

# Interactive Weathering of Depth-Inferred Videos

Lin Xia

Zilong Dong\*

Guofeng Zhang

Duanjin Chen

Hujun Bao

State Key Laboratory of CAD&CG, Zhejiang University

Email: {xialin, zldong, zhangguofeng, bao}@cad.zju.edu.cn

## Abstract

Aging or weathering is an important technique for generating natural images in computer graphics. In this paper, we propose a novel video weathering method which can synthesize the weathering effects for the real captured videos. We first recover the depth maps of the input video, and employ a depth-based interactive video object cutout method to efficiently extract the target objects. Then, we sample the 3D point cloud from the recovered depth maps within the specified object mask, and reconstruct the 3D geometry model. In order to make the obtained 3D point cloud accurate with small redundancy, we select the pixels with the priority of depth accuracy and project each sample to other frames to avoid reselection. In the weathering stage, with the reconstructed 3D mesh, we employ  $\gamma$ -ton tracing to create the weathering map, and then compute the correspondence point for each object pixel on the weathering map by intersecting with the 3D mesh, which is used to resemble the weathering value to blend the texture with the original pixel color. We also introduce two interactive tools which allow the user to guide the process of weathering. A number of weathering effects are shown in the experimental results to demonstrate the robustness and efficiency of the proposed method.

**CR Categories:** I.4.6 [Image Processing and Computer Vision]: Segmentation—Pixel classification; I.4.8 [Image Processing and Computer Vision]: Scene Analysis—Depth cues; I.6.8 [Simulation and Modeling]: Types of Simulation—Visual

**Keywords:** segmentation, weathering, appearance modeling, texturing

## 1 Introduction

With the increasing prevalence of digital cameras, capturing videos become more and more convenient and videos can be easily accessed by the home-users in daily life. The rapid development of hardware necessitates the development of efficient and flexible video editing tools, allowing the home-users to conveniently create various visual effects. For example, it is common to make some objects dirty and rusty to simulate the aging or weathering phenomena. However, different to image processing, any modification of video content need to maintain the temporal consistency over frames to avoid jittering or flicker artifacts. For instance, if we modify the pixel colors in one frame, the corresponding pixels in other frames also need to be changed. Therefore, even a simple operation in image editing, such as inserting or removing one object, implementing the similar operation in video editing is much more challenging. In order to effectively solve the temporal coherence

problem in video editing, the accurate 3D geometry of the video scene is required.

In the past decades, many aging or weathering methods have been proposed [Chen et al. 2005; Mérillou and Ghazanfarpour 2008; Gu 2010]. However, these methods are limited to the virtual 3D scene/object, and can not be directly applied to the real scenes captured by videos. In this paper, we propose a novel interactive weathering framework for the depth-inferred videos. The key contribution is that we successfully propose a solution to simulate the weathering effect for real videos, and develop a user-friendly interface to allow the user to interactively guide the simulation for creating desired weathering effect. To the best of our knowledge, it has not yet been thoroughly discussed in literatures. In addition, we propose a novel depth-based interactive video object cutout method, which can robustly and efficiently extract static video objects. It is more robust than the traditional methods with motion estimation. In order to reliably reconstruct the 3D geometry model, we also propose a novel point cloud sampling method to fuse the depth maps which can robustly reject outliers and simultaneously control point redundancy. Then with the reconstructed 3D geometry model, we employ the  $\gamma$ -ton tracing method [Chen et al. 2005] to compute the weathering map, and map it to the video frames in a temporally consistent way. Two interactive tools are also provided to allow the user to conveniently design and control the weathering effects.

## 2 Related Work

As our system involves procedures of video segmentation, weathering and editing, we briefly review the related work in these areas.

### 2.1 Interactive Image/Video Segmentation

Several interactive video segmentation/matting techniques [Chuang et al. 2002; Wang et al. 2005; Li et al. 2005; Bai and Sapiro 2007; Bai et al. 2009; Price et al. 2009] have been developed. Especially, most of video segmentation methods [Chuang et al. 2002; Wang et al. 2005; Li et al. 2005] assume the background is known or can be easily modeled (e.g., using global GMM, clean plate or image mosaicing techniques). Recently, Bai et al. [2009] proposed a robust interactive video object cutout system, which can handle more general scenes. In their system, a set of local windows are constructed around the foreground boundary in the first frame. Each window has a local classifier to integrate the corresponding local color statistics and shape prior. By motion estimation, these local windows and classifiers are propagated to sequential frames for foreground segmentation. Price et al. [2009] proposed a similar foreground extraction framework, which combines kinds of cues (e.g., local color statistics, adaptively weighted color-adjacency, temporal and shape priors) into energy function, and uses graph-cut to solve it.

Perhaps due to the difficulty of depth recovery, all these methods do not use depth information to help segmentation. Recently, Zhang et al. [2010] proposed a new moving object extraction method, which integrates depth recovery, optical flow estimation and moving object segmentation into a unified framework and solve them iteratively. However, this method is rather time-consuming, which require a few minutes for processing one frame with a 4-core CPU.

---

\*Corresponding author: Zilong Dong

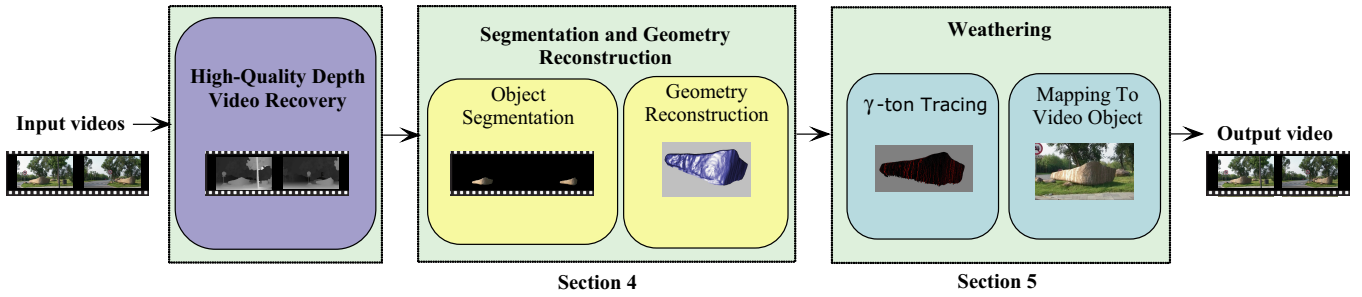


Figure 1: System overview.

## 2.2 Scene Weathering

Generating realistic natural images is one of the most important goals in computer graphics, and the aging or weathering simulation (e.g. add dust, patinas, cracks, rust, etc) on the surfaces of virtual objects usually plays a key role in realistic rendering. There are mainly two categories of techniques, measurement-based and generic models [Mérillou and Ghazanfarpour 2008]. The measurement-based techniques usually use elaborate equipments to capture the physically time-varying properties of materials [Lu et al. 2007], and re-simulate the process under the BRDF [Sun et al. 2007] or STDF [Gu 2010] models. Some image-based methods simplify the capture and analysis procedure based on the observation that the data over a surface captured at a single shot may represent different degrees of weathering [Wang et al. 2006; Xue et al. 2008]. Chen et al. [2005] proposed a generic model that works well for a large scale variations of weathering effect. They shoot and trace the  $\gamma$ -tons in the scene, and the local appearance is generated by linearly blending two texture images of the material according to different degree of weathering, whose weights are determined by  $\gamma$ -ton propagation and  $\gamma$ -transport. In this paper, we employ this method with some extension to synthesize weathering effect for real videos. The key extension is that our system allows the user to design and interactively guide the simulation (such as giving a sketch or an image). In contrast to the measurement-based approach, our weathering simulation system aims to generate desired visual effects rather than physically accurate simulation.

## 2.3 Video Editing

Different to image editing, the main challenging in video editing is how to maintain the temporal coherence while editing. Wang et al. [2004] proposed to treat the video as a space-time volume, and solve the problem by using a spatio-temporally consistent segmentation. Rav-Acha et al. [2008] proposed to unwrap the videos into 2D textures by tracking the point features of the objects, and any editing operation on the texture maps will project back to the original videos by a 2D-2D mapping. Most of the video editing methods maintain the temporal coherence by motion estimation and imposing temporal coherence constraint. If the 3D information of the video scenes can be recovered, video editing can be performed more straightforward and reliably [Kopf et al. 2008; Chen et al. 2011]. Both [Bhat et al. 2007] and [Zhang et al. 2009a] proposed to recover the depth maps by structure-from-motion and multi-view stereo techniques, and then use the recovered depths to generate kinds of visual effects (such as super-resolution, HDR, object insertion and removal, depth-of-field, fog synthesis, etc). However, to the best of our knowledge, how to add weathering effect to the real videos has not been discussed in previous work.

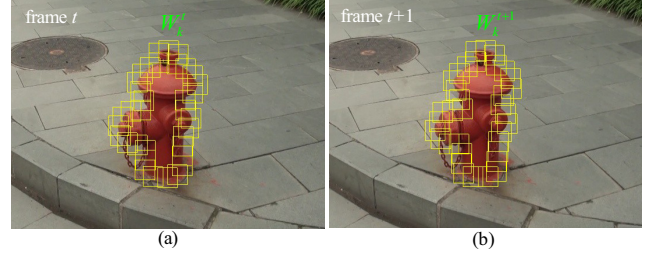


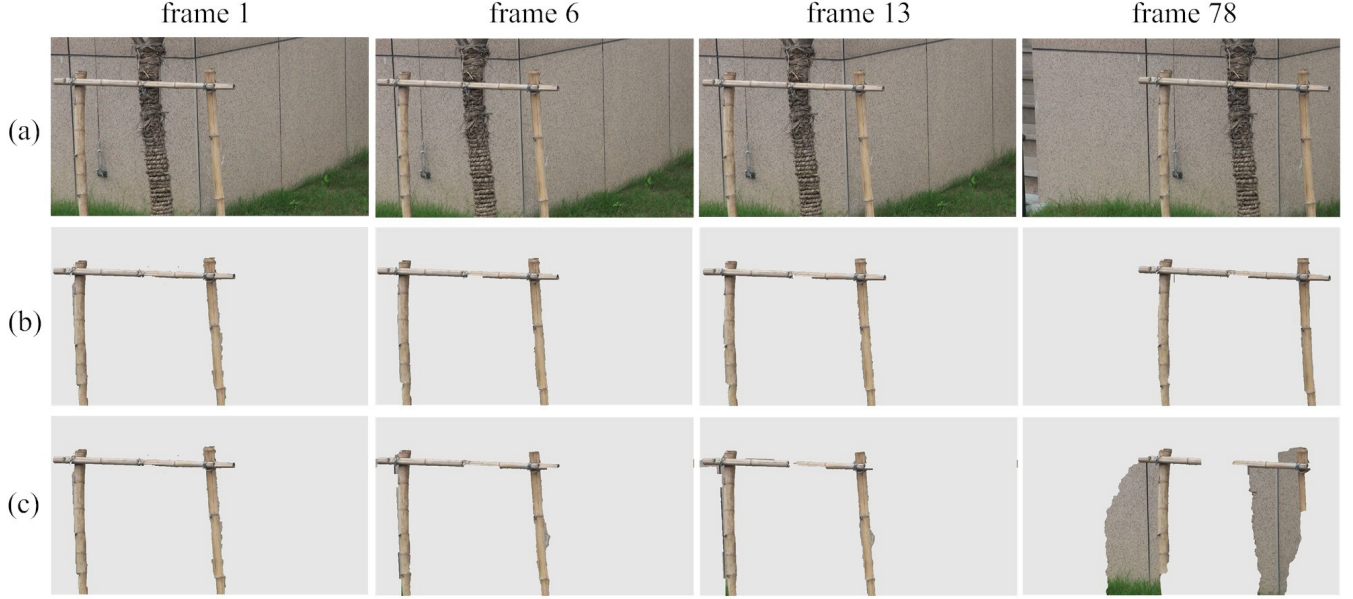
Figure 2: Local classifiers illustration. (a) Overlapping classifiers (yellow squares) are initialized along the object boundary (fireplug) in frame  $t$ . (b) The classifiers are propagated to the next frame by depth projection.

## 3 Framework Overview

Figure 1 illustrates an overview of our system. Given an input video sequence, we first employ the multi-view stereo method proposed in [Zhang et al. 2009b] to recover the depth map for each frame. Then we extract the interest object, and recover its 3D geometry. With the reconstructed 3D geometry, we synthesize the weathering effect for the specified object, such as making it dirty or rusty. Our system also allows the user to interactively guide the aging or weathering for creating the desired visual effect. The user can draw a guide sketch or simply provide an image, our system can automatically synthesize the desired visual effect.

## 4 Segmentation and Geometry Reconstruction

Before simulation of weathering, we first need to segment the interest object and recover its 3D geometry. We can use the interactive video segmentation technique to rapidly segment the interest object. Recently, Bai et al. [2009] proposed a fast interactive video object cutout system which combines a set of local classifiers, and uses the motion estimate to propagate them across frames to effectively cutout foreground object. However, motion estimation is not very reliable especially around discontinuous boundaries, which usually makes the propagation fail. Since we have recovered depth maps, we can utilize them to extract the video object more efficiently.



**Figure 3: Propagation comparison.** (a) The input video frames. (b) The propagation results of our method. (c) The propagation results of Snapcut. The top bar of the wooden frame is quite thin and has similar colors with the background wall, which is difficult for accurate motion estimation, so that Snapcut failed to propagate the local windows after 5 frames. In contrast, our method can successfully propagate over 70 frames.

#### 4.1 Depth-based Interactive Video Object Cutout

In order to solve the bilayer segmentation, we can define the following energy function:

$$E(\alpha) = \sum_{\mathbf{x}} (E_d(\alpha_{\mathbf{x}}) + w_{sh} E_{sh}(\alpha_{\mathbf{x}}) + w_s \sum_{\mathbf{y} \in N(\mathbf{x})} E_s(\alpha_{\mathbf{x}}, \alpha_{\mathbf{y}})) \quad (1)$$

where  $E_d(\alpha_{\mathbf{x}})$  is the data term, encoding the cost when the label of pixel  $\mathbf{x}$  is  $\alpha_{\mathbf{x}}$ .  $E_{sh}(\alpha_{\mathbf{x}})$  is the shape prior term, created from the last frame.  $E_s(\alpha_{\mathbf{x}}, \alpha_{\mathbf{y}})$  is the smoothness term, which penalizes the label difference between neighboring pixels.

Similar to [Bai et al. 2009], we uniformly sample a set of overlapping windows  $\{W_1^t, W_2^t, \dots\}$  around the foreground contour, as illustrated in Figure 2. Each window defines the applicability scope of a local classifier, where the foreground and background pixels are used to construct local GMMs. Thus the local color model in window  $W_k^t$  can be defined as follows:

$$E_d(\alpha_{\mathbf{x}}) = \begin{cases} -\log P^{W_k^t}(I_{\mathbf{x}}|F), & \alpha_{\mathbf{x}} = 1 \\ -\log P^{W_k^t}(I_{\mathbf{x}}|B), & \alpha_{\mathbf{x}} = 0 \end{cases} \quad (2)$$

where  $P^{W_k^t}(I_{\mathbf{x}}|F)$  and  $P^{W_k^t}(I_{\mathbf{x}}|B)$  are the probabilities computed by local GMMs.

Since we have recovered the depth maps and camera parameters, we can project the foreground mask to the next frame. The windows and alpha masks are naturally propagated to the next frame, and their corresponding local foreground and background GMMs can be re-estimated. The projected mask in frame  $t$  is denoted as  $\alpha_{t-1 \rightarrow t}$ , which is used to compute the shape prior. Then we can define the following shape prior similar to [Bai et al. 2009]:

$$E_{sh}(\alpha_{\mathbf{x}}) = w_{sh} \cdot \exp(-(dist(\mathbf{x})/\sigma_s^2)) \cdot |\alpha(\mathbf{x}) - \alpha_{t' \rightarrow t}(\mathbf{x})|, \quad (3)$$

where  $dist(\mathbf{x})$  stands for the distance from pixel  $\mathbf{x}$  to foreground boundary, and  $\sigma_s$  is a constant controlling the weight.

Figure 3 shows a comparison between our method and Snapcut [Bai et al. 2009]. As can be seen, for this example, the propagation result of Snapcut already contains notable segmentation errors after 5 frames. In comparison, our method uses depth information instead of motion estimate to propagate the local classifiers and estimate shape prior, which is much more robust and successfully propagates over 70 frames in this example.

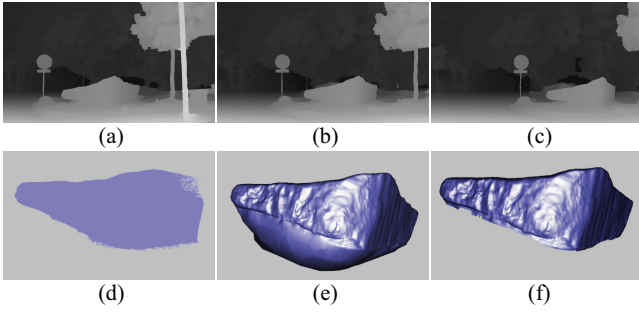
#### 4.2 Geometry Reconstruction

After obtaining the object mask, we need to generate the 3D point cloud from depth maps and use them to reconstruct the geometry model by Poisson Surface Reconstruction method [Kazhdan et al. 2006]. If we directly project the pixels in all frames to 3D space, the obtained 3D point cloud will contain large redundancy (i.e. multiple points almost have the same 3D positions) and lots of outliers. Therefore, we propose to optimize the sampling of point cloud. We would like to only select the pixels with accurate depths and the sampled point cloud has small redundancy. In order to achieve this, we need to measure the depth accuracy for each pixel.

It is difficult to precisely measure the depth accuracy without ground truth. Fortunately, we found that the depth accuracy can be reflected by projecting the pixels to the neighboring frames, and compare the photometric and geometric consistency. In general, if the consistency measure is large, the recovered depth should be accurate. Mathematically, the consistency measure can be defined by the following equation:

$$p(\mathbf{x}) = \exp(-\frac{\|I_f(\mathbf{x}) - I_{f'}(\mathbf{x}')\|^2}{2\sigma_c^2}) \exp(-\frac{\|D_f^{f \rightarrow f'}(\mathbf{x}') - D_{f'}(\mathbf{x}')\|^2}{2\sigma_d^2}), \quad (4)$$

where  $\mathbf{x}'$  is the projected pixel in frame  $f'$ , and  $D_f^{f \rightarrow f'}$  denotes the projected disparity map from frame  $f$  to  $f'$ .  $\sigma_c$  and  $\sigma_d$  are two standard deviations of color and depth, respectively. For each



**Figure 4:** Point cloud sampling and geometry reconstruction. (a-c) The depth maps of the video. (d) The sampled 3D point cloud (817,350 points). (e) The reconstructed 3D model by Poisson Surface Reconstruction method. (f) The 3D model after removing redundant triangles.

#### Algorithm 1 Point Cloud Sampling

1. Compute the consistency measure for all pixels in the segmented object.
2. Initialize  $V(\mathbf{x}) = 0$  for each pixel  $\mathbf{x}$ .
3. Sort the pixels with the estimated consistency measures.
  - for  $i = 1, \dots, m$ , %  $m$  is the number of pixels
  - If  $V(\mathbf{x}_i) < 2$ , add the 3D position of pixel  $\mathbf{x}_i$  into 3D point set  $\mathbf{C}$ , and project it to the neighboring frames.
  - $V(\mathbf{x}'_{f'}) += 2$ . %  $\mathbf{x}'_{f'}$  is the projected pixel in frame  $f'$ .
  - for each pixel  $\mathbf{y} \in W(\mathbf{x}'_{f'})$ , %  $W$  is a  $3 \times 3$  window
  - $V(\mathbf{y}) += 1$ .

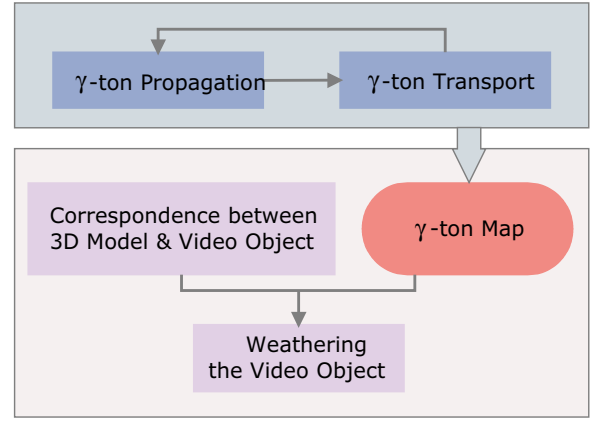
pixel, we can project it to the neighboring frames, and compute the consistency measure by Equation (4). In order to reduce the influence of occlusions, we sort the computed consistency measures in descending order  $\{p_1(\mathbf{x}), p_2(\mathbf{x}), \dots\}$ , and select 5 maximum ones, and compute the average to represent the depth accuracy:

$$p(\mathbf{x}) = \frac{1}{5} \sum_{i=1}^5 p_i(\mathbf{x}), \quad (5)$$

Then we use Algorithm 1 to sample the point cloud. For each pixel within the object mask, we can compute its depth accuracy by Equation (5). We sort the pixels with the consistency measure in descending order. For each pixel  $x$ , whose occupancy value  $V(x)$  is less than a threshold (e.g. 2), we select it to 3D point set  $\mathbf{C}$ , then project it to the neighboring frames. For the pixels within the small window of the projected point, their occupancies are increased. If the occupancy of one pixel is larger than the threshold, it will be not selected. This strategy can effectively control the redundancy of the sampled point cloud. Figure 4 shows the sampled point cloud and the reconstructed 3D geometry. If the object is not fully visible in the captured video, the reconstructed 3D point cloud also can not fully represent the object. However, Poisson Surface Reconstruction method [Kazhdan et al. 2006] always obtain a water-tight surface, even the provided point cloud is not complete. Thus, we need to remove the triangles which are introduced by Poisson Surface Reconstruction method but do not correspond to the pixels in the video. Figure 4(f) shows the geometry model after removing triangles.

## 5 Interactive Weathering of Videos

We employ a visual simulation of weathering technique called  $\gamma$ -ton tracing [Chen et al. 2005] to simulate the weathering effects for



**Figure 5:** Flowchart of  $\gamma$ -ton weathering.

real scenes. This technique can model a wide range of weathering effects, such as rust growing, stain or many other aging phenomena. To the best of our knowledge, existing systems that integrate this technique are only restricted to the virtual scenes, and weathering of real objects in videos have not been considered yet. In this paper, we adapt the  $\gamma$ -ton tracing and successfully synthesize the weathering effects for real videos by two passes processing.

In the first pass,  $\gamma$ -tons are emitted from  $\gamma$ -ton sources and traced through the object models in a way similar to photon tracing. The result of the first pass is a  $\gamma$ -ton map describing the  $\gamma$ -ton transport effects over all the surfels. From the  $\gamma$ -ton map, the second pass modifies the material properties of the mesh surfaces and produces the weathering effect to video objects by mapping the property changes to the video frames. The flowchart is summarized in Figure 5.

### 5.1 $\gamma$ -ton Propagation

At first, the  $\gamma$ -ton source is created as points, areas, or environment sources in space.  $\gamma$ -tons are shot from the source, and propagate through the scene, interacting with the geometry. The attributes of  $\gamma$ -ton can be defined as follows [Chen et al. 2005]:

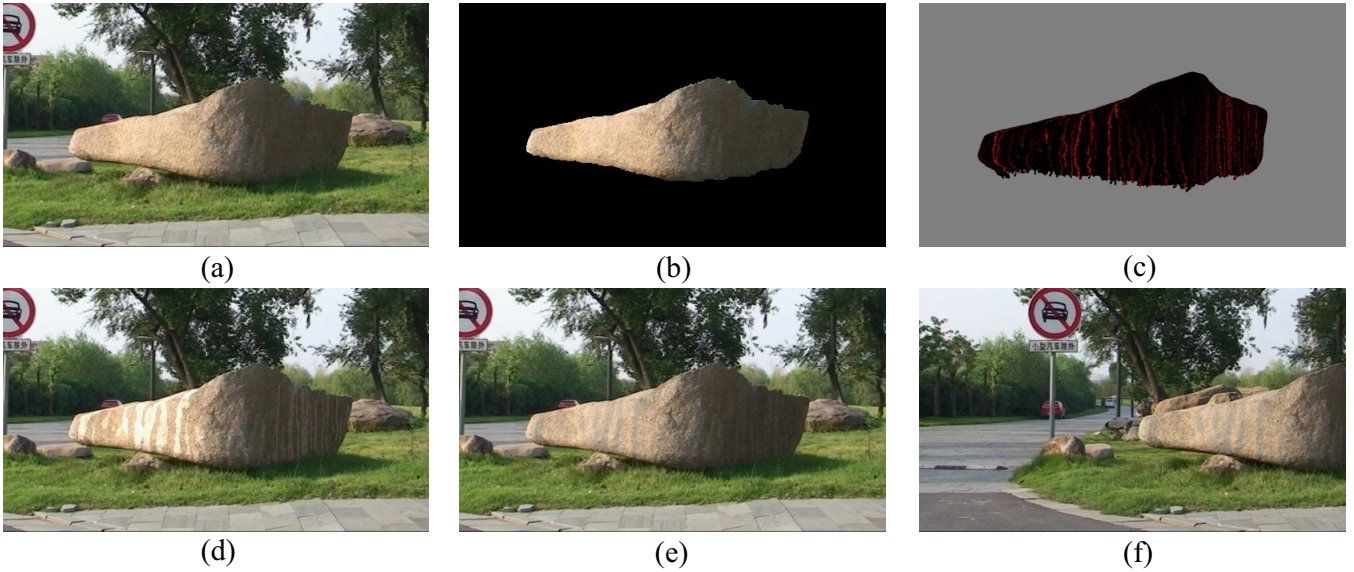
Ton {  
 Motion probabilities:  
 $k_s, k_p, k_f$  ;  
 Carrier attributes:  
 $s_h, s_d, \dots$   
 }

Each ton has two kinds of attributes, i.e. motion probabilities and carrier attributes. Motion probabilities decide its propagation behaviors while the carrier attributes store the substances the  $\gamma$ -ton represents. When  $\gamma$ -tons encounter the geometry, the next propagation operation is stochastically determined by the motion probabilities, which can only be one of the four motion types. Russian roulette [Arvo and Kirk 1990] is adopted to determine the next motion step. We use a uniformly distributed random variable  $\xi \in [0, 1]$  and make the following decisions:

$\xi \in [0, k_s]$	→ reflecting along a straight line
$\xi \in (k_s, k_s + k_p]$	→ bouncing along a parabolic curve
$\xi \in (k_s + k_p, k_s + k_p + k_f]$	→ flowing along the surface
$\xi \in (1 - k_s - k_p - k_f, 1]$	→ settling on the surface

So once shot,  $\gamma$ -ton keeps iteratively traveling in the first three ways until it settles on the surface. The carrier attributes of  $\gamma$ -ton store the





**Figure 6:** The  $\gamma$ -ton tracing and rendering result of “Rock” example. (a) The original video frame. (b) The extracted video object (rock). (c) The traced  $\gamma$ -ton map. (d) The computed result by directly mapping the  $\gamma$ -ton map to the video object. (e-f) The final blending results of weathering process.

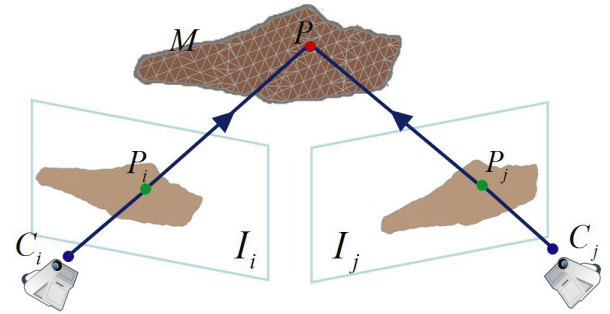
amount of substances in the real world that  $\gamma$ -ton represents, like  $s_h$  as heat, or  $s_d$  as dust. While  $\gamma$ -tons propagate through the scene, the ton’s attributes can affect the attributes of the surface material when it reaches the surface. The attributes can be freely designed by the user and the number of types is scalable.

## 5.2 $\gamma$ -ton Transport

To precisely record the weathering contributions from  $\gamma$ -tons and represent the local surface attributes, a point-based model is generated from the object mesh by resampling [Pfister et al. 2000]. At each surfel, it also contains two kinds of surface properties,  $\gamma$ -reflectance and material properties. The  $\gamma$ -reflectance affects the path of  $\gamma$ -ton propagations by modifying the motion attributions of  $\gamma$ -tons when tons are interacting with surfels. Material properties keep track of the blemish being modeled on the surface. The iterative nature of  $\gamma$ -ton tracing also allows us to change the  $\gamma$ -reflectance or material properties of the surfels using the previous  $\gamma$ -ton map’s material properties. For example, the more dust covers on the surface, the smaller bouncing chance the rain drops have. As  $\gamma$ -ton propagates through the scene, it picks up or deposits substances on the encountered surfel using its carrier attributes to change the surfel’s material properties, which is called  $\gamma$ -transport. The readers can refer to [Chen et al. 2005] for more details.

## 5.3 Weathering of Videos

After several iterations of  $\gamma$ -ton propagation and transport, we can get the distribution of blemish over the surface, and the  $\gamma$ -ton map as shown in Figure 6(c). With the  $\gamma$ -ton map, the weathering appearance of the video objects in every frame can be rendered in the second pass. Previous aging or weathering techniques only work with virtual 3D objects and could not directly weather the real objects in the captured videos. For weathering the video objects, we need to compute the correct correspondences between  $\gamma$ -ton map and the pixels in video objects. The synthesized weathering effect should maintain the temporal coherence among video frames. Here, we introduce an effective method to solve this problem.



**Figure 7:** Intersection and mapping. Since the pixels  $P_i$  in frame  $I_i$  and  $P_j$  in frame  $I_j$  correspond to the same scene point, their corresponding rays can intersect at the same 3D point  $P$  in the recovered 3D mesh.

We first use the bilayer segmentation method described in Section 4.1 to quickly cut out the specified video object. Then we sample the point cloud with the recovered depth maps and reconstruct the 3D geometry model using the method introduced in Section 4.2. As shown in Figure 7, the pixels  $P_i$  in frame  $I_i$  and  $P_j$  in frame  $I_j$  correspond to the same 3D point in the scene. Therefore, these two pixels should get the same value from the  $\gamma$ -ton map, so that the temporal coherence of weathering can be guaranteed.

For every pixel in the extracted video object, we project it to 3D space and intersect with the recovered 3D mesh. Once the intersection point is computed, the  $\gamma$ -ton map values are resembled by looking up the surfels in the point-based model, and then used to blend the texture color of the intersection point with the original pixel color. Both the mesh and the  $\gamma$ -ton map are organized by kd-tree, so that the intersection computation and map looking up can be significantly accelerated. With this method, the relations between surfels on the  $\gamma$ -ton map and the pixels in the video objects can be correctly built, so that weathering effects can be faithfully synthesized. Figure 6 shows a weathering example. Please refer to



**Figure 10:** Multiple weathering effects synthesis. Top row: three selected original frames. Bottom row: the weathering results. The hydrant becomes both dusty and rusty.



**Figure 8:** “Weathering Brush” example. (a) The brush painted locations are highlighted with white dots. (b) The synthesized mud-fluid effect on the hydrant.



**Figure 9:** “Weathering Watermark” example. (a) The specified “Transformer” mask on the wall. (b) The moss-growing result.

our supplementary video<sup>1</sup> for the complete results.

#### 5.4 User Control

In our system, the weathering effects are generated by the  $\gamma$ -ton tracing. The limitation of this technique is that it is hard for users to set the proper ton and surface attributes to get the desired results because it does not provide any interactive visual feedback. For example, since the shooting direction of the  $\gamma$ -ton is stochastically decided, in which way the natural substances do, the tons may shoot at the unnecessary places or there are not sufficient tons to reach the surface where the weathering effect is desired to take place. In this paper, we introduce two interactive weathering tools, which allow

the user to design the effects visually and freely on video objects.

As shown in Figure 8, the first tool, called “weathering brush”, can help locate the exact place where the user wants the weathering effects to take place. Figure 8(a) shows the “weathering brush” painting on the video object to specify the target surfaces. Using the method described in Section 5.3, for every painted pixel, we can find its corresponding 3D position on the  $\gamma$ -ton map. Figure 8(b) shows the brush effect to the hydrant after shooting and tracing certain mud-fluid behaved  $\gamma$ -tons according to the place that the “weathering brush” painted.

Another tool, namely “weathering watermark”, can be used to generate the effects guided by certain specified patterns, such as watermark, onto the video objects. Similar to the “weathering brush” tool, using the method introduced in Section 5.3, we can accelerate or constrain certain  $\gamma$ -ton’s attributes when they hit the surfaces that the watermark covers. Figure 9(a) shows the user-provided “Transformer” watermarks on the wall. We accelerate the moss growth’s speed in the specific area by adding more dirt attributes to the  $\gamma$ -tons when they encounter the watermark area. Figure 9(b) shows the results of “weathering watermark”, where the vivid moss marks are growing on the wall.

## 6 Experimental Results

We implemented our video weathering system on a PC with a Core2 Quad 2.6GHZ CPU and 4GB memory. The resolution of all test videos is 960×540. It takes about 1~3 minutes to synthesize the weathering effect for each frame, which depends on the number of vertices in the mesh and the size of video object.

Figures 6, 8 and 9 have shown the synthesized water stain, mud-fluid and moss-growing effects by our method, respectively. Our system also allows the user to simulate multiple weathering effects on a single video object. Figure 10 shows an example, which synthesizes multiple weathering effects on the hydrant (the hydrant becomes both dusty and rusty). The rust first takes places beneath the bonnet and nozzle caps where there is much humidity, and then expands along the surface. The dust is accumulated evenly on the hydrant and makes it dusty. Please refer to our supplementary video that gives a better presentation of the results.

<sup>1</sup>The supplementary video can be downloaded from the website <http://www.cad.zju.edu.cn/home/gfzhang/>

## 7 Conclusion

In this work, we propose an interactive weathering method for real captured videos. We first use a depth-based interactive segmentation method to quickly cut out the specified video object, and then reconstruct its 3D model by fusing the depth maps of different frames with a novel point sampling algorithm. With the reconstructed 3D model, the weathering effects can be synthesized in a temporally consistent way. Our system also provides two interactive weathering tools, which allows the user to specify the weathering area or tune the weathering speed to achieve the desired effects. With our system, a variety of weathering effects can be easily created by a home user.

Our system still has some limitations. First, our method is restricted to static scenes since the 3D models of dynamic objects can not be recovered by multi-view geometry. Second, sometimes it may require a significant time to set a suitable weathered texture for the specified video object and tune the weathering parameters. Hence, one of our future work is to extract the weathering textures from a large number of videos, learn their growing patterns, and finally build a weathering database. With this database, given a 3D geometry, the user could quickly select a suitable weathering texture and set the appropriate parameters.

## Acknowledgements

The authors would like to thank Hanqing Jiang for his great help in geometry reconstruction. Thanks to Qianling Li for his help to generate some results. This work is supported by the 973 program of China (No. 2009CB320802), and NSF of China (Nos. 60903135 and 61103104).

## References

- ARVO, J., AND KIRK, D. 1990. Particle transport and image synthesis. In *Proceedings of the 17th annual conference on Computer graphics and interactive techniques*, ACM, New York, NY, USA, SIGGRAPH '90, 63–66.
- BAI, X., AND SAPIRO, G. 2007. A geodesic framework for fast interactive image and video segmentation and matting. In *ICCV*.
- BAI, X., WANG, J., SIMONS, D., AND SAPIRO, G. 2009. Video snapcut: robust video object cutout using localized classifiers. *ACM Trans. Graph.* 28, 3.
- BHAT, P., ZITNICK, C. L., SNAVELY, N., AGARWALA, A., AGRAWALA, M., COHEN, M., CURLESS, B., AND KANG, S. B. 2007. Using photographs to enhance videos of a static scene. In *Eurographics Symposium on Rendering*.
- CHEN, Y., XIA, L., WONG, T.-T., TONG, X., BAO, H., GUO, B., AND SHUM, H.-Y. 2005. Visual simulation of weathering by -ton tracing. In *ACM SIGGRAPH 2005 Papers*, ACM, New York, NY, USA, SIGGRAPH '05, 1127–1133.
- CHEN, J., PARIS, S., WANG, J., MATUSIK, W., COHEN, M., AND DURAND, F. 2011. The video mesh: A data structure for image-based three-dimensional video editing. In *IEEE International Conference on Computational Photography (ICCP 2011)*.
- CHUANG, Y.-Y., AGARWALA, A., CURLESS, B., SALESIN, D., AND SZELISKI, R. 2002. Video matting of complex scenes. *ACM Trans. Graph.* 21, 3, 243–248.
- GU, J. 2010. *Measurement, modeling, and synthesis of time-varying appearance of natural phenomena*. PhD thesis, New York, NY, USA. AAI3420876.
- KAZHDAN, M. M., BOLITHO, M., AND HOPPE, H. 2006. Poisson surface reconstruction. In *Symposium on Geometry Processing*, 61–70.
- KOPF, J., NEUBERT, B., CHEN, B., COHEN, M., COHEN-OR, D., DEUSSEN, O., UYTENDAELE, M., AND LISCHINSKI, D. 2008. Deep photo: model-based photograph enhancement and viewing. *ACM Trans. Graph.* 27 (December), 116:1–116:10.
- LI, Y., SUN, J., AND SHUM, H.-Y. 2005. Video object cut and paste. *ACM Trans. Graph.* 24, 3, 595–600.
- LU, J., GEORGHIADES, A. S., GLASER, A., WU, H., WEI, L.-Y., GUO, B., DORSEY, J., AND RUSHMEIER, H. 2007. Context-aware textures. *ACM Trans. Graph.* 26 (January).
- MÉRILLOU, S., AND GHAZANFARPOUR, D. 2008. A survey of aging and weathering phenomena in computer graphics. *Computers & Graphics* 32, 2, 159–174.
- PFISTER, H., ZWICKER, M., VAN BAAR, J., AND GROSS, M. H. 2000. Surfels: surface elements as rendering primitives. In *SIGGRAPH*, 335–342.
- PRICE, B. L., PRICE, B. L., AND COHEN, S. 2009. Livecut: Learning-based interactive video segmentation by evaluation of multiple propagated cues. In *ICCV*, 779–786.
- RAV-ACHA, A., KOHLI, P., ROTHER, C., AND FITZGIBBON, A. 2008. Unwrap mosaics: a new representation for video editing. In *ACM SIGGRAPH 2008 papers*, ACM, New York, NY, USA, SIGGRAPH '08, 17:1–17:11.
- SUN, B., SUNKAVALLI, K., RAMAMOORTHY, R., BELHUMEUR, P. N., AND NAYAR, S. K. 2007. Time-varying brdfs. *IEEE Transactions on Visualization and Computer Graphics* 13 (May), 595–609.
- WANG, J., XU, Y., SHUM, H.-Y., AND COHEN, M. F. 2004. Video tooning. In *ACM SIGGRAPH 2004 Papers*, ACM, New York, NY, USA, SIGGRAPH '04, 574–583.
- WANG, J., BHAT, P., COLBURN, A., AGRAWALA, M., AND COHEN, M. F. 2005. Interactive video cutout. *ACM Trans. Graph.* 24, 3, 585–594.
- WANG, J., TONG, X., LIN, S., PAN, M., WANG, C., BAO, H., GUO, B., AND SHUM, H.-Y. 2006. Appearance manifolds for modeling time-variant appearance of materials. In *ACM SIGGRAPH 2006 Papers*, ACM, New York, NY, USA, SIGGRAPH '06, 754–761.
- XUE, S., WANG, J., TONG, X., DAI, Q., AND GUO, B. 2008. Image-based material weathering. *Computer Graphics Forum* 27, 617–626(10).
- ZHANG, G., DONG, Z., JIA, J., WAN, L., WONG, T.-T., AND BAO, H. 2009. Refilming with depth-inferred videos. *IEEE Transactions on Visualization and Computer Graphics* 15, 828–840.
- ZHANG, G., JIA, J., WONG, T.-T., AND BAO, H. 2009. Consistent depth maps recovery from a video sequence. *IEEE Trans. Pattern Anal. Mach. Intell.* 31, 6, 974–988.
- ZHANG, G., JIA, J., HUA, W., AND BAO, H. 2010. Robust bi-layer segmentation and motion/depth estimation with a handheld camera. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (Jun).