Enhancement of 3D Capture of Room-Sized Dynamic Scenes with Pan-Tilt-Zoom Cameras

Asad Ullah Naweed, Lu Chen, Mingsong Dou, and Henry Fuchs

University of North Carolina at Chapel Hill

Abstract. We present a method for augmenting the 3D reconstruction of a dynamic indoor scene using Pan-Tilt-Zoom cameras. The system combines Pan-Tilt-Zoom cameras with static wide field-of-view depth cameras within a networked platform. Traditionally, Pan-Tilt-Zoom cameras have been extensively used in surveillance applications, since their ability to pan, tilt and zoom in on an object allows them to cover a large area with a reduced number of cameras. However, most of the existing work with PTZ cameras deals with scanning or tracking objects in large outdoor environments, where objects are typically large distances away from the cameras. We use PTZ cameras in an indoor setting to zoom in on relevant and interesting objects to get fine visual details. The fine details and high resolution imagery enables us to augment and refine the room's 3D surface as constructed from off-the-shelf depth cameras statically mounted around the room. We show significant improvements in both texture quality and geometry when high-resolution imagery from multiple PTZ cameras is used to supplement the 3D model built from fixed commodity depth cameras.

1 Introduction

Obtaining 3D reconstruction of a dynamic room-sized indoor scene is of great use in many applications. A few examples of such applications are 3D telepresence, virtual and augmented reality systems, 3D animation and motion capture systems. A dynamic environment poses significant challenges: not only are we concerned with the motion of rigid objects, such as furniture and other props, but we also have to account for objects that move in a non-rigid fashion and also deform, e.g. human beings. In recent years, the proliferation of commodity depth cameras, such as the Microsoft Kinect, has opened up new avenues for development of a suitable 3D scanning system for an indoor room-sized environment. These cameras are favored due to their low cost and reasonable degree of accuracy. However, the current state-of-the-art reconstruction using just these depth cameras is still a long way from achieving the level of detail that is required in certain domains, most notably 3D telepresence.

In this paper, we present a method to refine and improve such a dynamic 3D reconstruction of a room-sized indoor environment by using narrow field-of-view Pan-Tilt-Zoom cameras. The coarse 3D reconstruction is obtained by means of multiple static wide field-of-view Kinect cameras, and the refinement is brought

G. Bebis et al. (Eds.): ISVC 2014, Part I, LNCS 8887, pp. 379–389, 2014.

about by incorporating high resolution imagery acquired from multiple PTZ cameras in various stages.

Objects in the environment can be broadly classified into one of the following three categories, as stated by Dou and Fuchs in [1]:

1. Static Background
2. Rigid Semi-static object
3. Dynamic non-rigid objects

The walls, ceiling and any large and heavy items can be assumed to belong to the first category, since they will not move, deform or change their appearance for the duration of the capture. Objects such as chairs and small props belong to the second category, since they can be moved, but will only deform rigidly. Human actors in the scene belong to the third category, since not only do they move, but also deform non-rigidly from one time instant to another. In our system, each of these categories is treated differently in order to obtain the maximum amount of information required to reconstruct a scene with a large amount of detail. We will explore how the high-resolution imagery from variable field-of-view cameras is used to enhance the 3D reconstruction of static, semi-static and dynamic objects.

This endeavour presents several technical challenges. First, the problem of segmenting out semi-static object from completely static objects in a fully automated way is a hard problem, and making this decision efficiently would greatly help in deciding how to treat different objects and areas in the scene and assign PTZ cameras to dynamic regions rather than static ones. 2D-3D registration is also a significant problem, since 2D features cannot be trivially matched against 3D features. This has a direct bearing on camera pose estimation, which is a major component of our algorithm. Inter-sensor calibration between cameras with different focal lengths is non-trivial. Finally, there is the problem of the entire process being too computationally expensive to be performed in real-time, though there have been some attempts to perform such tasks on distributed clusters and GPUs.

The remainder of this paper is structured as follows. Section 2 provides an overview of current and relevant literature in the field. Section 3 gives an overview of our system and pipeline developed to run the algorithms defined in subsequent sections. The core contributions of this work are twofold, and described in Sections 4 and 5 respectively. The first contribution is registration of high-resolution images onto a 3D mesh model with per-vertex color values, and then synthesis of textures to improve visual quality of the reconstruction. The main algorithm and results for this part are outlined in Section 4. The second contribution is stereo reconstruction of dynamic objects using only high resolution imagery from the PTZ cameras [2], and then fusing this high quality reconstruction into the coarse model reconstructed using only commodity depth cameras. The algorithm and results for this part are outlined in Section 5. Finally, Section 6 concludes the paper, explains limitations and highlights possible avenues for future work.

2 Related Work

2.1 3D Geometry Reconstruction

Dou and Fuchs [1] have developed a surface tracking algorithm that refines the surfaces of dynamic objects over time using data acquired from multiple static Kinect cameras. Their work also handles inter-surface penetrations appropriately and fills in holes in the surfaces by accumulating temporal information from the movement of the dynamic object. However, fixed cameras are not suitable for capturing fine geometric features in dynamicly deforming and moving regions, such as those commonly found on human faces. Therefore, their system does not do a well enough job of reconstructing high reslution geometry, which is a significant limitation. These details are very valuable in 3D telepresence applications, since human faces are areas of great interest in such scenarios, and will get a lot of attention. The system also behaves erratically when dynamic objects do not move around much, since the hole-filling algorithm greatly depends on obtaining multiple projective views of the deforming object over time. Davis [3] presents a scheme for multi-scale motion recovery by means of multiple cameras and careful camera assignment to visual targets. However, this scheme utilizes LED markers bound to objects of interest, and so is not suitable for our applications. Beck et al [4] showcase a 3D immersive telepresence system, but their technique does not use temporal coherence between consecutive frames to improve the quality of the reconstruction. Matsuyama et al [5] present a real-time method for 3D shape reconstruction and mesh deformation on a cluster of PCs for the purposes of recording high-fidelity 3D video. Yous et al [2] present a resource assignment scheme to control multiple Pan-Tilt cameras for the purpose of obtaining a 3D video of a moving object. However, they only present a camera assignment scheme, and do not present an algorithm to compute detailed meshes of the observed object. They propose photometric consistency based space-carving [6] and deformable mesh models [5] as appropriate algorithms for generating the 3D structure of the object.

2.2 Pan-Tilt-Zoom Cameras

Ilie and Welch [7] propose a PTZ camera assignment scheme to track subjects undergoing physical training. However, their approach is suitable for large outdoor environments, and does not deal with obtaining fine details about objects in indoor environments. Wan and Zhou [8] present a technique for stereo vision from a PTZ camera-pair which uses a form of spherical rectification and epipolar geometry to cater for the varying zoom scale of the two cameras. Sinha and Pollefeys [9] survey techniques for calibration of a network of PTZ cameras, and also provide a novel technique based on feature-based alignment and bundle adjustment of images acquired by a rotating PTZ camera. This technique was then applied to generating multi-resolution giga-pixel panoramas from PTZ cameras [10]. There are also a number of other systems which utilizes the motion of PTZ cameras, such as [3] and [11].

3 System Overview

The camera network consists of 10 static Microsoft Kinect sensors mounted at various points in the room, and three Axis 233D Pan-Tilt-Zoom cameras mounted in a horizontal row at the front of the room. The Kinect cameras are connected to a PC via dedicated USB ports, and the PTZ cameras are IP cameras that are connected to the PC by a dedicated ethernet hub. The IP cameras can be manipulated and queried for images, pan-tilt angles and zoom scales via HTTP requests. The image acquisition system employed a self-clocking barrier synchronization scheme to make sure that the frames acquired from PTZ cameras and the Kinect cameras were closely synchronized. The main improvements in the reconstruction are brought about by a two-phase process, as described below.

The first phase is designed to acquire as much information about the static background as possible in order to achieve the best possible reconstruction quality for those objects. The data acquisition for this step is accomplished by moving a single Kinect in an otherwise static room and a PTZ camera set at a static high zoom scale and varying pan-tilt angles. The Kinect provides a sequence of RGB-D frames, and the PTZ camera provides a RGB image for every pan-tilt angle pair. A post-processing step corrects the Kinect images for depth bias [1]. The data collected in this phase is called the "pre-scan" data. Further details are given in Section 4.

The goal of the second phase is to collect information about dynamic objects in the scene. Data acquisition for this phase is done by means of multiple Kinects mounted on the walls of the room, and multiple PTZ cameras focused on one or more areas of interest. These areas are regions which humans are naturally sensitive to in telepresence scenarios, e.g. human faces and readable text. This second phase is called the "live session". More details on this are present in Section 5.

4 Augmenting Static Object Reconstruction with High Quality Textures

High-quality textures are synthesized for parts of the model which form the static background. Since these objects do not move or change their appearance for the duration of the capture, we can perform a pre-scan of the entire room prior to the actual capture session. All the RGB-D frames from the single moving Kinect are aligned using a technique similar to that used in [12], which is based on bundle adjustment and feature-based alignment. Once the images are aligned, a volumetric depth map fusion yields a consolidated point cloud, which is then triangulated using a marching cube technique [13]. The surface resulting from this step is shown in Figure 1. We let this surface be called S_{Kinect}.

The model generated from this step yields only a single color per vertex, and at most 1 vertex per centimeter cube, so visual quality is poor, especially in regions with fine or detailed texture, human-readable text and other areas humans are naturally sensitive to. To improve the visual quality of the reconstruction,

Fig. 1. The 3D surface of the static background using only RGB-D frames from a single moving Kinect. The surface is a color-by-vertex model and lacks detail.

we use the images acquired from the PTZ cameras during the pre-scan to generate high-resolution textures for the reconstructed model. The exact procedure is detailed below.

First, for each image acquired from the moving Kinect during the pre-scan, we compute a set of visual SIFT features [14]. Let I_{Kinect} the set of RGB-D frames acquired from the Kinect during the pre-scan. Then, for each $i \in I_{Kinect}$, we compute a set of visual SIFT features. We let this set be F_i. Each $f \in F_i$ has, by the nature of its computation, a keypoint location (a 2-element vector) and a feature descriptor (a 128-element vector) associated with it. Let these be called $\mathbf{x}_f$ and D_f respectively.

Then, for each such f, we compute and store the following:

- The 3D point on the reconstructed surface corresponding to the 2D location of the feature in the image it was found in. This is a straightforward computation, since we already computed the alignment parameters for each image during the surface construction. We call this point $\mathbf{X}_f$.
- An image patch as an additional descriptor. This patch is simply a small block of the image centered at the feature's keypoint location. Let this patch by P_f.
- A 3x3 homography matrix which maps P_f onto the image plane of the canonical camera positioned at the origin and looking down the negative z-axis. Let this homography be H_f.
- The surface normal at $\mathbf{X}_f$. We let the normal vector in this direction be $\hat{\mathbf{n}}_f$

Fig. 2. The final output of texture transfer from PTZ images. We show regions of high-resolution texture against the low-resolution surface generated from images from a single moving Kinect. We report an approximately 6x improvement in texture quality, measured as color sample per unit area, in areas where the texture was transferred as compared to where it was not.

Hence, for every feature f, we store the tuple $\langle \mathbf{x}_f, D_f, \mathbf{X}_f, P_f, H_f, \hat{\mathbf{n}}_f \rangle$. We compute such a tuple for every feature in every image in the set I_{Kinect}. We let the set of all such tuples be T, and also refer to it as the set of "oriented texture patches", since each tuple contains a complete descriptor of a small texture patch, its location and orientation in the 3D surface, and a SIFT descriptor for feature matching.

Once the set of oriented texture patches has been computed, we then process the images from the PTZ cameras. From the metadata acquired with the RGB images, we have the pan angle, tilt angle and camera intrinsic matrix available for each image j from the PTZ camera. We let these values be θ_j, ϕ_j and K_j respectively. Therefore, for each image acquired by a PTZ camera, we have a tuple $\langle j, \theta_j, \phi_j, K_j \rangle$. Let the set of all such tuples be I_{PTZ}. For each $i \in I_{PTZ}$, we attempt to synthesize a high-resolution texture from it. For any given image, we first attempt to compute an initial camera pose based on SIFT feature correspondences between the PTZ image and S_{Kinect}. If a strong correspondence cannot be found, the initial camera pose is estimated by using the camera pose from a previous image, and then adjusting it based on the angle difference in the respective orientations of the camera for those two images. The camera pose is then iteratively refined by projecting the oriented texture patches onto the camera plane, and then correcting the camera pose to align the pathces with features found on the PTZ image. In all cases where the algorithm found a strong estimate of the camera pose, the number of iterations needed was always less than 5.

The two main steps in the algorithm are Camera Pose Estimation and Surface Generation. Both steps were independently evaluated, and resulted in an improvement in the visual quality of the surface. In our experiments, we acquired a total of 1980 high-resolution images from the PTZ cameras and then attempted to estimate camera pose for each image. We define a reliable match to be one where the NCC measure between the corresponding normalized SIFT descriptor vectors is at least 0.8, and at least 0.3 (roughly 25%) more than the second best match. We were able to find a reliable set of matches and a camera pose in roughly 64% of the images. Of the matched images, we observer the pixel re-projection error mean and standard deviation to be 1.25 and 0.232 pixels respectively. Results for the Surface Generation and the subsequent fusion with the base surface are shown in Figure 2. We observed an approximately 6x improvement in texture quality, measured as color sample per unit area, in areas where the texture was transferred as compared to where it was not. Given the dimensions of our room, the average distance between color samples in the PTZ-textured regions was 1.63 mm, whereas the average distance between samples in the Kinect-sampled surface was 1 cm.

5 Augmenting the Reconstruction of Dynamic Objects

We improved the geometry by multi-view stereo reconstruction using images from the PTZ cameras. To get a base surface, we utilized an approach similar to

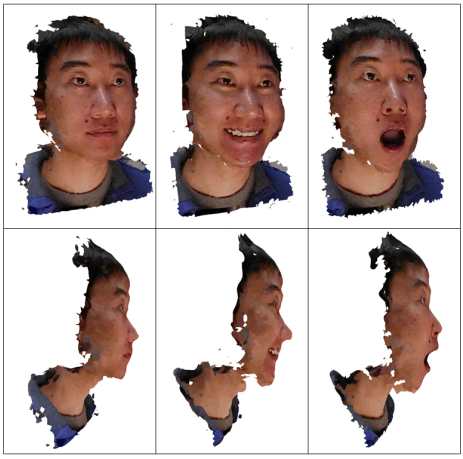

Fig. 3. The 3D point clouds obtained using only images from the PTZ cameras. These point clouds represent the union of three distinct point clouds, and then fused using known camera parameters of the PTZ cameras.

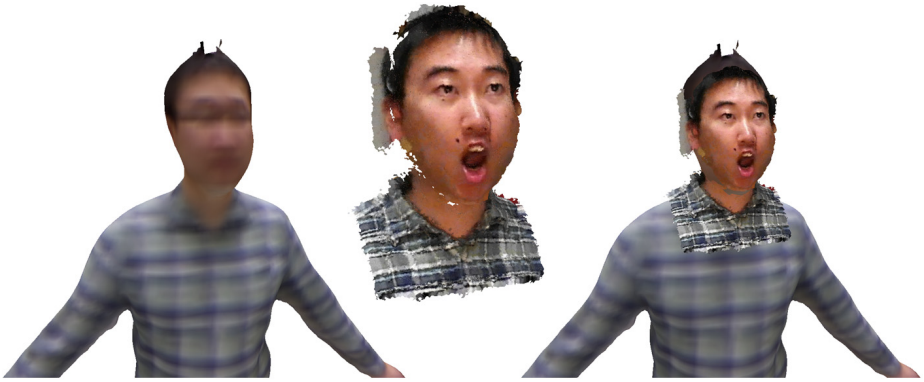

Fig. 4. The base surface from the RGB-D frames from multiple static Kinect cameras (left), the stereo reconstruction from the high resolution PTZ cameras (center) and the final fused surface (right)

the one descibed in [1]. RGB-D frames from multiple static Kinect cameras were used to build a coarse surface of the moving object, and the surface was refined over time as more and more frames were used to smooth out the surface and fill any holes. However, excessive smoothing of the surface resulted in losses in surface detail. Additionally, the system failed when the object was not moving, since the algorithm relies on spatial and temporal cohesion between frames, and varying views of the object over a period of time. Consequently, the results are far from satisfactory. This problem is an important one to tackle, since humans

are very perceptive of errors in reconstructing other human faces, and we expect that such a large loss of detail would result in a poor user experience.

In our method, we used the PTZ cameras to focus on a human face to obtain stereo-pair images for the dynamic face. We then generated a detailed surface for the face using stereo matching. The algorithm used was a Semi-Global Blocking Matching Algorithm [15] on stereo-rectified image pairs to compute a disparity map in the central PTZ cameras frame of reference. The disparity map and the camera matrices of the PTZ cameras were then used to obtain a 3D point cloud. Since we worked with 3 PTZ cameras, we had three distinct pairs of cameras, and therefore three distinct point clouds for each set of frames at a given time instance. The final point cloud was obtained as a union of the three point clouds in the same frame of reference. These point clouds contained a large amount of noise. For noise removal, we employed a Laplacian smoothing transform, similar to that utilized in [16] followed by a Poisson Reconstruction algorithm [17] for surface generation. We evaluated our method by making the subject perform a variety of facial expressions and movements. The final surfaces obtained are shown in Figure 3. We see that the high quality of the images results in a very detailed surface, and most of the facial geometry can be recovered for a variety of facial expressions and movements.

This final result was then fused with the base surface obtained from static Kinect cameras. The fusion was done by means of the camera pose estimation algorithm employed in Section 4. Background features in the PTZ images were matched against the precomputed features on the static model, thereby getting an accurate estimate of camera pose. This was then used to project the reconstructed surface back onto the base surface from the Kinect cameras. Further refinement of the surface alignment was done by means of the Iterative Closest Point algorithm [18] with automated feature detection and alignment. This results in a marked improvement in both texture and geometry of the dynamic human face. Figure 4 shows the state of the surface before and after fusion with the geometry obtained from the PTZ cameras.

6 Limitations and Future Work

The algorithm described in this paper is an improvement to the system described in [1], but there is still a lot of room for improvement. While the texture transfer technique described in Section 4 is a robust technique for the images that were matched, we still observe roughly a third of the total amount of images unable to be mapped onto the surface due to unreliable feature matching and camera pose estimation for those images. It is possible that a new kind of visual feature and keypoint descriptor would increase the robustness of the method and improve the accuracy of the estimated camera pose even further. The stereo vision and geometry transfer techniques described in Section 5 also leave room for future work, especially in the area of real-time tracking of faces instead of manual input for that purpose. Incorporating improved object-aware stereo reconstruction methods could yield a denser point cloud. Alternatively, we also intend to

explore the possibility of incorporating cameras with a higher resolution that our PTZ cameras. This would provide us more pixels of our object of interest, and therefore, a denser point cloud for geomtry reconstruction. Furthermore, the technique to re-project the dense surface back onto the base surface is also imperfect. We observe offsets at the edges of the dense surface where it does not blend well with the base surface. An example of this artefact can be seen in Figure 4. We aim to develop a smoothing technique which causes the edges of the dense surface to be continuous with the low resolution base surface.

Our system is also limited by the computing speed of the current hardware. Due to the computationally expensive nature of the 3D reconstruction being done, the computation is performed completely offline using pre-recorded datasets. We can expect that a future implementation of the same scheme on the GPU or a PC-cluster will drastically reduce the computation time per frame. An example of a real-time system which runs on a PC-cluster is explained in [5]. We used only three PTZ cameras for pair-wise stereo reconstruction of a deforming object. A possible extension of this system would include using more PTZ cameras in conjunction with an online assignment scheme such as those described in [2] and [7] to perform online assignment of PTZ cameras to obtain fine geometric features of interesting objects while they moved around in the room.

Acknowledgments. The authors would like to thank Jan-Michael Frahm, Adrian Illie and Jim Mahaney for their continued support and guidance. This work was supported in part by CISCO Systems and by the BeingThere Centre, a collaborative effort between the University of North Carolina at Chapel Hill, ETH Zurich and the Nanyang Technological University in Singapore, supported by the Singapore National Research Foundation under its International Research Centre @ Singapore Funding Initiative and administered by the Interactive Digital Media Programme Office of the Media Development Authority.

References

1. Dou, M., Fuchs, H.: Temporally enhanced 3d capture of room-sized dynamic scenes with commodity depth cameras. In: IEEE VR (2014)
2. Yous, S., Ukita, N., Kidode, M.: An assignment scheme to control multiple pan/tilt cameras for 3d video. Journal of Multimedia 2 (2007)
3. Davis, J.E.: Mixed Scale Motion Recovery. PhD thesis, Stanford University (2002)
4. Beck, S., Kunert, A., Kulik, A., Froehlich, B.: Immersive group-to-group telepresence, IEEE Transactions on Visualization and Computer Graphics (2013)
5. Matsuyama, T., Wu, X., Takai, T., Nobuhara, S.: Real-time 3d shape reconstruction, dynamic 3d mesh deformation, and high fidelity visualization for 3d video. Journal of Computer Vision and Image Understanding 96, 393 (2004)
6. Kutulakos, K.N., Seitz, S.M.: A theory of shape by shape carving. Interntional Journal of Computer Vision 38, 198 (2000)
7. Ilie, A., Welch, G.: Automated camera selection and control for better training support. In: 15th International Conference on Human-Computer Interaction, p. 50 (2013)

8. Wan, D., Zhou, J.: Stereo vision using two ptz cameras. Computer Vision and Image Understanding 112, 184 (2008)
9. Sinha, S.N., Pollefeys, M.: Towards calibrating a pan-tilt-zoom camera network. In: Workshop on Omnidirectional Vision and Camera Networks, ECCV (2004)
10. Sinha, S.N., Pollefeys, M., Kim, S.J.: High resolution multiscale panoramic mosaics from pan-tilt-zoom cameras. In: 4th Indian Conference on Computer Vision, Graphics and Image Processing (2004)
11. Kumar, S., Micheloni, C., Piciarelli, C.: Stereo localization using dual PTZ cameras. In: Jiang, X., Petkov, N. (eds.) CAIP 2009. LNCS, vol. 5702, pp. 1061–1069. Springer, Heidelberg (2009)
12. Dou, M., Guan, L., Frahm, J.-M., Fuchs, H.: Exploring high-level plane primitives for indoor 3D reconstruction with a hand-held RGB-D camera. In: Park, J.-I., Kim, J. (eds.) ACCV Workshops 2012, Part II. LNCS, vol. 7729, pp. 94–108. Springer, Heidelberg (2013)
13. Lorensen, W.E., Cline, H.E.: Marching cubes: A high resolution 3d surface construction algorithm. Computer Graphics 21 (1987)
14. Lowe, D.G.: Object recognition from local scale-invariant features. In: International Conference on Computer Vision, p. 1150 (1999)
15. Hirschmuller, H.: Stereo processing by semiglobal matching and mutual information. IEEE Transactions on Pattern Analysis and Machine Intelligence 30 (2008)
16. Sorkine, O., Cohen-Or, D., Lipman, Y., Alexa, M., Rossl, C., Seidel, H.P.: Laplacian surface editing. ACM Transactions on Computer Graphics and Modeling 175 (2004)
17. Kazhdan, M., Bolitho, M., Hoppe, H.: Poisson surface reconstruction. In: Eurographics Symposium on Geometry Processing (2006)
18. Zhang, Z.: Iterative point matching for registration of free-form curves and surfaces. International Journal of Computer Vision (1994)