EUROGRAPHICS 2021 / N. Mitra and I. Viola
(Guest Editors)

Volume 40 (2021), Number 2

Spatiotemporal Texture Reconstruction for Dynamic Objects Using a Single RGB-D Camera

Hyomin Kim, Jungeon Kim, Hyeonseo Nam, Jaesik Park, and Seungyong Lee

POSTECH

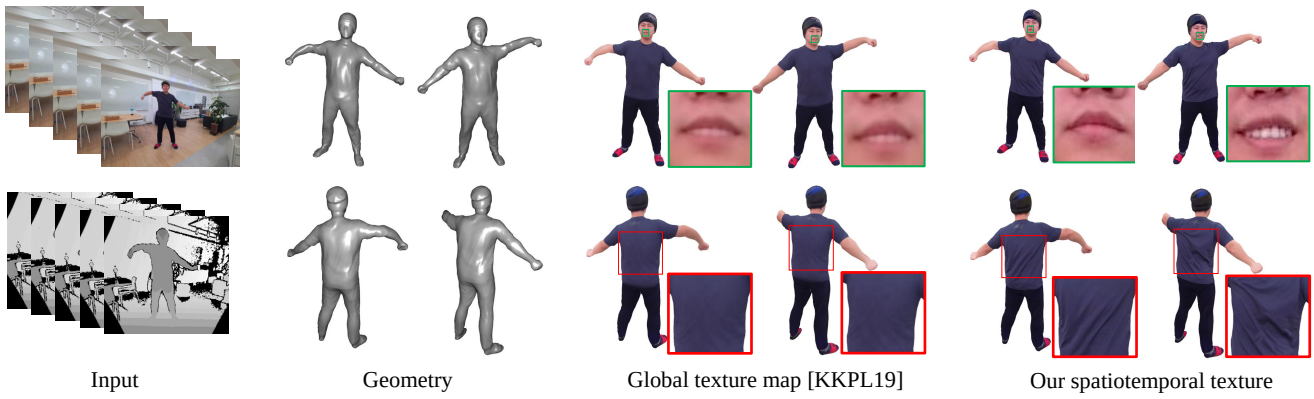

Figure 1: *Our approach reconstructs a time-varying (spatiotemporal) texture map for a dynamic object using partial observations obtained by a single RGB-D camera. The frontal and rear views (top and bottom rows) of the geometry at two frames are shown in the middle left. Compared to the global texture atlas-based approach [KKPL19], our method produces more appealing appearance changes of the object. Please see the supplementary video for better visualization of time-varying textures.*

Abstract

This paper presents an effective method for generating a spatiotemporal (time-varying) texture map for a dynamic object using a single RGB-D camera. The input of our framework is a 3D template model and an RGB-D image sequence. Since there are invisible areas of the object at a frame in a single-camera setup, textures of such areas need to be borrowed from other frames. We formulate the problem as an MRF optimization and define cost functions to reconstruct a plausible spatiotemporal texture for a dynamic object. Experimental results demonstrate that our spatiotemporal textures can reproduce the active appearances of captured objects better than approaches using a single texture map.

CCS Concepts

• ***Computing methodologies*** → ***Texturing;***

1. Introduction

3D reconstruction methods using RGB-D images have been developed for static scenes [NIH*11, NZIS13, DNZ*17] and dynamic objects [NFS15, IZN*16, DKD*16]. The appearance of the reconstructed 3D models, which are often represented as color meshes, is crucial in realistic content creation for Augmented Reality (AR) and Virtual Reality (VR) applications. Subsequent works have been proposed to improve the color quality of reconstructed static scenes [GWO*10, ZK14, JJKL16, FYY*18, LGY19] and dynamic objects [OERF*16, DCC*18, GLD*19]. For static scenes, most of the approaches generate a global texture atlas [WMG14, JJKL16, FYY*18, LGY19], and the main tasks consist of two parts: selecting the best texture patches from input color images and relieving visual artifacts such as texture misalignments and illumination differences when the selected texture patches are combined.

While single texture map is widely used to represent a static scene, it is not preferred for a dynamic object due to appearance changes over time. A multi-camera system could facilitate creating

© 2021 The Author(s)
Computer Graphics Forum © 2021 The Eurographics Association and John Wiley & Sons Ltd. Published by John Wiley & Sons Ltd.

a spatiotemporal (time-varying) texture map of a dynamic object, but such setup may not be affordable for ordinary users. As an alternative, recovering a spatiotemporal texture using a *single RGB-D camera* can be a practical solution.

Previous reconstruction methods using a single RGB-D camera maintain and update voxel colors in the canonical frame [NFS15, IZN*16, GXY*17, YGX*17, YZG*18]. However, this voxel color based approach cannot properly represent detailed time-varying appearances, e.g., cloth wrinkles and facial expression changes for capturing humans. Kim et al. [KKPL19] proposed a multi-scale optimization method to reconstruct a global texture map of a dynamic model under a single RGB-D camera setup. Still, the global texture map is static and cannot represent time-varying appearances of a dynamic object.

In this work, we propose a novel framework to generate a spatiotemporal texture map for a dynamic template model using a single RGB-D camera. Given a template model that is dynamically aligned with input RGB-D images, our main objective is to produce a spatiotemporal texture map that provides a time-varying texture of the model. Since a single camera can obtain only partial observations at each frame, we bring color information from other frames to complete a spatiotemporal texture map. We formulate a Markov random field (MRF) energy, where each node in the MRF indicates a triangular face of the dynamic template model at a frame. By minimizing the energy, for every frame, an optimal image frame to be used for texture mapping each face can be determined. Recovering high-quality textures is another goal of this work because it is crucial for free-viewpoint video generation of a dynamic model. Due to imperfect motion tracking, texture drifts may happen when input color images are mapped onto the dynamic template model. To resolve the problem, we find optimal texture coordinates for the template model on all input color frames. As a result, our framework can handle time-varying appearances of dynamic objects better than the previous approach [KKPL19] based on a single static texture map.

The key contributions are summarized as follows:

- A novel framework to generate a stream of texture maps for a dynamic template model from color images captured by a single RGB-D camera.
- An effective MRF formulation including selective weighting to compose high-quality time-varying texture maps from single view observations.
- An accelerated approach to optimize texture coordinates for resolving texture drifts on a dynamic template model.
- Quantitative evaluation and user study to measure the quality of reconstructed time-varying texture maps.

2. Related Work

2.1. Dynamic Object Reconstruction

3D reconstruction systems for dynamic objects can be classified into two categories: template-based and templateless. A representative templateless work is DynamicFusion [NFS15], which shows real-time non-rigid reconstruction using a single depth camera. VolumeDeform [IZN*16] considers a correspondence term made from SIFT feature matching to improve motion tracking. BodyFusion [YGX*17], which specializes in human body, exploits human skeleton information for surface tracking and fusion.

Template-based non-rigid reconstruction assumes that a template model for the target object is given, and deforms the template model to be aligned with input data. Most template-based methods [LAGP09, ZNI*14, GXW*15, GXW*18] are based on embedded deformation (ED) graph model [SSP07], which is also utilized in templateless works [LVG*13, DTF*15, DKD*16, DDF*17]. High-end reconstruction systems using a number of cameras can capture a complete mesh of the target object at each timestamp, so they commonly use template tracking between keyframes for compressing redundant data [CCS*15, PKC*17, GLD*19].

In this paper, we use a template-based method [GXW*15, GXW*18] to track the motion of a dynamic object, while any method can be used if it can generate a sequence of meshes with a fixed connectivity conforming a RGB-D image sequence.

2.2. Appearances of Reconstructed Models

Static scenes When scanning a static scene with a single RGB-D camera, color information obtained by volumetric fusion tends to be blurry due to inaccurate camera pose estimation. Gal et al. [GWO*10] optimize projective mapping from each face of a mesh onto one of the color images by performing combinatorial optimization on image labels and adjusting 2D coordinates of projected triangles. Zhou and Koltun [ZK14] jointly optimize camera poses, non-rigid correction functions, and vertex colors to increase the photometric consistency. Jeon et al. [JJKL16] optimize texture coordinates projected on input color images to resolve misalignments of textures. Bi et al. [BKR17] propose patch-based optimization to align images against large misalignments between frames. Fu et al. [FYY*18] and Li et al. [LGY19] improve the color of a reconstructed mesh by finding an optimal image label for each face. Fu et al. [FYLX20] jointly optimize texture and geometry for better correction of textures. Huang et al. [HTD*20] exploit a deep learning approach to reconstruct realistic textures.

Non-rigid objects with multiple cameras Non-rigid object reconstruction has received attention in the context of performance capture using multi-view camera setup [CCS*15, PKC*17, DKD*16, OERF*16]. Fusion4D [DKD*16] is a real-time performance capture system using multiple cameras, and it shows the capability to reconstruct challenging non-rigid scenes. However, it uses volumetrically fused vertex colors to represent the appearance, resulting in blurred colors of the reconstructed model. To resolve the problem, Holoportation [OERF*16] proposes several heuristics, including color majority voting. Montage4D [DCC*18] considers misregistration and occlusion seams to obtain smooth texture weight fields. Offline multi-view performance capture systems [CCS*15, GLD*19] generate a high-quality texture atlas at each timestamp using mesh parameterization, and temporally coherent atlas parameterization [PKC*17] was proposed to increase the compression ratio of a time-varying texture map.

Non-rigid scenes with a single camera Kim et al. [KKPL19] produces a global texture map for a dynamic object by optimizing

© 2021 The Author(s)
Computer Graphics Forum © 2021 The Eurographics Association and John Wiley & Sons Ltd.

Figure 2: System overview. Given a template model and a RGB-D image stream, we first reconstruct a dynamic object. We then optimize texture coordinates to resolve texture drifts among frames. Next, we calculate the geometric similarity and temporal texture variation, and formulate an MRF energy. By minimizing the MRF energy, frame labels are assigned to each face so that a high-quality spatiotemporal texture volume can constituted. As a post-processing, color correction resolves intensity differences among texture patches.

texture coordinates on the input color images in a single RGB-D camera setup. Pandey et al. [PTY19] propose semi-parametric learning to synthesize images from arbitrary novel viewpoints. PIFu [SHN 19] combines learning-based 3D reconstruction, volumetric deformation, and light-weight bundle adjustment to reconstruct a clothed human model in a few seconds from a single RGB-D stream. These methods produce either a single texture map for the reconstructed dynamic model [KKPL19, SHN19] or temporally independent blurry color images for novel viewpoints [PTY 19]. They all cannot generate a sequence of textures varying with motions of the reconstructed model over time. Our framework aims to produce a visually plausible spatiotemporal texture for a dynamic object with a single RGB-D camera.

3. Problem Definition

3.1. Spatiotemporal Texture Volume

Given a triangular mesh M of the template model, we assume that a non-rigid deformation function Y_t is given for each time frame t, satisfying $Y_t(M) = M_t$, where M_t denotes the deformed template mesh fitted to the depth map D_t. Our goal is to produce a spatiotemporal (time-varying) texture map T for the dynamic template mesh. By embedding M onto a 2D plane using a parameterization method [Mic11], each triangle face f of M is mapped onto a triangle in the 2D texture domain. A spatiotemporal texture map can be represented as stacked texture maps (or spatiotemporal texture volume), whose size is $W \times W \times T$, where W is the side length of a rectangular texture map and T is the number of input frames. We assume that our mesh M_t keeps a fixed topology in the deformation sequence, and the position of face f in the 2D texture domain remains the same regardless of time. Only the texture information of face f changes with time t to depict dynamically varying object appearances.

3.2. Objectives

To determine the texture information for face f of the deformed mesh M_t at time t in a texture volume, we need to solve two problems: color image selection and texture coordinate determination. We should first determine which color image in the input will be used for extracting the texture information. A viable option is to use the color image C_t at time t. However, depending on the camera motion, face f may not be visible in C_t, and in that case, texture should be borrowed from another color image $C_{t'}$ where the face is visible. Moreover, when there are multiple such images, the quality of input images should be considered, as some images can provide sharper textures than others.

Once we have selected the color image $C_{t'}$, we need to determine the mapped position of f in $C_{t'}$. Since we already have the deformed mesh $M_{t'}$ at time t', the mapping of f onto $C_{t'}$ can be obtained by projecting face f in $M_{t'}$ onto $C_{t'}$. However, due to the errors in camera tracking and non-rigid registration, mappings of f onto different input color images may not be completely aligned. If we use such mappings for texture sampling, the temporal texture volume would contain little jitters in the part corresponding to f.

An overall pipeline of our system is shown in Figure 2. Starting from RGB-D images and a deforming model, we determine the aligned texture coordinates using global texture optimization (Section 4). Then, to select color images for each face f at all timestamps, we utilize MRF optimization (Section 5). To reduce redundant calculation, we pre-compute and tabulate the measurements that are required to construct the cost function before MRF optimization. Finally, we build a spatiotemporal texture volume and refine textures by conducting post-processing.

4. Spatiotemporal Texture Coordinate Optimization

Camera tracking and calibration errors induce misalignments among textures taken from different frames. Non-rigid registration

© 2021 The Author(s)
Computer Graphics Forum © 2021 The Eurographics Association and John Wiley & Sons Ltd.