LITERATURE REVIEW II

Name : Shwetambara Narayan Patil

UML ID : 01666303

E-mail : ShwetambaraNarayan_Patil@student.uml.edu

Date : 03/21/2018

PRIMARY PAPER

Real-Time View Correction for Mobile Devices

@ARTICLE{8007219,
author={T. Schöps and M. R. Oswald and P. Speciale and S. Yang and M. Pollefeys},
journal={IEEE Transactions on Visualization and Computer Graphics},
title={Real-Time View Correction for Mobile Devices},
year={2017},
volume={23},
number={11},
pages={2455-2462},
keywords={augmented reality;calibration;cameras;image colour analysis;image sensors;image sequences;rendering (computer graphics);RGB-D;augmented reality applications;depth discontinuities;image gradients;mobile devices;real-time view correction system;screen-camera calibration;virtual camera views;virtual reality applications;Augmented reality;Cameras;Image color analysis;Interpolation;Mobile handsets;Real-time systems;Rendering (computer graphics);Augmented Reality (AR);Depth Image Based Rendering (DIBR);Mobile Devices;View Correction},
doi={10.1109/TVCG.2017.2734578},
ISSN={1077-2626},
month={Nov},}

SECONDARY PAPER

SUPERPIXEL-BASED DEPTH MAP INPAINTING FOR RGB-D VIEW SYNTHESIS

@INPROCEEDINGS{7351624,
author={P. Buyssens and M. Daisy and D. Tschumperlé and O. Lézoray},
booktitle={2015 IEEE International Conference on Image Processing (ICIP)},
title={Superpixel-based depth map inpainting for RGB-D view synthesis},

year={2015},
volume={},
number={},
pages={4332-4336},
keywords={image colour analysis;image restoration;image segmentation;RGB-D scenes;RGB-D view synthesis;occlusion situations;superpixel oversegmentation;superpixel-based depth map inpainting;virtual view synthesis;Cameras;Complexity theory;Image color analysis;Image reconstruction;Indexes;Rendering (computer graphics);Robustness;Depth map disocclusion;View synthesis;superpixels},
doi={10.1109/ICIP.2015.7351624},
ISSN={},
month={Sept},}

REVIEW

The primary paper here gives a method for rendering novel virtual camera views from given RGB-D (color and depth) data of a different viewpoint. Incomplete inputs or disocclusions resulting in missing colors or depth information are inpainted in a consistent way. The location of strong image gradients is used as a reference for depth discontinuities. The method is also adaptable to mobile devices and explains how to obtain a screen-camera calibration and the options for obtaining depth input.

AR applications on mobile devices put their content over the device's camera image directly, which they display on the screen. This causes a difference between the image and user's perspective on the scene. This affects the user experience and may be an obstacle for these applications' usage. To implement their method, the authors studied previous works that dealt with viewpoint interpolation, image synthesis, depth image based rendering, and depth inpainting.

The authors have referred the secondary paper which came up with a method to inpaint the holes in depth maps that are formed while creating virtual views from RGB-D scenarios. This method overcomes the many occlusion situations by using superpixel oversegmentation of both the original and synthesized views. The superpixels efficiently deal with inaccurate depth maps. The secondary paper states that problem arises when background areas that are hidden (and not known) by a foreground object in the original view may have to be rendered in the synthesized view. Disocclusion refers to filling these holes.

The secondary paper proposed the following algorithm :

- compute superpixel oversegmentations of the original and synthesized views, and the correspondence between both their superpixels,
- for each pixel to inpaint, find a set of candidates superpixels candidates in the original view,

- modelize the remaining superpixels by planes, and infer the depth value of the pixel by a linear combination of these planes.

The secondary paper highlights the major problem that is caused when an occlusion hole that has to be filled with background is only surrounded with foreground. It focuses on the inpainting of occlusions that appeared in depth maps after a warping process. The main idea here was to be able to identify the FG and BG depth values to use to fill holes. In the primary paper, as the authors inpaint disocclusions, the work done in secondary paper was of greater significance. The authors used a single image and a potentially incomplete depth map, both recorded at the same source camera view, as an input for the view correction. The aim of this primary paper was to build a complete virtual image and a corresponding complete depth map as they would have been seen from a different camera view (target). Their 7 step process consists of following steps :

- Compute inpainting weights from input image
- Inpaint the incomplete depth map to get a complete depth map.
- Reproject the complete depth map to target to get a warped depth map with disocclusions, and warped inpainting weights.
- Reproject input image to target using warped map to get an incomplete warped image.
- Reproject depth map and image from the previous iteration of the pipeline to warped depth map and image to achieve temporal consistency.
- Inpaint warped depth map to get the complete depth map.
- Inpaint warped image to get the complete image.

As it is clear from this algorithm that Inpainting is a very crucial step in this method, the inpainting method proposed in the secondary paper using superpixel oversegmentations of the original and synthesized views, is of great importance. As warping may not be accurate, warping a single pixel adds more errors to it. The secondary paper addresses how superpixels provide more robustness in warping, which turns beneficial in this algorithm. The authors also discussed different types of inpainting like Fast Inpainting & Convolution-based Inpainting. The authors also shed light on how the algorithm they developed can be implemented for mobile devices. This system integration consists of following steps :

- Depth Map Acquisition

  This process consists of following factors :

  1. Camera Pose Estimation
     To determine the camera poses of the mobile device in realtime, authors used the motion tracking ability of Google Tango. The use of accelerometer measurements in this approach made the absolute scale of the trajectory observable.

  2. Depth from Sensor

The proposed method can directly use the depth maps obtained from a depth sensor, even if they are sparse. Using this direct type of depth map acquisition, i.e., without temporal integration, has the advantage that no outdated information is used.

3. Depth from 3D Reconstruction
   As an alternative to directly using depth sensor readings, the authors perform 3D reconstruction to accumulate the raw sensor estimates to gain more complete and accurate information. They have used the 3D reconstruction implementation of Google Tango for this task.

- Observer Position

  This process consists of following factors:

  1. Screen-Camera Calibration
     In this step, authors estimated the extrinsic calibration between the screen and the rear camera of a mobile device.

  2. Eye tracking
     The paper states that the user's eye positions must be known to determine the target frame for the view correction. For the scenario of a device whose screen moves relative to the user's eyes, the user's eyes need to be tracked. In order to simulate this scenario and for evaluation purposes, authors used a second mobile device which observes the first one.

Given sufficient depth information nearby, the primary paper is able to fill in missing regions and correctly handle them. The results show how virtual objects can get occluded by real objects in an Augmented Reality use case for the scenario of a Virtual Reality headset, using the inpainted target frame depth map for occlusion testing.

Since the authors had the corresponding input image, the warped target image was already known. Hence, they calculated the average pixel-wise error between the view correction result and the ground truth image over an entire sequence. They did not enforce temporal consistency for this experiment. The evaluation has been limited to pixels which are farther away than 40 pixels from the image border. This is to avoid comparing border regions where there is very little or no adjacent color and depth information which can be used for inpainting, such that it would not be fair to evaluate these regions. The paper states that on the tablet, the method took a maximum of about 100 ms for one frame, however it often took less than 30 ms if sufficient depth data from the sensor or the previous frame was available.

The paper also discussed several limitations like unable to use the front and rear camera of the mobile simultaneously. Since the camera uses a low resolution, the results are of a relatively low output resolution, and the hardware is not suitable for observing fastmoving objects. Also, as the inpainting algorithm only diffuses colors which creates areas of relatively constant color, inpainted regions will be over-smooth and appear less plausible in strongly textured scenes.

The authors conclude that the core algorithm of processing pipeline is a diffusion-based inpainting algorithm that favors depth discontinuities at possible object edges, which are usually indicated by color or brightness changes in the input image. They also ensured that the inpainting returns temporally consistent results, which is important to avoid flickering artifacts. Hence, this pipeline method can yield pleasing results at real-time frame rates on current mobile devices.