images in the left column are the original diffuse map and normal map of the Chinese cheongsam. The image in the middle is the rendering result of the detail-added 3D Chinese cheongsam. The two images in the right are the zoom-in view of the normal map and color shading of the rose

cheongsam is animated with a walking motion and a running motion. The deformation of the roses are naturally animated with the cheongsam. In addition, the Chinese cheongsam and the woman have also been used in a piece of animated news produced by Next Media Animation.

Moreover, we would like to understand whether our algorithm can work on different types of objects in addition to human models. Figures 11 and 12 show that our algorithm can successfully add details onto different types of objects. At the same time, we would like to understand the memory and texture usage of our algorithm when comparing to the texture parameterization algorithm with multi-texturing shading. We first get the texture parameterization implementation from Yao et al. [12] to add a set of texture coordinates for each added detail texture. The corresponding memory and texture usage statistics are collected and shown in Table 1. Our algorithm can render the results with the general shading pipeline without the need of the multi-texturing rendering techniques. Because our algorithm can use less numbers of textures for shading and relieve the the requirement for extra sets of texture coordinates, our implementation require less memory usage.

Due to the limitation of space, the supplemental material can be found in http://graphics.csie.ntust.edu.tw/pub/Tattoo/Main.html.

5 Usefulness user study

In order to verify the usefulness of our proposed tool, a user study was conducted. Subjects were asked to execute two detail addition tasks as shown in Fig. 13a and b. In the first task, subjects were asked to add a flower pattern onto a cylinder model with a flower pattern. In

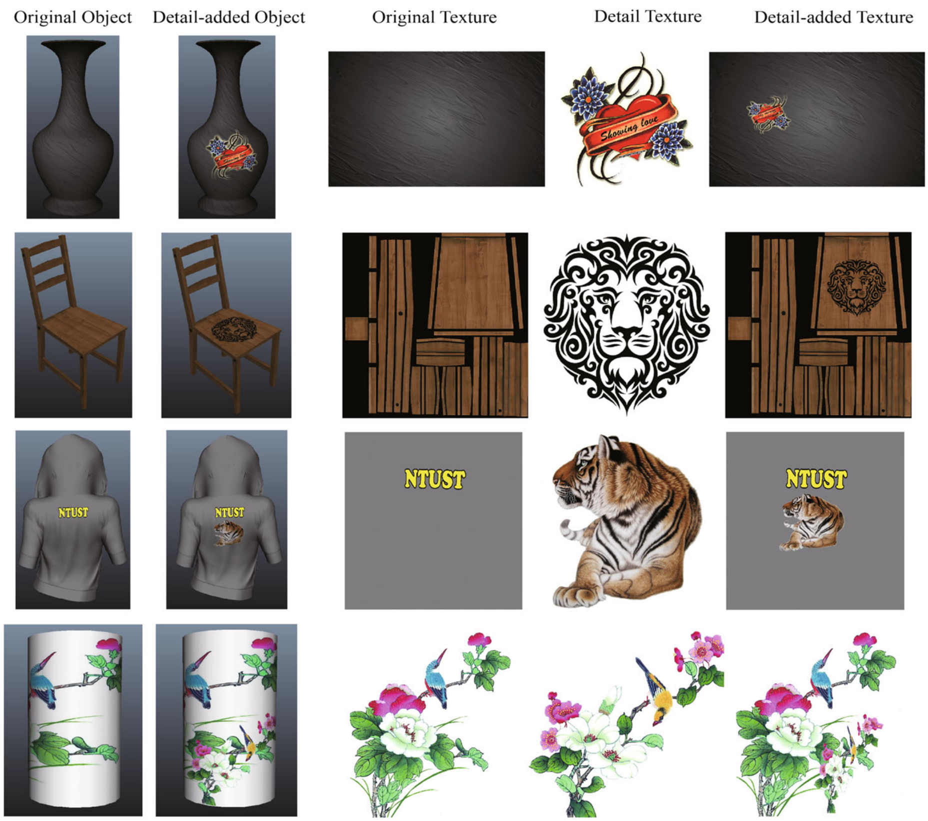

Fig. 11 This shows more results when stitching details onto different types of objects. From top to bottom are a vase, chair, cloth, and cylinder. The first column shows the rendering of the original models. The second column shows the rendering of the detail-added models. The third column shows the original textures. The fourth column shows the detail textures. The fourth column shows the detail-added textures

the second task, subjects were asked to add a tiger tattoo onto a women's back. These two tasks were designed in a similar way as those tasks regularly required during animated news production. Two tools were used to complete the tasks of detail addition. One is the texture alias tool provided by 3DS Max and the other is the proposed tool. Both tools are operated as follows. (1) Subjects were instructed about the location, orientation, and scale of the added detail texture. (2) Subjects were asked to add details onto the original texture using the built-in tool/our proposed tool. (3) The instructor reviewed the result. If the result was not satisfied, subjects repeated step 1 and 2 until the result is satisfied with the instructor. To avoid the studying effect, the order of using these two tools was randomly chosen for each subject. During the study, we recorded how much time the subject spent for obtaining the desired detail addition results. In total, 26 subjects participated in our experiments, and all are with normal or corrected-to-normal vision. Their ages range from 21 to 35 years; In addition, all of them have at least two years of experiences using animation tools. The means and standard deviations of the time for Task 1 and 2 are listed in Table 2. With the proposed tool, the time to add a detail texture was roughly reduced to 16 % of the one needed when using the built-in tool.

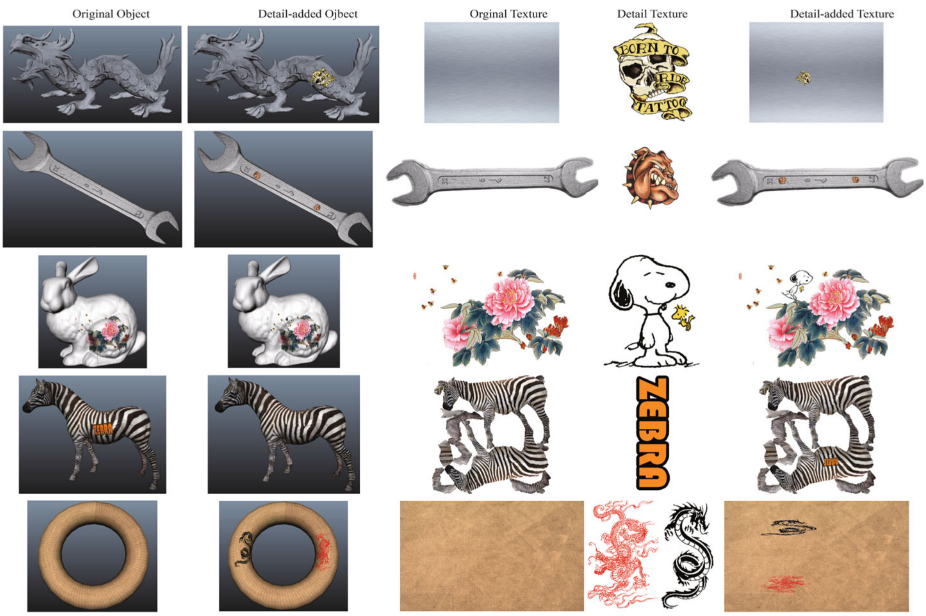

Fig. 12 This shows more results when stitching details onto different types of objects. From top to bottom are a dragon, spanner, bunny, zebra, and torus. The first column shows the rendering of the original models. The second column shows the rendering of the detail-added models. The third column shows the original textures. The fourth column shows the detail textures. The fourth column shows the detail-added textures

This study aims at evaluating the usefulness of our proposed tool, and there exist interfering factors, including sex, age, experience and others, which are called nuisance variations.

Table 1 This shows the comparison statistics between our algorithm and the texture parametrization algorithm proposed by Yao et al. [12]

	Memory (KB)		# of Textures	
	Yao's	Ours	Yao's	Ours
Chair	9.51	8.2	2	1
Cylinder	6.99	6.11	2	1
Torus	20.7	16.1	3	1
Spanner	234	181	3	1
Vase	283	247	2	1
Cloth	5700	4980	2	1
Bunny	2230	1950	2	1
Dragon	640	560	2	1
Women	350	223	5	1
Man	430	273	5	1
Cheongsam	51.4	39.9	4	2

Memory is the amount of memory usage for representing this model in GPU, and # of Textures denotes the number of textures and texture coordinate sets used for shading

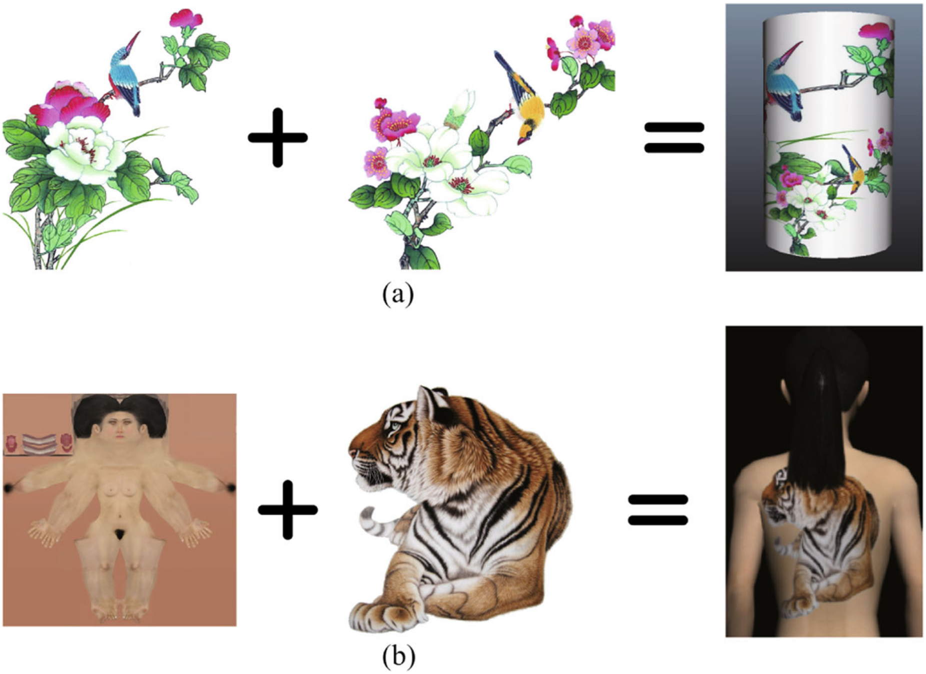

Fig. 13 This illustrates the detail addition tasks for studying the usefulness of the proposed tool. **a** shows the process of adding the flower pattern onto the cylinder and **b** shows the process of adding the tiger tattoo onto the back of the women figure

Random block factorization design (RBFD) applies a blocking procedure to isolate nuisance variations to prevent estimation errors. The data was collected and listed in the format as shown in Fig. 14 and the complete data of this study is provided in the supplemental material. The procedure involves 26 blocks of 2×2 homogeneous experimental units, where 26 blocks correspond to the subjects and 2×2 units correspond to 2 tools and 2 tasks respectively. In Fig. 14, $Y_{i,j,k}$ is the score set for the time required to finish the task. According to Kirk's book [6], the expectation for $Y_{i,j,k}$ can be expressed formally by a mixed model for type RBF-pq design as $Y_{ij} = \mu + \alpha_j + \beta_k + (\alpha\beta)_{jk} + \pi_i + \epsilon_{ijk}$ where μ denotes the overall population mean; α_j denotes the effect of the j-th tool; β_k denotes the effect of the k-th task; $(\alpha\beta)_{j,k}$ denotes the joint effect of the j-th tool and the k-th task; π_i denotes the

Table 2 This shows the means and standard deviations of the time of detail addition for task 1 and 2 in the usefulness study

	Time(sec)	
	Mean	Std
Task 1(3DMax)	32.0	12.3
Task 1(Ours)	259	151
Task 2(3DMax)	59.0	21.2
task 2(Ours)	331	294

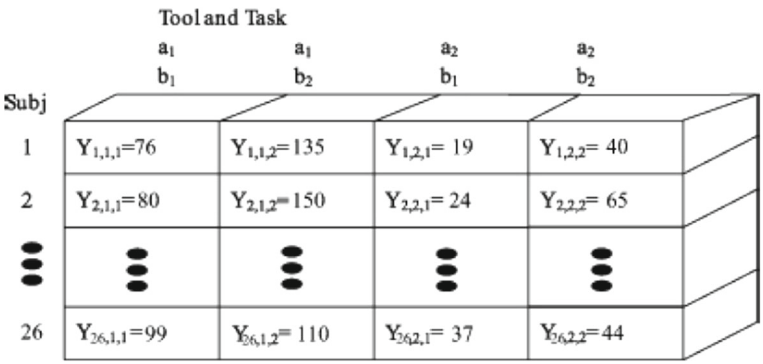

Fig. 14 The data arrangement for RBF-22 of the randomized block factorial design (RBFD). Y_{ijk} denotes a score in one of the $i = 1, \cdots, n$ blocks for subjects, $j = 1, \cdots, p$ for tool types, and $k = 1, \cdots, q$ for scenes. Since there are two tools, the 3DS Max built-in and our proposed tool, and two scenes, there are totally four combinations for each subject

effect of the i-th subject; and $\epsilon_{i,j,k}$ denotes the experimental error. The single and overall deviation can be derived from experimental data in a similar manner as discussed in Kirk's book [6]. Finally, the F value of the collected data on the tool types is computed according to Table 9.5-1 of the book [6]. After computing F-variance (70.84), ANOVA tests whether the differences among subjects are important and whether our tool enhances the efficiency of detail addition at 0.001 level of significance. The result shows that the tool type is an affecting factor for the detail addition efficiency. In other words, the proposed tool does improve the efficiency of detail addition.

6 Conclusion

This work gives short description of the animated news production pipeline used in Next Media Animation and develops a simple and robust detail-adding algorithm to quickly add distinguished elements onto 3D objects under the constraints of no extra rendering pass, no change to the pipeline and no need of artistic skills. Since the algorithm keeps the original rendering pipeline intact, rendering efficiency is maintained. Our algorithm depends on the existing set of texture coordinates, the coordinate quality affects the adding result. Additionally, when projecting onto objects with irregular triangulation and quadrangulation, the planar projection can also induce artifacts. Therefore, we would like to explore other parameterization spaces to construct the correspondence between the existing and detail textures for better stitching results. Furthermore, we use the texture cube as the transformation media and thus, when the attached part of the object contains a high curvature, the projection may induce distortions on the final stitching results. We would like to remove these artifacts using decal decomposition [8] to estimate the transformation and decompose our detailed patches. Resolution mis-matching between the detail and original texture results in blurring artifacts, and we would like to develop an automatic resolution matching algorithm for the existing and detail textures based on super resolution algorithms to avoid the possible resampling artifacts when stitching details.

Acknowledgments This work is supported by the National Science Council, Taiwan under NSC 104-2221-E-011-029-MY3, 103-2221-E-011-114-MY2, 397 and 103-2218-E-011-014, NSC 103-2221-E-011-076.

References

1. Benson D, Davis J (2002) Octree textures. ACM Trans Graph 21(3):785–790
2. Blnn J, Newell M (1976) Texture and reflection in computer generated images. Commun ACM 19(10):542–547
3. Carr NA, Hart JC (2004) Painting detail. ACM Trans Graph 23(3):845–852
4. DeBry Dg, Gibbs J, Petty DD, Robins N (2002) Painting and rendering textures on unparameterized models. ACM Trans Graph 21(3):763–768
5. Gingold YI, Davidson PL, Han JY, Zorin D (2006) A direct texture placement and editing interface. In: Proceedings of the 19th annual ACM symposium on user interface software and technology, UIST '06, pp 23–32
6. Kirk R E (1982) Experimental design. Wiley Online Library
7. Lee TY (2008) Texture mapping with hard constraints using warping scheme. IEEE Trans Vis Comput Graph 14(2):382–395
8. Schmidt R, Grimm C, Wyvill B (2006) Interactive decal compositing with discrete exponential maps. ACM Trans Graph 25(3):605–613
9. Sheffer A, Lévy B, Mogilnitsky M, Bogomyakov A (2005) Abf++: fast and robust angle based flattening. ACM Trans Graph 24(2):311–330
10. Tarini M, Hormann K, Cignoni P, Montani C (2004) Polycube-maps. ACM Trans Graph 23(3):853–860
11. Tzur Y, Tal A (2009) Flexistickers: photogrammetric texture mapping using casual images. ACM Trans Graph 28(3):1–10
12. Yao CY, Chu HK, Ju T, Lee TY (2009) Compatible quadrangulation by sketching. Comput Anim Virtual Worlds 20(2–3):101–109
13. Zhang E, Mischaikow K, Turk G (2005) Feature-based surface parameterization and texture mapping. ACM Trans Graph 24(1):1–27
14. Zhou K, Wang X, Tong Y, Deesbrun M, Guo B, Shum HY (2005) Texturemontage: seamless texturing of arbitrary surfaces from multiple images. ACM Trans Graph 24(3):1148–1155
15. Zwicker M, Pauly M, Knoll O, Gross M (2002) Pointshop 3d: an interactive system for point-based surface editing. ACM Trans Graph 21(3):322–329

Yu-Chi Lai received the B.S. from National Taiwan University, Taipei, R.O.C., in 1996 in Electrical Engineering Department. He received his M.S. and Ph.D. degrees from University of Wisconsin-Madison in 2003 and 2009 respectively in Electrical and Computer Engineering and his M.S. and Ph.D. degrees in 2004 and 2010 respectively in Computer Science. He is currently an assistant professor in NTUST and his Research interests are in the area of graphics, vision, and multimedia.

Chun-Wei Wang received the B.S. and M.S. degree from National Taiwan University of Science and Technology, Taipei, R.O.C., in 2013 in Computer Science department. He is currently an senior software engineer in International Games System CO., LTD. and his research interests are in the area of graphics, visual effect and game design.

Kuang-Yi Chen received the B.S. and M.S. in Department of Computer Science and Information Engineering from National Taiwan University of Science and Technology, Taipei, R.O.C. in 2011 and 2014 respectively. And He is currently an employee in International Game System. His research interests are computer graphics and GPU shader.

Dobromir Todorov received his B.S form Technical University - Varna, Varna, Bulgaria in 2009 in Computer Systems and Technologies. He has received the M.S. degree from National Taiwan University of Science and Technology, Taipei, R.O.C., in 2013 in Computer Science department. He is currently a senior software engineer in Next Media CO., LTD. and his research interests are in the area of graphics, visual effect and game design.

Chih-Yuan Yao is an Assistant Professor in the Department of Computer Science and Information Engineering at National Taiwan University of Science and Technology (NTUST), Taipei, Taiwan. He received his M.S. and Ph.D. degrees in Computer Science and Information Engineering from National Cheng-Kung University, Tai-wan, in 2003 and 2010, respectively. His research interest is computer graphics, including mesh processing and modeling, and non-photorealistic rendering (NPR).