Re-texturing by intrinsic video

Jianbing Shen, Xing Yan, Lin Chen, Hanqiu Sun, Xuelong Li

Beijing Lab of Intelligent Information Technology, School of Computer Science, Beijing Institute of Technology, Beijing 100081, PR China
Department of Computer Science and Engineering, The Chinese University of Hong Kong, Shatin, NT, Hong Kong
Center for Optical Imagery Analysis and Learning (OPTIMAL), State Key Laboratory of Transient Optics and Photonics, Xi’an Institute of Optics and Precision Mechanics, Chinese Academy of Sciences, Xi’an 710119, Shaanxi, PR China

Article history:
Received 24 February 2013
Received in revised form 1 December 2013
Accepted 9 February 2014
Available online xxxx

Abstract

In this paper, we present a novel re-texturing approach using intrinsic video. Our approach first indicates the regions of interest by contour-aware layer segmentation. The intrinsic video including reflectance and illumination components within the segmented region is recovered by our weighted energy optimization. We then compute the texture coordinates in key frames and the normals for the re-textured region using the optimization approach we develop. Meanwhile, the texture coordinates in non-key frames are optimized by our energy function. When the target sample texture is specified, the re-textured video is finally created by multiplying the re-textured reflectance component with the original illumination component within the replaced region. As shown in our experimental results, our method can produce high quality video re-texturing results with a variety of sample textures, and also the lighting and shading effects of the original videos are well preserved after re-texturing.

© 2014 Elsevier Inc. All rights reserved.

1. Introduction

Image/video re-texturing is a process to replace the existing texture in the concerned region of a photograph or video by new textures, while preserving the original shading and lighting effects. Even for a rich body of work on image/video re-texturing [1,3,5,8,10,17,16,24], the re-texturing using intrinsic video for recovering the illumination and shading effects receives relatively little attention.

Intrinsic images are usually referred to the separation of illumination and reflectance components from an input photograph. The recovery of intrinsic images aims at decomposing an image into illumination component and reflectance component. It is still technically challenging on how to recover the intrinsic images when only a single photograph is available. There has been approaches proposed for such decomposition either from a single image or from image sequences. Tappen et al. [4] proposed an algorithm for recovering intrinsic images from a single photograph. Their approach is based on a trained classifier, which classifies the image derivatives as being caused by either shading or reflectance change. The shading
and reflectance images are calculated by the classified derivatives. Shen et al. [12] improved Retinex-based algorithm by assuming similar textures correspond to similar reflectance. Recently, Bousseau et al. [15] presented a novel approach for calculating the intrinsic images only from a single image. Their approach requires user interactions for indicating the regions of constant reflectance or illumination, and achieves intrinsic image decomposition by a novel propagation energy using linear least-squares. Shen et al. [18,26] proposed a novel recovery algorithm of intrinsic images using optimization, which is based on the local property assumption of the objects and the Lambertian model. However, these approaches are only limited to the single image re-texturing, and do not support intrinsic video recovery and video re-texturing.

Our work is also closely related to image and video re-texturing techniques, which have been attracting large attentions for a long time, and image-based texture replacement methods have been developed. Liu et al. [3] proposed to use user-assisted adjustment on the regular grid of the real texture, and obtained a bijective mapping between the regular grid of the texture and the deformed grid of the surface image. Their method requires elaborate user interactions and is more suitable for near regular textures. Fang and Hart [1,5] proposed an efficient object texture replacement technique called Textureshop. Their method is based on the assumption that the lighting satisfies Lambertian reflectance model. However, in order to obtain an accurate normal map, non-trivial user interaction is often required. Khan et al. [6] presented an image-based material editing method for making objects transparent and translucent. Their technique also supports re-texturing of high dynamic range (HDR) images with arbitrary surface materials. The major limitation is that their approach assumes the input image is given as a high dynamic range image, so not suitable for re-texturing an ordinary image. More recently, Guo et al. [10] proposed an image and video re-texturing approach which preserved the original shading effects without knowing the underlying surface and lighting conditions. Their method however needs to create a parameterized mesh for the replaced region by the user interaction, and introduces visual artifacts when the mesh parametrization is incorrect.

In this paper, we present a novel recovery algorithm of intrinsic video for decomposing the input video into reflectance and illumination components, which is inspired by the recent work on intrinsic images [4,15]. We propose an efficient recovery approach of intrinsic video based on a new weighted energy optimization, for removing the compressed JPEG artifacts. Then a novel video re-texturing method is developed using intrinsic video, which can be used to re-texture the uniform texture regions. In order to reduce the un-consistent texture distortions in the video re-texturing process, we also present the way for optimizing texture coordinates for video re-texturing, which requires no 3D models and can generate realistic results with the input sample texture.

The main contributions of our paper are summarized as follows:

- A novel re-texturing approach is proposed using intrinsic video for uniform texture region, which is based on our new methods in computing the texture coordinates through energy optimization;
- An efficient recovery algorithm of intrinsic video is presented by using a weighted energy optimization.

The remaining parts of this paper are organized as follows. Section 2 describes our approach in intrinsic video re-texturing, including layer segmentation, weighted energy optimization, normal recovery, and intrinsic video re-texturing. Section 3 presents the experimental results using our approach and discussion with others, and Section 4 gives the summary.

2. Intrinsic video re-texturing

In our framework, we first extract the region to be replaced from the input video using contour-aware layer segmentation. We indicate the contour of the re-textured regions in key frames, and use the contour-aware layer tracking technique to identify the segmented regions in other non-key frames (Section 2.1). We recover the intrinsic video from the above segmented texture regions. Then, we use scribble brushes to indicate the pixels that share a similar illumination or a similar reflectance in key frames, and the positions of scribble brushes for other non-key frames are tracked through energy optimization. Thus, the final intrinsic video can be calculated using our new weighted energy optimization method (Section 2.2).

With the recovered intrinsic video, we compute the normal maps using optimization for the replaced regions (Section 2.3), and obtain the texture coordinates in key frames. Using the texture coordinates in the nearest adjacent key frames, we optimize the texture coordinates in other non-key frames by optical flow. When the target sample texture is specified, the re-texturing video is created through multiplying the re-textured reflectance component by the original illumination component (Section 2.4). As shown in our experimental results, our video re-texturing approach can produce high quality results with a variety of sample textures, while the lighting and shading effects of the original video are preserved well. Table 1 presents pseudocode for the proposed video re-texturing scheme. More details of intrinsic video recovery and video re-texturing scheme are presented in the following sections.

2.1. Contour-aware layer segmentation

To extract the regions to be replaced, our system supports contour-aware layer segmentation using energy optimization. Similar to the interactive tracking and segmentation approach [9,13,20], our system is based on the user defined polygon \( C = \{ c_k : c_k \in \mathbb{R}^2 \}_{k=1}^N \) which indicates the contour of the regions to be replaced in key frames. In order to extract the layer of the re-textured regions, we optimize the following energy function:
$E(f) = \sum_{N_{k=1}} X \sum_{q \in N_{k}} w_{k}(q)f_{2}(c_{k} + w_{k} + q) - f_{1}(c_{k} + q) + \lambda \sum_{k=1}^{N} \frac{\dd{d}}{d}(c_{k} - c_{k+1})] + \frac{d}{d}(w_{k} - w_{k+1})$  \hspace{1cm} (1)

Table 1

Pseudocode for re-texturing by intrinsic video.

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
</table>
| Input: the original video and the replaced texture | // Extract the re-textured regions from input video
while (1 end of input video) do
  if (key frame) then
    Indicate the contour of the re-textured regions
  else
    Calculate the re-textured regions by contour-aware layer segmentation (Eq. 1)
  end if
end if
end
Output: the segmented video within re-textured regions

// Recover intrinsic video for the re-textured regions
while (1 end of segmented video) do
  if (key frame) then
    Indicate the scribbles in key frames
  else
    Do scribble tracking using energy optimization
  end if
  Calculate the intrinsic video using our newly weighted energy optimization (Eq. (2)-(8))
end if
Output: the intrinsic video within re-textured regions

// Perform video re-texturing
while (1 end of segmented video) do
  Do normal recovery (Eq. (9)-(15))
  if (key frame) then
    Compute texture coordinates (Eq. 16)
  else
    Optimize texture coordinates using optical flow and texture coordinates in nearest-adjacent key frame (Eq. 17 and 18)
  end if
  Create re-texturing video (Eq. 19 and 20)
end if
Output: the re-texturing result video

2.2. Intrinsic video by weighted energy optimization

Here, we present a novel method for recovering the intrinsic video. Our method is inspired by the work of Bousseau et al. [15] in the context of intrinsic images recovery technique. Their approach assumes that the set of reflectance colors is sparse in the distribution of colors in natural scenes, and the reflectance variations lie locally in a 2D plane in RGB color space.

Therefore, the pixels in planes parallel to the reflectance have constant illumination. As defined in [15], we use the following energy to calculate the reflectance and illumination components:

$E(s, a) = \sum_{i} \left[ \sum_{j \in N_{i}} (s_{i} - a_{i} \cdot l_{j} + \epsilon a_{i})^{2} \right]$  \hspace{1cm} (2)

where $s$ denotes the illumination component, $N_{i}$ defines a local window over the neighborhood of pixel $i$, $a$ is a constant over a local window $N_{i}$, $i$ and $j$ are the position indices of pixels, and $\epsilon$ is a small constant (typically $10^{-6}$) for avoiding the ambiguous cases.

The main limitation in [15] is that their approach will introduce visual artifacts for strongly compressed JPEG images. Through our extensive experiments, we have found that the quality of intrinsic image can be improved by adding a Gaussian weighting function $W_{i}$ as follows:
\[
\begin{align*}
E(s, a) &= \sum_i \left[ \sum_{j \in N_i} W_j (s_j - a_i \cdot I_j)^2 + ea_i^2 \right] \\
W_j &= \frac{\exp(||j - i||^2)}{2a^2}
\end{align*}
\]

The simple improvement is effective to reduce JPEG compression artifacts. In fact, previous decomposition (Eq. 3) is equivalent to:

\[
e(s, a_i) = \left[ \begin{array}{c}
\sqrt{W_{ij}} \\
\sqrt{W_{im}} \\
0 \\
0 \\
0
\end{array} \right] - \left[ \begin{array}{c}
\sqrt{W_{ij}} \\
\sqrt{W_{im}} \\
\sqrt{W_{im}} \\
\sqrt{W_{im}} \\
\sqrt{W_{im}}
\end{array} \right]
\left[ \begin{array}{c}
0 \\
0 \\
\sqrt{W_{im}} \\
\sqrt{W_{im}} \\
\sqrt{W_{im}}
\end{array} \right]
\left[ \begin{array}{c}
s_{ij} \\
0 \\
0 \\
0 \\
0
\end{array} \right]
\left[ \begin{array}{c}
\sqrt{W_{ij}} \\
\sqrt{W_{im}} \\
\sqrt{W_{im}} \\
\sqrt{W_{im}} \\
\sqrt{W_{im}}
\end{array} \right]
\left[ \begin{array}{c}
0 \\
0 \\
\sqrt{W_{im}} \\
\sqrt{W_{im}} \\
\sqrt{W_{im}}
\end{array} \right]
\left[ \begin{array}{c}
a_i^e \\
a_i^e \\
a_i^e \\
a_i^e \\
a_i^e
\end{array} \right]
\]

\[
\hat{W} = \text{diag} \left( \sqrt{W_{ij}}, \ldots, \sqrt{W_{im}}, 0, 0, 0 \right)
\]

We can further re-write Eq. (5) as follows:

\[
e(s, a_i) = (\hat{W}_i S_i - \hat{W}_i M_i A_i)^2 = (\hat{W}_i S_i - M_i A_i)^2
\]

where \( M_i = \hat{W}_i M_i \). The above local energy is minimized by a linear least-square algorithm as follows:

\[
f(S_i) = (\hat{W}_i S_i - \hat{M}_i (\hat{M}_i^T \hat{M}_i)^{-1} \hat{M}_i^T \hat{W}_i S_i)^2
\]

Then the global energy is obtained using the matrix \( N_i \) as follows:

\[
\sum_i (N_i S_i)^2 = \sum_i S_i^T N_i^T N_i S_i
\]

\[
N_i = \hat{W}_i - \hat{M}_i (\hat{M}_i^T \hat{M}_i)^{-1} \hat{M}_i^T \hat{W}_i
\]

Eq. (8) is optimized using Gaussian-Seidel method in our implementation. Similar to [15], in order to let the user specify the local cues of the reflectance and illumination, we integrate the constant-reflectance brush and constant-illumination brush into our energy function.

Fig. 1 compares the intrinsic images recovered by [15] and our approach. We can see that our approach achieves better visual quality with less JPEG compression artifacts, and preserves more shadows and highlight components in the illumination image (Fig. 1(c)). Although Boureau et al.’s interactive approach is effective to reconstruct intrinsic images, it is very tedious and error-prone to require the user to indicate scribbles for every video frame. To solve this problem, we propose an efficient intrinsic video recovery method, which is based on the aforementioned weighted energy optimization and contour tracking technique (Section 2.1). The user only needs to indicate the contour brushes for the re-textured region in key frames, then the contour tracking approach [9] is applied to other non-key frames. After all the contour brushes are tracked through the whole video, our algorithm calculates the illumination components and reflectance components automatically.

As shown in Fig. 2 and Fig. 5, our algorithm can produce consistent intrinsic video through a few user scribbles on key frames.

2.3. Normal recovery

Normal recovery refers to the estimation of Shape-from-Shading (SfS) that aims to compute surface normal from shading information in the image. Given an input image \( I \), our SfS method is based on the following Lambertian imaging model:

\[
I = p N^T L
\]

where \( p \) denotes the surface albedo, \( N = (n_x, n_y, n_z)^T \) is a unit vector that represents the surface normal, and \( L = (l_x, l_y, l_z)^T \) denotes a unit vector that represents the direction of a distant light source.

Unlike the interactive normal reconstruction method proposed in [11], their approach requires the user to assign normals to a few pixels, and then minimizes an energy function to obtain the estimated lighting direction. In order to automatically solve the unknown terms on the right side of Eq. 5, we simplify the process by initializing the lighting direction \( L \) with the light source detection algorithm [6]. The normals \( N \) can be computed by minimizing the following energy function [11]:

\[
E = \sum_{i \in R} \frac{1}{p} || l_i - N_i^T L ||^2 + \sum_{i \neq j} || N_i - N_j ||^2
\]
where $R$ denotes the re-textured region, $(i,j)$ is a first-order neighbor pair, and $\lambda$ denotes a regularization factor. The first term in the energy function is the data energy, which defines the fitness of the imaging model, while the second term is the smoothness energy, which defines the smoothness constraint between the normals.

Now we can improve the accuracy of normals using Gaussian-Seidel method with successive over-relaxation (SOR) as follows:

$$
N_{n+1}^k = N_n^k + \lambda \left( \frac{1}{\rho} \times I - L^T \cdot N_{n+1}^k \right)
$$

$$
\hat{p}^{n+1} = \hat{p}^n - \alpha (I \times (\hat{p}^n \cdot I \cdot N_{n+1}^k))
$$

where $k = \{1, 2, 3\}$ represents the surface normals in $x, y$, and $z$ directions, respectively. $\bar{N}$ denotes the average normal in the first-order neighbor, and $\hat{p} = 1/\rho, \alpha = 0.5$.

In order to obtain accurate normals with the evenly distributed errors, similar to [11], we use the following energy model to reconstruct the height field $H$:

$$
E_3 = \sum_{(i,j)} ((h_i - h_j) - q_{ij})^2
$$

where $h_i$ and $h_j$ denotes the heights at pixel positions $i$ and $j$, and $q_{ij}$ represents the relative height between $i$ and $j$ on the surface.

Fig. 3 gives the details on how to calculate $q_{ij}$. We also use the following Gaussian-Seidel method with successive over-relaxation:

$$
h^{n+1} = \bar{h}^n + \bar{q}^n
$$

$$
h^{n+1} = \omega h^n + (1 - \omega) h^{n+1}
$$

where $\omega$ is a relaxation factor ($\omega \in (0, 2)$) to help convergence of a diverging iterative process. $\bar{h}$ and $\bar{q}$ denote the average depth and the relative depth in the first-order neighbor, respectively.

After the depth is reconstructed, we can obtain the surface gradients $g_x = [1, 0, \frac{\partial h}{\partial x}]$ and $g_y = [0, 1, \frac{\partial h}{\partial y}]$. Therefore, the updated normals are reconstructed by the following equation:

$$
\bar{N} = g_x \times g_y
$$

For video normal recovery application, high accuracy is considered to be more important than performance. Therefore, we calculate the normals frame by frame for the textured regions.
2.4. Re-texturing intrinsic video

In general, the recovered intrinsic video and normals are sufficient to be used to estimate a warping of a given texture $T$. In order to reduce the user interactions, our approach employs the contour-aware tracking technique to segment the re-textured region for video. Unlike [10], our algorithm does not need to generate a specific triangle mesh for calculating the texture distortion coordinates. Bousseau et al. [15] proposed a novel reflectance editing approach for texture replacement, they calculate the replaced texture coordinates using a similar normal-from-shading approach [1] in the reflectance image. Their approach can produce good re-texturing result for single image, but it produces the drifting phenomenon of re-texturing regions when we directly apply the method to process the videos frame by frame. In contrast, our approach address the intrinsic video recovery and re-texturing using intrinsic video, where the temporal coherence across the video sequence is preserved well.

Fig. 2. Illustration of intrinsic video recovery: (a)–(d) input video; (e)–(h) the recovered intrinsic (reflectance component) video; (i)–(l) the recovered intrinsic (illumination component) video; (m)–(p) the re-textured video.

Fig. 3. Illustration of computing texture coordinates using the recovered normals.

2.4. Re-texturing intrinsic video

In general, the recovered intrinsic video and normals are sufficient to be used to estimate a warping of a given texture $T$. In order to reduce the user interactions, our approach employs the contour-aware tracking technique to segment the re-textured region for video. Unlike [10], our algorithm does not need to generate a specific triangle mesh for calculating the texture distortion coordinates. Bousseau et al. [15] proposed a novel reflectance editing approach for texture replacement, they calculate the replaced texture coordinates using a similar normal-from-shading approach [1] in the reflectance image. Their approach can produce good re-texturing result for single image, but it produces the drifting phenomenon of re-texturing regions when we directly apply the method to process the videos frame by frame. In contrast, our approach address the intrinsic video recovery and re-texturing using intrinsic video, where the temporal coherence across the video sequence is preserved well.
In order to overcome the drifting phenomenon of texture coordinates, we propose a novel energy optimization approach for calculating texture coordinates. We first optimize the texture coordinates according to the recovered normals in key frames by the following energy function:

\[ E' = \sum_{(i,j) \in M} ((t^v_i - t^v_j) - \Delta t^v_{ij})^2 \]

\[ E' = \sum_{(i,j) \in M} ((t^h_i - t^h_j) - \Delta t^h_{ij})^2 \]

\[ \Delta t^v_{ij} = \frac{\sqrt{n_z^2 + n_u^2}}{|n_z|}, \Delta t^h_{ij} = \frac{\sqrt{n_z^2 + n_v^2}}{|n_z|} \]  \hspace{1cm} (16)

where \( M \) denotes the segmentation mask for re-textured region, \( \Delta t^v_{ij} \) and \( \Delta t^h_{ij} \) are the texture distortion coordinates in horizontal direction and vertical direction, respectively. \( t^v_i, t^v_j, t^h_i \) and \( t^h_j \) represent the texture coordinates in pixel positions \( i \) and \( j \). Fig. 3 shows the details for calculating the values of \( \Delta t^v_{ij} \).

For other non-key frames, we further optimize the texture coordinates according to the high accurate optical flow estimation algorithm [2] and the obtained texture coordinates in nearest-adjacent key frame with the following energy function:

\[ t^v_{k,ij} = \begin{cases} 
 t^v_{k,ij, p - u(p), \min} & \text{if } Q_{k,ij, p - u(p)} = 1; \\
 \arg\min_{(p,q) \in Q_{k,ij}} ((t^v_{k,ij,p} - t^v_{k,ij,q}) - \Delta t^v_{k,ij,pq})^2, & \text{otherwise};
\end{cases} \]  \hspace{1cm} (17)

\[ t^h_{k,ij} = \begin{cases} 
 t^h_{k,ij, p - v(p), \min} & \text{if } Q_{k,ij, p - v(p)} = 1; \\
 \arg\min_{(p,q) \in Q_{k,ij}} ((t^h_{k,ij,p} - t^h_{k,ij,q}) - \Delta t^h_{k,ij,pq})^2, & \text{otherwise};
\end{cases} \]  \hspace{1cm} (18)

where \( i \) denotes the current \( i \)-th non-key frame, and \( k \) denotes its corresponding nearest-adjacent key-frame. \( p \) and \( q \) are the adjacent pixel coordinates. \( U(p) \) and \( V(p) \) represent the motion vector in horizontal direction and vertical direction, which are calculated through optical flow estimation algorithm [2]. Here, \( Q_{k,ij, p - u(p)} = 1 \) means that the pixel \( p \) (in current frame \( i \)) is inside the target region to be replaced, and the pixel \( p \) is derived from the pixel \( p - U(p) \) (in forward frame \( i - 1 \)) using optical flow tracking algorithm.

The new color value \( C(x,y) \) of the reflectance component is derived from the color value of the sample texture \( T(t^v, t^h) \) with the following equation:

\[ C(x,y) = T(t(i,j)^x, t(i,j)^y) \]  \hspace{1cm} (19)

Finally, we multiply the re-textured reflectance component \( C(x,y) \) by the illumination component \( S(x,y) \) to obtain the final retextured result \( I(x,y) \) for each video frame:

\[ I(x,y) = C(x,y) \ast S(x,y) \]  \hspace{1cm} (20)

As shown in Fig. 4, the proposed texture coordinates optimization is effective for removing the visual artifacts such as the severe distortion phenomenon (Fig. 4(b)), thus, produces more natural visual appearance (Fig. 4(c)), our result has more consistent texture distortion of the shirt (Fig. 4(c)), while the texture distortion before optimization (Fig. 4(b)) is too heavy that makes the final re-texturing result look unnatural.

---

Fig. 4. Illustrating the importance of our texture coordinates optimization approach.
3. Experimental results and discussion

We have applied the proposed approach to a variety of videos, and the experimental results demonstrated that our re-texturing technique can generate visually pleasing results. All the examples shown in this paper are tested on a laptop with Pentium IV 2.2 GHz CPU + 2 GB RAM. We have mentioned some re-texturing results in the previous sections with Figs. 2 and 4. In Fig. 5, we present more video re-texturing results using our approach. Both the intrinsic video (Fig. 5(d)–(i)) and the video normals (Fig. 5(j)–(l)) in the re-textured region are recovered accurately. Our re-texturing results (Fig. 5(m)–(o)) are quite convincing without noticeable artifacts, where the lighting and shading effects are transferred properly.

In Fig. 6, we compare our algorithm with the previous approach [10]. Our result is visually better than the one by the mesh-guided method [10]. Because the shading and lighting effects are preserved by combining the Hue and Saturation components of the sample texture and Value component of the original image to the target re-textured region (Fig. 6(c)) in [10], the approach has the visual artifacts of color distortions. In contrast, our approach can preserve the

![Fig. 5. Re-texturing using intrinsic video: (a)–(c) the input video; (d)–(f) the recovered intrinsic video (reflectance component); (g)–(i) the recovered intrinsic video (illumination component); (j)–(l) the recovered normals; (m)–(o) the final re-texturing results. Note that the input sample texture is the inset in (m) (right top in last row).](image)
shading and lighting effects (green* and red lighting illumination) as well as the color faithfully in the to-be-replaced region (Fig. 6(d)).

In Fig. 7, we compare our algorithm with Textureshop [1]. Fig. 7(b) are directly taken from Textureshop [1], and Fig. 7(c) is our texture replacement result. The result generated by our approach is visually comparable to that of the cluster-based texture replacement method [1], and texture distortions are recovered more naturally in drapes of the sculpture in our result. However, the tedious user interactions are required to obtain the normal map for re-texturing in [1]. In contrast, our normal recovery process is fully-automatic by the Lambertian model and our energy optimization approach.

Fig. 6. The input image is shown in (a); (b) is the illumination component by our intrinsic image decomposition approach; (c) is the result by the approach in [10]; (d) is our result. Note that the input texture is the inset in (a) (right top).

Fig. 7. Comparison results. (a) Is the input image. The result by Textureshop [1] and our result are shown in (b) and (c) respectively. Note that the texture distortions are significantly reduced in our method. Image courtesy of Fang et al. [1].

* For interpretation of color in Fig. 6, the reader is referred to the web version of this article.

Please cite this article in press as: J. Shen et al., Re-texturing by intrinsic video, Inform. Sci. (2014), http://dx.doi.org/10.1016/j.ins.2014.02.134
4. Summary

This paper presents a novel re-texturing approach using intrinsic video. Our approach begins with extracting the region of interest by contour-aware layer segmentation. The intrinsic reflectance and illumination components for the segmentation region are recovered by our new weighted energy optimization. After that, we compute the normals for the re-textured regions using energy optimization. Finally, when the target sample texture is specified, the re-texturing video is created through multiplying the re-textured reflectance component by the original illumination component within the replaced region. As shown in our experimental results, our method produces high quality re-texturing results with a variety of sample textures, while the lighting and shading effects of the original videos are also well preserved.

Our approach mainly focuses on the video re-texturing with the uniform textures. The failure cases may occur when the input video contains people dressed in non-uniform colored textures. For such cases, we plan to incorporate the texture decomposition and classification algorithms [7,27] to extract the uniform regions from the non-uniform textures, and then apply the proposed video re-texturing method to the extracted uniform regions. Another interesting future work is to adopt graphics hardware acceleration [19] for real-time performance, and further investigate the extension to other applications in computational photography [14,22,23,25], such as relighting using intrinsic video and superpixel segmentation [28] techniques.

References