Spatio-Temporally Coherent 3D Animation Reconstruction from Multi-view RGB-D Images using Landmark Sampling

Naveed Ahmed

***Abstract*—We present a system for spatio-temporally coherent 3D animation reconstruction from multi-view RGB-D images using landmark sampling. Our system captures multi-view synchronous RGB-D images from six RGB-D cameras and we show that by sampling landmarks from both depth and color images, it is possible to reconstruct a spatio-temporally consistent 3D animation from a non-coherent time-varying data. The reconstructed spatio-temporally coherent 3D animation can be used in a number of applications that require time-coherent data, e.g. motion analysis, gesture recognition, compression, free-viewpoint video and CG animations.**

***Index Terms*— Dynamic surface reconstruction, multi-view video, three-dimensional animation reconstruction, three-dimensional dynamic scene geometry**

I. Introduction

SPATIO-TEMPORALLY coherent time-varying dynamic scene geometry is employed in a number of applications. It can be used for 3D animation in digital entertainment production, electronic games, 3D television, motion analysis, gesture recognition etc. First step in obtaining spatio-temporally coherent 3D video is to capture the shape, appearance and motion of a dynamic real-world object. One or more video cameras are employed for this acquisition, but unfortunately, data obtained by these video cameras has no temporal consistency, as there is no relationship between the consecutive frames of a video stream. In addition, for a multi-view video, there is no spatial consistency between cameras even for the same frame of the video. In order to reconstruct a spatio-temporally coherent 3D animation, a spatial structure between cameras has to be established along with the temporal matching over the complete video data.

In this paper we present a new method for capturing spatio-temporal coherence between RGB-D images captured from six RGB-D video cameras. In principle, any type and combination of RGB and depth cameras can be used, but in our work we use an acquisition system comprising of Microsoft Kinect [13] cameras. Microsoft Kinect is a hybrid color (RGB) and depth camera system which provides both the color and depth information at the rate of 30 frames per second. Our acquisition system can acquire synchronous streams of RGB-D data from multiple Microsoft Kinects. We show that by extracting landmarks from both color and depth data we can establish a spatial and temporal structure that can be used to reconstruct a spatio-temporally coherent 3D animation. Our landmark sampling approach uses color data for an initial estimate for mapping two consecutive frames. In the next step, we employ a geometric based refinement method to find the accurate matching of the dynamic geometry representation in the three-dimensional space. Results from our work can be employed in a number of scenarios to enhance or analyze the representation of a dynamic real world scene.

Traditionally, the multi-view video recordings are acquired by a setup of color video cameras that are placed around a real-world object in a circular arrangement [7]. A hardware trigger is used to synchronously record a real-world dynamic object from all cameras. The recorded color multi-view video streams are then used to reconstruct a dynamic three-dimensional representation of the real-world scene. One of the pioneering works in this area, which uses multi-view video data to reconstruct free-viewpoint video, was presented by Carranza et al. [7]. They used eight cameras to record a moving person and used the multi-view data to capture the shape, motion and appearance of the person. This work was later extended by Theobalt et al. [16] to capture the shape, motion and appearance but also the surface reflectance properties of the real-world dynamic object. Vlasic et al. [17] and Aguiar et al. [8] presented enhanced methods for reconstructing very high quality of dynamic scenes. Both of these methods first acquired a high quality laser scan of the real-world person, which was then animated using a skeleton based or data driven deformation. One of the earlier works on creating spatio-temporally coherent 3D animation using landmarks was presented by Ahmed et al. [1]. Unlike other works, they did not use template geometry for tracking the dynamic object; rather the three-dimensional representation was directly obtained from RGB images. Unlike our approach they also did not incorporate geometric information for accurate matching, instead they relied on the color information to obtain the dense matches at a higher resolution than the original three-dimensional surface representation.

With the advent of low cost depth cameras, acquisition of three-dimensional geometry at high frame rate has become really feasible. Time-of-Flight [11] cameras are extensively used to obtain and manipulate the depth information in a number of applications [3][11]. Microsoft recently released Kinect as an input peripheral of Xbox 360 that can not only capture depth but also the color information at 30 frames per second. Kinect has been adopted by the research community

Manuscript received September 12, 2012; revised October 20, 2012. This work was supported by the seed grant for new faculty members from the Department of Graduate Studies, University of Sharjah, UAE.

because of its low cost, and has been used in a number of applications ranging from motion capture, gesture recognition and also dynamic three-dimensional surface deformation.

Recently, one or more depth cameras are used to reconstruct both static and dynamic real-world objects. Kim et al. [11] and Castaneda et al. [6] presented method of reconstructing a three-dimensional representation of a static object using depth cameras. Berger et al. [4], Girshich et al. [9], Weiss et al. [18], and Baak et al. [3] used depth cameras for reconstructing three-dimensional shape, pose and motion. They demonstrate that it is possible to get good quality results by employing both depth and color cameras. For capturing the dynamic scene data using depth sensors, two methods were recently presented by Kim et al. [10] and Berger et al. [4]. In [10] authors employ RGB cameras along with Time of Flight sensors to get both depth and color information while the [4] employ four Microsoft Kinect cameras to reconstruct the motion of a real world human actor. Both of these methods do not reconstruct temporally coherent animation from the captured depth and color data.

Our work derives from the work of Ahmed et al. [2] that captures six synchronous multi-view RGB-D streams using an acquisition system comprising of Kinects. In that work only the acquisition is performed with no further analysis of the acquired RGB-D data. The main motivation of our work is to present a system which starting from the acquisition of synchronous RGB-D video streams uses both color and depth information to establish a spatio-temporal consistency in the unstructured data. We assume the dynamic three-dimensional content to be stored in the form of 3D point clouds coupled with its color information. A Microsoft Kinect camera implicitly provides this information. Unlike a setup of stereo cameras to reconstruct the depth, the major advantage of Kinect is that it provides the depth information from one sensor. Thus in principal only four Kinects can provide full 360^o reconstruction of a real-world object, whereas traditional acquisition systems comprising of color cameras required eight or more cameras for the similar reconstruction. Our spatio-temporal reconstruction method is not limited to the data from Kinect cameras. It can be easily applied to any dynamic three-dimensional content that is in the form of point clouds with the available color information. The main contributions of our work are:

1) A system for automatic acquisition of time-varying RGB-D data with a new algorithm for background subtraction using 3D point clouds.

2) A color and depth based landmark sampling method that is used to establish spatial and temporal coherence between consecutive frames of dynamic three-dimensional point clouds. The point cloud data with the color information can be estimated from color multi-view video data or RGB-D multi-view data from Microsoft Kinect.

II. Multi-view video acquisition

Our acquisition system is comprised of six Microsoft Kinect cameras. Four cameras are placed around the real-world person with each making an angle of 90^o with the adjacent corner cameras. Two cameras are placed on left and right between the corner cameras. They make an angle of 180^o with respect to each other and an angle of 45^o with their adjacent corner cameras. This arrangement gives us 360^o coverage of the real-world actor and allows the recording of a dynamic scene within an area of 2m x 3m. Since Kinect projects an infrared pattern to reconstruct depth information, having two or more Kinects recording at the same time can cause interference in the depth estimation. Ideal placement of two Kinects would be with a separation of 180^o. For our work, we do not try to reduce the interference in any way under the assumption that the depth information for a point not recorded by one camera due to interference will be compensated by one of the other cameras recording the same point. Our results validate this assumption. In addition to the data from our acquisition system, we also use data from Ahmed et al. [1] to validate our algorithm.

The Kinect provides two video data streams, one color stream and one depth stream. Both streams are of the resolution 640x480 with the frame rate of 30 frames per second. Using the new Microsoft Kinect SDK it is possible to obtain a higher resolution depth stream but at the cost of lower frame rate. For our work, high-speed recording is more important therefore we decided to use the lower quality of video streams. Our acquisition setup is software synchronized and does not require any hardware trigger. To minimize I/O overhead that comes with writing video data to the storage device, the video streams are captured in the high-speed system memory buffer and the writing is performed once the recording is finished.

Once the acquisition is completed we acquire sequence of color and depth images of a person performing some motion. For each frame we record six color images (Fig. 1) and six depth images (Fig. 2).

Fig. 1. One color frame captured from the RGB camera.

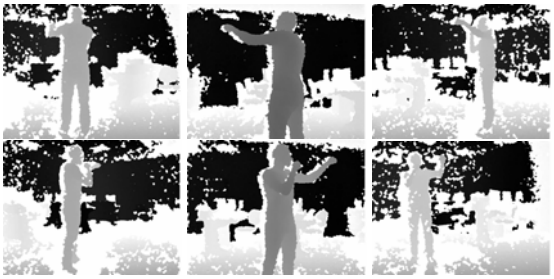


Figure 2: One depth frame captured from the depth camera.

III. Calibration and Background Subtraction

A multi-view acquisition system requires both local and global calibration. Local calibration provides camera

ISBN: 978-988-19251-8-3
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

specific parameters, or intrinsic parameters. On the other hand, the global calibration or extrinsic parameters provide the spatial mapping between the cameras.

For a Microsoft Kinect, which has two sensors, there is an additional level of local calibration. In the first step, both the depth and color sensors have to be calibrated to estimate their intrinsic parameters. Secondly, a mapping should be established between the depth and color sensors so that color data can be projected on the depth data. Finally, depth values are mapped to real-world distances in order to get 3D positions in a global coordinate system.

The intrinsic parameters are obtained using Matlab Camera Calibration toolkit. We record a checkerboard from both color and infrared sensors to facilitate this calibration. To convert the depth data to meters we employ the method proposed by Nicolas Burrus [5]. We use the Kinect RGB Demo software to do the full internal calibration. Using the internal calibration we obtain a 3D point cloud for each camera along with its mapping to the color data. An example of the 3D point cloud with depth to color mapping can be seen in Fig. 3.

Figure 3: A dynamic 3D point cloud from one camera with the depth to color mapping.

We perform the global calibration between the six cameras by means of the Iterative Closest Point (ICP) algorithm that minimizes the distance between the six point clouds. The SIFT [12] features obtained from the color data and correspondences from the recorded checkerboard are used to initialize the ICP method. We make use of OpenCV for extracting corner points from the checkerboard and the Point Cloud Library [14] for the ICP.

After merging the point clouds in a global coordinate system, we segment the moving actor by estimating the bounding box of the dynamic scene geometry of the entire sequence. This is achieved by finding the correspondences between every tenth frames for all cameras in the RGB images using SIFT. Using the mapping between the depth and RGB images, we find the 3D points that exhibit motion over these frames. These dynamic points are then used to estimate an axis aligned bounding box. We compute one bounding box for the whole sequence. It is possible to generate a bounding box for every n frames to make segmentation more precise. After estimation, the bounding box is slightly scaled to make it a conservative estimate. We found this method to be more robust than a pure RGB image or depth image based segmentation. Depth-image-based segmentation is straightforward and simple but due to the noisy nature of depth data from the Kinect sensor, we found a number of false positives over the complete scene. The results of global registration and background subtraction can be seen in Fig. 4 and 5. After segmentation, we also perform simple Gaussian filtering to remove outliers and random noise using the Point Cloud Library.

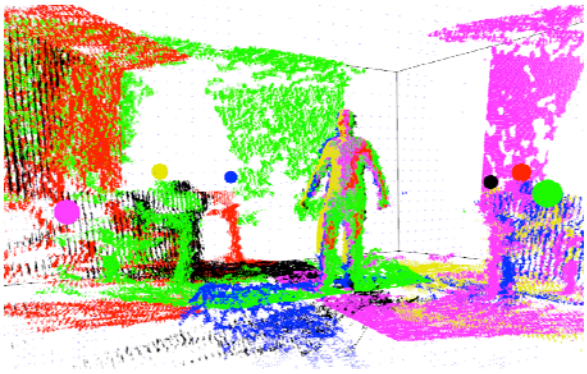


Figure 4: Result from the global calibration. Point clouds from six cameras are merged in a global coordinate system. The cameras (shown in circle) and their corresponding point clouds are color-coded.

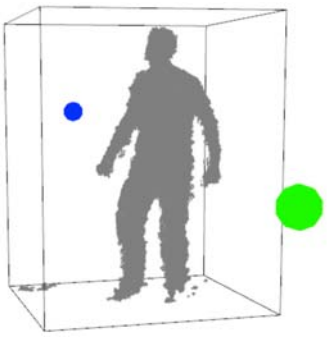


Figure 5: Result from the background subtraction. The estimated bounding box used to segment the model is shown along with the complete 3D model. It can be seen that the segmentation works really well in separating the foreground from the background.

IV. Temporally Coherent 3D Animation Reconstruction

As explained in the previous sections, from local and global calibrations we obtain dynamic 3D point clouds of the real world actor. This dynamic representation is not temporally coherent because each frame of the dynamic data is independent of the other. In addition to the data from our acquisition system, we use data from Ahmed et al. [1]. Given the visual hull representation from the data, the 3D point clouds are extracted and the color images are projected onto the 3D data using the camera calibration parameters. Our method, explained below, that reconstructs spatio-temporally consistent 3D animation works equally well on both data sets.

Generating temporally consistent dynamic scene representation has a number of benefits. It is useful in terms of visualization, as the appearance of the model does not change from one frame to the next. In addition, it allows analyzing a number of attributes of the dynamic data, e.g. motion, gesture or action. Given the temporal coherence data can be easily compressed and its transmission is simplified by means of some parameterization unlike the non-coherent data, which requires all the frames to be transmitted individually.

ISBN: 978-988-19251-8-3
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

While our work on global calibration establishes the spatial correspondence for one frame between six cameras, we have developed a new method to establish temporal correspondence between two consecutive frames of the dynamic point cloud. Starting from the first two frames, this temporal correspondence is tracked over the sequence to generate a spatio-temporally coherent 3D animation.

Our temporal coherence extraction scheme is based on matching landmarks over two consecutive frames. The process comprises of two steps:

1) *Establishing reliable landmarks on each frame*
2) *Matching landmarks accurately*

These two steps are not discreet; rather we propose an iterative process that first establishes a rough correspondence between two frames and then refine it to get an accurate match.

In the first step we extract SIFT features from each color image for the frames t_0 and t_1. Where t_0 is the first frame of the animation and t_1 is the second frame of the animation. Matching of the feature points gives us a reliable matching in the RGB data for each of the six cameras. Since we have depth to RGB mapping, we can directly find the mapping of a 3D point at t_0 to the corresponding 3D point at t_1. Unfortunately the 3D correspondences are not accurate because depth to RGB mapping is many-to-one. Thus multiple 3D points match to a single pixel in the RGB image. Given a number of mappings from 3D points at t_0 to t_1, we use the approach proposed by Tevs et al. [15] to randomly choose one of the mapping as the landmark. This gives us the first rough map between the two point clouds.

In the second step, we start an iterative process that randomly picks one of the matching M_0 to M_1 found in the first step. Here we are assuming that M_0 is not just a single 3D point but a set of all 3D points that can potentially match to corresponding 3D points M_1 at frame #1. It is to be noted that in the first step we chose just one of the matching randomly as the coarse matching to facilitate the iterative process. Given the coarse matching from M_0 to M_1 we search for three non-collinear nearest landmarks in the two point clouds with respect to the Euclidean distance. In practice we never found three nearest collinear landmarks but in case the three landmarks are collinear, the one at the farthest is to be discarded and the next closest one is to be selected. The non-collinear matches are required because once found we use the three 3D positions on each frame to construct a plane with normal pointing outwards to the point cloud. Assuming the nearest landmarks at frame #0 are L_{00}, L_{01} and L_{02} and on frame #1 are L_{10}, L_{11} and L_{12}. We find two planes at each frame P_0 and P_1 with their normal being n_0 and n_1 respectively. Given the two planes, their normal and the root points, it is trivial to parameterize the matching points M_0 and M_1 with respect to P_0 and P_1:

$$M_0 = L_{00} + u(L_{01}-L_{00})+v(L_{02}-L_{00}) \quad (1)$$

$$M_1 = L_{10} + u(L_{11}-L_{10})+v(L_{12}-L_{10}) \quad (2)$$

Where u and v are the two parameters that define the projection of each 3D position in M_0 and M_1 on P_0 and P_1. It is to be noted that the root points L_{00} *and* L_{10} *are* chosen randomly. This assumption is important because this step is repeated multiple times and the random selection reduces the bias in the estimation. Given the parameterization in Equations 1 and 2, for all 3D positions within the landmark matches M_0 to M_1 that are obtained in the first step, we define the new match that has the minimum distance within the parameterized space, i.e. its u and v coordinates at t_0 and t_1. The second step is repeated multiple times, with the starting point chosen at random, and the root points also chosen at random. As shown by Tevs et al. [15] that the random sampling with an iterative process is sufficient to correctly establish an unbiased mapping, thus we obtain a correct matching of two frames using a geometric based mapping algorithm which uses color based matching as the starting point. The iterative process stops when the matching points are stabilized over the sequence of 5 iterations.

Once the mapping between t_0 and t_1 is established, it is propagated to the mapping between t_1 and t_2, ideally till the end of the sequence or unless it degenerates.

V. Results

We use two types of data sets to validate our method. First data set is acquired through our multi Kinects acquisition setup which is described in detail in the earlier sections. We recorded three sequences using six Kinects where the actor performs a slow rotating motion, a medium walking motion and a fast boxing motion. We use this data set to verify our background subtraction approach. Fig. 1 and 2 show the captured color and depth images for one frame. Fig. 3 shows that 3D point cloud with the color information, whereas Fig. 4 and 5 shows the results from global calibration and background subtraction. It can be seen that we manage to separate the moving actor reliably from the static background.

Our spatio-temporal reconstruction method manages to track more than 180 frames from the slow motion, around 150 frames from the medium motion and 130 frames from the fast motion. This is expected because as the motion gets faster it introduces motion blur in the color data which results in lower number of landmarks thus affecting the matching. Additionally, Kinect's depth sensor provides the depth data marred with a very high random noise. We try to remove this noise using the simple Gaussian filtering but it is still not completely eliminated. Thus for the faster motion there are far more outliers compared to the slower motion.

We also validated our approach on the data from Ahmed et al. [1]. Our method managed to track both sequences reliably as their data is noise free with high quality of color images. Fig 6a shows two frames from the walking sequences without any temporal coherence. Fig. 6b shows the same two frames generated from our spatio-temporal 3D animation reconstruction method. In the non-coherent animation the point cloud visibly changes over the two frames and the effect is really pronounced over the complete animation. The spatio-temporal 3D animation on the other hand tracks the single point cloud over the whole sequence that results in a visually smoother animation, which can be used in a number of applications.

Our method is subject to couple of limitations. One of the major limitations is the quality of the color data. If the number of SIFT features are low then our method is suspect to producing incorrect results because of a low number of landmarks. We use three nearest landmarks based on Euclidean distance for our iterative random sampling based matching algorithm. If the number of landmarks is low then

ISBN: 978-988-19251-8-3
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

the matching plane will have incorrect orientations over the two frames and the matching will go meaningless. Ideally Geodesic distance should be used instead of Euclidean distance similar to Ahmed et al. [1], but it requires a surface representation. Since we are dealing with the point clouds with a very high random noise, therefore the surface reconstruction is not an option. A true dynamic surface reconstruction from the depth data acquired by Kinect is a complete research problem in itself. Other possibility would be to estimate the pose skeleton and also use the nearest joint position as a landmark. We are planning to extend our work in this direction.

Despite the limitations, we show that it is possible to reconstruct spatio-temporally coherent 3D animation from 3D dynamic point clouds using both color and depth information.

VI. Conclusion

We presented a method to reconstruct spatio-temporal 3D animation from dynamic point clouds using a color and depth based landmark sampling approach. We showed that data from multiple Kinects can be used to create a dynamic representation of a real-world object that can be merged together to capture the object from 360^o. Our new method for background subtraction reliably separates the foreground dynamic object from the static background. Our system can incorporate any number of cameras, as we demonstrated that not only it works for the data acquired using Kinects but also through the traditional acquisition system comprising of color cameras. Our works leads to a number of exciting directions in the future. We plan to use new Microsoft's Kinect SDK to capture not only the depth but also the pose of the human actor. This information can greatly enhance the landmark sampling algorithm. In addition we would also like to explore 3D surface reconstruction from the dynamic 3D point cloud data. The spatio-temporal 3D animation can also be used for the motion analysis, compression and parameterization of the 3D video data.

References

[1] AHMED N, THEOBALT C, ROSSL C, THRUN S, and SEIDEL H.-P. Dense correspondence finding for parametrization-free animation reconstruction from video. In CVPR, 2008.

[2] AHMED N, A System for 360 degree Acquisition and 3D Animation Reconstruction using Multiple RGB-D Cameras. In CASA, 2012.

[3] BAAK A., MULLER M., BHARAJ G., SEIDEL H.-P., THEOBALT C.: A data-driven approach for real-time full bodypose reconstruction from a depth camera. In ICCV (2011).

[4] BERGER K., RUHL K., SCHROEDER Y., BRUEMMER C., SCHOLZ A., MAGNOR M. A.: Markerless motion captureusing multiple color-depth sensors. In VMV (2011), pp. 317–324.

[5] BURRUS N.: Kinect RGB demo. http://labs.manctl.com/rgbdemo/

[6] CASTANEDA V., MATEUS D., NAVAB N.: Stereo time-of-flight. In ICCV (2011).

[7] CARRANZA J., THEOBALT C., MAGNOR M. A.,SEIDEL H.-P.: Free-viewpoint video of human actors. In Siggraph 2003

[8] DE AGUIAR E., STOLL C., THEOBALT C., AHMEDN., SEIDEL H.-P., THRUN S.: Performance capture from sparsemulti-view video. ACM Trans. Graph. 27, 3 (2008).

[9] GIRSHICK R., SHOTTON J., KOHLI P., CRIMINISIA., FITZGIBBON A.: Efficient regression of general-activity humanposes from depth images. In ICCV (2011).

[10] KIM Y. M., CHAN D., THEOBALT C., THRUN S.:Design and calibration of a multi-view tof sensor fusion system. In IEEE CVPR Workshop on Time-of-flight 2008.

[11] KIM Y. M., THEOBALT C., DIEBEL J., KOSECKA J.,MICUSIK B., THRUN S.: Multi-view image and tof sensor fusionfor dense 3d reconstruction. In IEEE Workshop on 3-D Digital Imaging and Modeling (3DIM) 2009.

[12] LOWE D. G.: Object recognition from local scaleinvariant features. In ICCV (1999), pp. 1150–1157.

[13] MICROSOFT: Kinect for microsoft windows and xbox360.

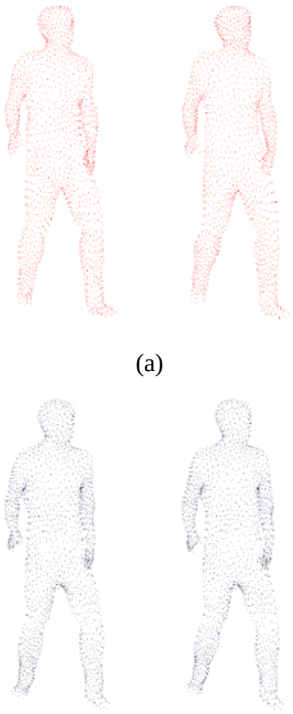
[14] RADU B. R. and STEVE C.. 3D is here: Point Cloud Library (PCL). In ICRA, 2011.

[15] TEVS A, BERNER A., WAND M., IHRKE I., SEIDEL H.-P.: Intrinsic Shape Matching by Planned Landmark Sampling. In Eurographics 2011,

[16] THEOBALT C., AHMED N., ZIEGLER G., SEIDEL H.-P.: High-quality reconstruction of virtual actors from multi-view video streams. IEEE Signal Processing Magazine 24, 6 (November 2007).

[17] VLASIC D., BARAN I., MATUSIK W., POPOVIC J.:Articulated mesh animation from multi-view silhouettes. ACMTrans. Graph. 27, 3 (2008).

[18] WEISS A., HIRSHBERG D., BLACK M. J.: Home 3dbody scans from noisy image and range data. In ICCV (2011).



(a)

(b)

Figure 6: Two non-coherent consecutive frames of 3D point cloud are shown in (a). Whereas (b) shows the same two frames generated using landmark sampling method to reconstruct spatio-temporally coherent 3D animation. It can be observed that the point clouds do not change in the coherent animation. This result is especially visible in the legs and feet.

ISBN: 978-988-19251-8-3
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)