Journal of Computers Vol. 30 No. 3, 2019, pp. 102-116
doi:10.3966/199115992019063003008

A Spatio-Temporal-Contour-Consistency-Based Shape Tracking Framework from Multi-View Video

Chu-Hua Huang[1*], Jin Qin[2], Zhi Li[3]

[1] College of Computer Science and Technology, Guizhou University, Guiyang, 550025, China
chhuang@gzu.edu.cn

[2] College of Computer Science and Technology, Guizhou University, Guiyang, 550025, China
qin_gs@163.com

[3] College of Computer Science and Technology, Guizhou University, Guiyang, 550025, China
lizhigzu@163.com

Received 26 August 2017; Revised 18 December 2017; Accepted 1 February 2018

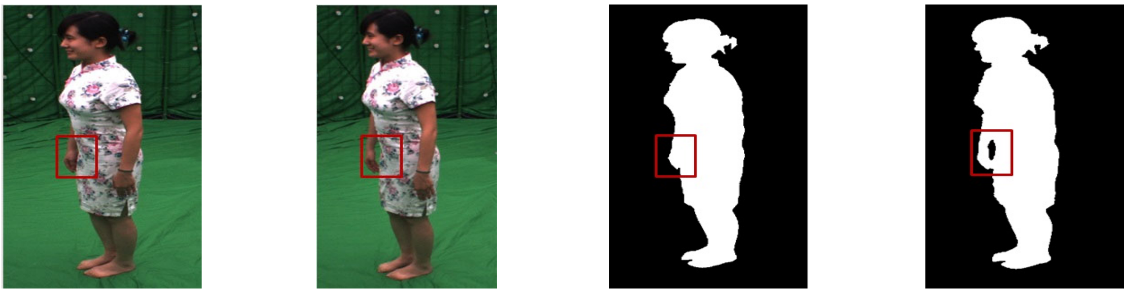
Abstract. The Spatio-temporal coherence of time-varying model sequence is an important and highly-desirable property in varies of applications. We propose an efficient framework that reconstructs a Spatio-temporally coherent point cloud sequence for dynamic objects from multi-view video. The main distinction of the framework is that it does not complicatedly and strictly match between any two contours on the same model or the neighboring model. The point cloud between the discrete frames is obtained by transporting the neighboring models or by reconstructing the in-between model based on interpolation silhouette. The time-varying and quasi-dense point cloud of such dynamic objects shape can be tracked successfully thanks to Spatio-Temporal-Contour consistency and distance field interpolation employed in our shape tracking framework. Compared to existing approaches, the resulting point cloud sequence has better Spatio-temporal coherence. Experimental results demonstrate that the framework obtains the promise.

Keywords: contour consistency, distance field interpolation, free-viewpoint video, shape tracking framework

1 Introduction

In recent years, Free-Viewpoint Video (FVV), which consists of Spatio-temporally coherent dynamic model, becomes a popular topic in the computer vision field. FVV with the higher frame-rates is preferable in various domains such as entertainment, education, sports, medicine, and culture heritage [1]. The reconstruction of dynamic scene is a key technology to obtain FVV. However, the Spatio-temporal coherence of time-varying model sequence, which is generated from multi-view video, is not retrieved well. As is illustrated in Fig. 1, the red rectangle indicates the right arm which moves between two frames. The actor's arms between two frames are not captured by any of the camera. The temporal consistency of data obtained by these video cameras is not enough to generate time-varying model sequence with the better Spatio-temporal coherence. However, Spatio-temporal coherence is an important and highly-desirable property in varies of applications [2]. For example, when the position of the points change so quickly from frame to frame, it is not visually pleasing because it distracts the viewer from the actual animation. It is necessary to reconstruct the model when the deformation takes place between frames. Otherwise the Spatio-temporal coherence of time-varying model sequence is destroyed.

[*] Corresponding Author



(a) are the neighboring frames from a video sequence

(b) are the neighboring frames from a video sequence

(c) is the neighboring silhouette corresponding

(d) is the neighboring silhouette corresponding

Fig. 1.

To solve the problem, it is necessary to increase the number of the in-between model between the key models, namely improve the temporal coherence. Our work is motivated by the problem. It allows the transport between two models represented by point cloud. The point cloud representation of each frame is obtained by transporting the point cloud of the previous frame towards the optimal shape defined by the silhouettes corresponding to the time instance.

The primary contributions of this paper are as follows:

(1) It speeds up generating the quasi-dense point cloud using Spatio-Contour consistency from scratch, separately for each time instance.

(2) It improves the temporal coherence of the time-varying point cloud sequence using distance field interpolation and Temporal-Contour consistency.

Our work aims at obtaining the point cloud sequence with better Spatio-temporal coherence under the suitable time-complexity. In particular, the proposed framework targeted at handling a video of dynamic scene captured by a multiple-view system.

This paper is organized as follows: first, in Section 2, we discuss the related work on the shape tracking work; in Section 3, the overall shape tracking system, which makes use of multi-view silhouette image, Spatio-Temporal-Contour consistency, and distance field interpolation (DFI), is described. Section 4 provides and discusses the experimental results, and finally, Section 5 gives concluding remarks.

2 Related Work

In this section, we briefly review the shape tracking work. In the past, most of the work on the shape tracking has focused on mesh deformation with small-scale deformation [3-11]. There is actually very little work with respect to the shape tracking based on point cloud representation [12-15]. Related work of our work falls mainly into two categories [16]:

Physical-based. The approaches need to obtain the priori physical model which controls the transport of the model. The intermediate model is obtained according to the energy function. To some extent, the transport simulates the real physical phenomena. The optimized energy function determines transport path. The most shortcoming of the approach is the large amount of calculation. It means the approach is time-consuming. The approach is unsuitable to reconstruct the dynamic scene because it is difficult to represent the trajectory of the dynamic object or the part of one using the identified physical function [16].

Geometry-based. The approach yields the intermediate geometry model between source models and target ones. It is the morph by shape deformation techniques. More recently, the approach further falls mainly into two categories [12]:

(1) Surface-based approaches

In the approaches, the match is an integral part of most shape tracking pipelines and a very crucial part. It usually start by finding correspondences between the source surface and target one, which is also an important process for non-rigid surface registration, otherwise known as deformable object registration [5, 12]. Unfortunately, establishing correct correspondence is very difficult in reality. As is known to all, obtaining the correspondence between the points is a typical NP-hard problem.

A large number of algorithms have been developed to address the challenging problem. The artificial marker is often used. Zhao et al. [17] proposed a novel non-linear algorithm that achieves as-rigid-as-possible deformation through constraining the rigidity of the local neighborhoods. The original point cloud is efficiently simplified by clustering, and then the deformation is performed on the simplified point cloud. A dynamic resampling method is introduced to eliminate the redundant points. Their approach preserves both detail and volume under deformations. The main disadvantage of the proposed approach is time-consuming and it needs artificial markers. In order to overcome the limitations of artificial markers, Iterative approaches, such as the classic Iterated-Closest-Point algorithm and its variants, are often used when the relative transformations between the two scans are not very large. Because there is a large number of wrong matches by establishing the relationship based on the ruler of neighboring point when the shape between two models exhibits larger difference. Therefore the iteration may fall into local optimization rather not be global optimization and the most optimal match cannot be attained between two models. Some approaches alternate the computation of correspondences with refining the aligning transformation, and almost always require multiple iterations before the final alignment is found. Li et al. [4] combine the calculation of obvious correspondence with the optimization of the global deformation. To some extent, their approaches overcome the limitation of the small-scale deformation. Other researchers apply random sample consensus filtering, deformation driven strategy, Mobius cluster and so on, to solve the problem of model match. However, these approaches overhead time is not satisfactory.

Cmollk et al. [14] classify the point of the source and target point cloud by clustering operations and built the binary tree of the model respectively. Their approach considers only the local geometric information expressed by the point locations in 3D space. But the approach may appear a lot of cracks and holes during transport process and focus on static objects. Hence it is time-consuming. Similarly, Nakajima et al. [12] formulate the interpolation as point cloud transport rather than non-rigid surface deformation. Guo et al. [18] propose a novel L_0 strategy that integrate into the available non-rigid motion tracking pipeline. When human body motions with occlusions, facial and hand motions, the tracking technique substantially improves the robustness and accuracy in motion tracking.

Like our work, Ahmed et al. [2] deals with multi-view video and obtains the unstructured point cloud. Their approach reconstructs source model and target one from scratch, separately for each time instance and focus on the match between models.

The above-mentioned approaches have the following disadvantages:

(a) The approaches require obtaining the source model and target model before the deformation. It is not always possible to acquire the appropriate models that one wishes to deform.

(b) The approaches need the smooth compact connected surfaces. However, It could be difficult if the course model or target model contain significant noise and also discontinuities due to occlusions.

(c) The approaches require establishing the correct correspondence. Nevertheless it is very difficult, demanding in reality, and time-consuming. It is unsuitable to reconstruct quickly the dynamic object.

(2) Volume-based approaches

The volume-based approaches treat topologically different shapes in a more natural way by interpolating signed distance fields for source shapes and target ones.

Cohen-Or et al. [19] adapt interpolating 3D distance field to control the mesh deformation. Moreover, the approach has the shortcoming that it is computationally expensive to obtain the 3D distance field. What is more, the approach is invalid to point cloud since it requires the closure surface. The paper inspires ours. Yemez et al. [20] can obtain the surface by the expression $f(x,y,z)=0$. $f(\bullet)$ stands for the 3D iso-level function. Sucmuth et al. [8] given a set of unordered point cloud, their algorithm is able to compute coherent meshes which approximate the input data at arbitrary time instances. They obtain reconstructions for further time-steps which have the same connectivity as the previously extracted mesh while recovering rigid motion exactly. The approach needs prior template mesh and maintains the topology of the dynamic model during deformation process. Shinya et al. [15] employ distance field as the factor of the energy function which controls the deformation path. The mathematical model is a very crucial problem in the approach. Yang et al. [21] propose a novel sparse-sequence fusion (SSF) algorithm for handheld scanning using commodity depth cameras, obtaining the fused result by integrating the refined depth images into the truncated signed distance field (TSDF) of the target. Ji et al. [22] term the network SurfaceNet. It takes a set of images and their corresponding camera parameters as input and directly infers the 3D model. Their approach is that both photo-consistency as well geometric relations of the surface structure can be directly learned for the purpose of multi−view stereopsis in an end-to-end fashion.

Existing reconstruction approaches for reconstruction dynamic and static object are difficulty to apply them for real-time reconstruction of the dynamic scene. Because it may lead to the following problems:

(a) It is rather difficult to obtain accurate match between two models. Although some approaches achieve this goal, the reconstruction speed is not satisfactory. At the same time, these approaches limit the degree of the deformation of the object.

(b) It is difficult to maintain the topological consistency during deformation. Reconstruction quality is difficult to be guaranteed. The approaches may appear a lot of cracks and holes during transport.

Based on the strengths and drawbacks of the existing methods, we adopted shape-from-silhouette (SFS) technique, Spatio-Temporal-Contour consistency to reconstruct a quasi-dense point cloud sequence.

3 Shape Tracking Framework

The Fig. 2 illustrates the shape tracking framework. Here the term “shape tracking”, in the way we use it, refers to gradual and continuous transport one shape to another shape, while producing the in-between point cloud; The term “high confidence point”, refers to these points which make up of the sparse point cloud; the term “expansion point” refer to the point which is obtained by interpolating between high confidence points.

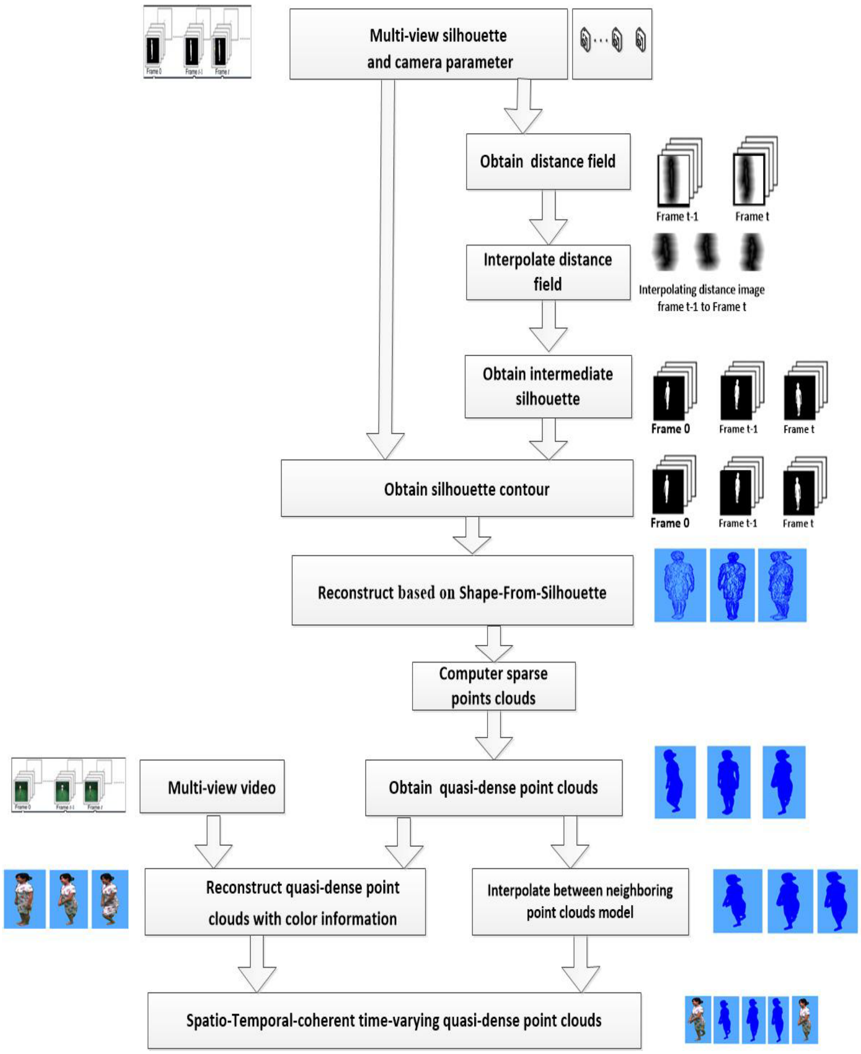

Multi-view silhouette
and camera parameter
Obtain distance field
Frame t-1
Frame t
Interpolate distance field
Interpolating distance image
frame t-1 to Frame t
Obtain intermediate silhouette
Frame 0
Frame t-1
Frame t
Obtain silhouette contour
Frame 0
Frame t-1
Frame t
Reconstruct based on Shape-From-Silhouette
Computer sparse points clouds
Multi-view video
Obtain quasi-dense point clouds
Reconstruct quasi-dense point clouds with color information
Interpolate between neighboring point clouds model
Spatio-Temporal-coherent time-varying quasi-dense point clouds

Fig. 2. Block diagram of the proposed shape tracking system

We use the point cloud as an approximate model in the framework. There is an important reason that the approach, which reconstructs the object from the multi-view video, generates a large number of points. We extend SFS to the dynamic case so as to address the problem of shape tracking. The shape tracking technique is mainly based on interpolating distance field. This is mainly because the distance values to the boundary change smoothly during the transformation. Therefore the technique can reduce the discrimination between source curves and target one during point cloud transport.

The shape tracking framework consists of three stages: Reconstructing sparse point cloud; Expanding sparse point cloud based on Spatio-Contour consistency; Tracking quasi-dense point cloud sequence based on Temporal-Contour consistency. Details of each of these stages are given in the following subsections.

3.1 Reconstructing Sparse Point Cloud

3.1.1 Extracting the Intersection between Epipolar Line and Occluding Contours

In this section, we solve the problem how to extract accurate intersection between epipolar line and occluding contours. The method consists of three steps:

Firstly, we obtain the epipolar line. In order for the efficiency and simplicity, we simulate the epipolar line using the algorithm of digital differential analyzer (DDA) [23].

Secondly, we traverse the epipolar line according to increasing order of X coordinate component from pixel startpoint to endpoint, or vice versa.

Thirdly, the relationship between pixel and contour is determined by isolevel function. That is, we obtain the accurate intersection between contour edge and epipolar line by isolevel function. Assumed $I = \{I_i \mid 1 \le i \le M\}$ be the set of corresponding silhouette images taken at each slot, the isolevel function is represented by Eq. (1):

$$G(u,v) = (1-a)((1-b)I(\lfloor u \rfloor,\lfloor v \rfloor) + bI(\lfloor u \rfloor,\lfloor v \rfloor+1) + a((1-b)I(\lfloor u \rfloor+1,\lfloor v \rfloor) + bI(\lfloor u \rfloor+1,\lfloor v \rfloor+1)). \quad \textbf{(1)}$$

Where $(\lfloor u \rfloor,\lfloor v \rfloor)$ denotes the integer part, and (a,b) the fractional part of the projection coordinate (u,v) on the binary silhouette image I_n. The function G, taking values between 0 and 1, is the bilinear interpolation of the sub-pixel projection corresponding to the binary silhouette image I_n (0 for outside, 1 for inside).

During the traversal on the epipolar line, we obtain the intersection between contour edge and epipolar line when the function value changes from negative value to positive value, or vice versa. By this means, we can improve the accuracy of the intersection

3.1.2 Reconstructing Raw Point Cloud

We reconstruct the raw point cloud using the silhouette-based static object reconstruction method. Parts of this work, which is our completed work, have appeared in Huang et al. [24]. The raw model is semi-structured point cloud because we can get the ordered lists of points according to the index of each silhouette contour, respectively. The information of the order is benefit of the expansion of the point cloud.

3.1.3 Removing the Outliers

In practice, there exist errors in the process of reconstructing sparse model because of calculation error. The errors would drift over time in the next stages. To avoid error propagation, we remove the noise point by projecting the raw model onto the image plane and enforce of silhouette contour constraint. Our method consists of the following main steps: obtaining iso-level value of the pixel; judging relationship between silhouette contours and pixel by the statistical scheme.

Obtaining iso-level value of the pixel. To express the idea mathematically, assumed $I = \{I_j \mid 0 \le j < M\}$ (M is the number of the view) is the set of corresponding silhouette images taken at each slot; point cloud $O=\{P_i \in \mathbb{R}^3 \mid 0 \le i < N\}$ (N is the number of point), Perspective projection $\prod_j$

(j=0, 1, 2, 3… $M-1$) is represented by a 3×4 matrix. P_i is projected onto the all the other silhouette planes by Eq. (1):

$$p_i^j = \{P_i \mid \Pi_j(P_i)\}. \quad \textbf{(1)}$$

where p_i^j is the pixel corresponding to silhouette image I_j. The pixel p_i^j is IN if the isolevel value is 0.5, OUT if -0.5 and ON if in-between.

Judging location relationship between point and model. Generally speaking, if one point is inside or onside the model, it must be its projection point is inside all silhouette contours. However, the numerical instability leads that it is difficult to judge the location relationship between point and model. Because the pixel corresponding to the point may be inside some silhouette contours or outside another silhouette contours. Therefore the above-mentioned conclusion is not also true. In order to solve the problem, we utilize the statistical scheme to judge location relationship between point and model.

3.2 Expanding Sparse Point Cloud Based on Spatio-Contour Consistency

3.2.1 Spatio-Contour Consistency

There exists the Spatio-Contour consistency between two neighboring camera views at same slot *t*. As is shown in Fig. 3(a) is the source silhouette image, Fig. 3(c) is the target silhouette image, Fig. 3(b) is the interpolation silhouette image between Fig. 3(a) and Fig. 3(c). The consistency of the neighboring spatio contour will rise along with increasing the number of the views.

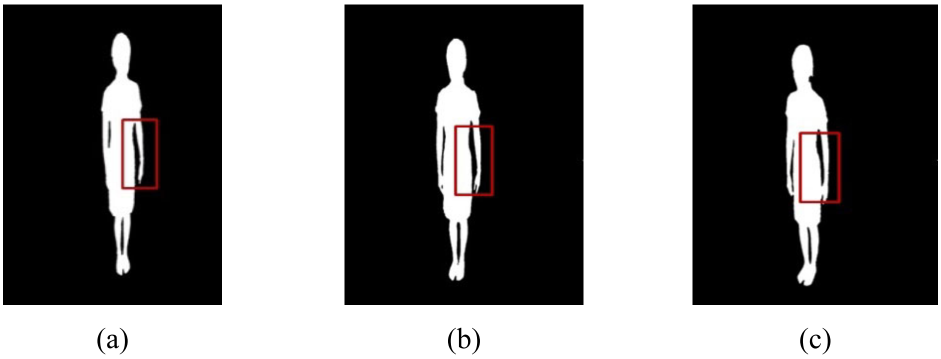


Fig. 3. The silhouette image from neighboring views, respectively

Franco et al. [25] mesh the point cloud using the similarly consistency in order to improve the reconstruction quality. We now use the consistency for expanding the sparse point cloud.

3.2.2 Expanding Sparse Point Model

According to the Spatio-Contour-Consistency theory, we interpolate between two neighboring contour curves. Given a point cloud O consists of Ncontour curve corresponding to M camera views. Let us consider for simplicity the interpolating two neighboring contour $C_{Source} \in O$ and $C_{Target} \subseteq O$. The expansion procedure is implemented in sequential order as: finding corresponding point-pair; interpolating between two curves; removing the outliers.

Finding corresponding point-pair. Assumed $P_{i,j}^{k_1} \in C_{Source}, P_{i,j}^{k_2} \in C_{Target}$, i=0, 1, 2,…… $M-1$, j=0, 1, 2,…… $N-1$; *M*, *N* is the number of camera views, contour, respectively ; k_1=0, 1, 2,…… $m-1$, k_2 =0, 1, 2,…… $n-1$, *m*, *n* is the number of the point of the contour C_{Source}, C_{Target}, respectively. We find the corresponding point for each point $P_{i,j}^{k_1}$. The rule is the closest Euclidean distance in spatial.

In order to find the point-pair, we minimize the sum of square differences between C_{Source} and C_{Target}.

The cost function is given by Eq.(2) :

$$E = \arg\min_{k_1,k_2} \sum_{k_1=0}^{m} \sum_{k_2=0}^{n} d\left(P_{i,j}^{k_1}, P_{i,j}^{k_2}\right). \qquad \textbf{(2)}$$

where $d\left(P_{i,j}^{k_1}, P_{i,j}^{k_2}\right) = \left\| P_{i,j}^{k_1} \text{-} P_{i,j}^{k_2} \right\|$ is the point-to-point Euclidean distance. $\left(P_{i,j}^{k_3}, P_{i,j}^{k_4}\right)$ stands for the corresponding point-pair, k_3 =0, 1, 2,...... $m_1 - 1$, k_4 =0, 1, 2,...... $m_1 - 1$, m_1 is the number of the point-pair.

To improve the efficiency, one can reduce the traverse range by employing a sliding window strategy. The sliding window contains the last N_v point (N_v = 20 in our implementation). At same time, one can adopt the Ratio Test method or the Cross Test method for improving the point-pair accuracy.

With regard to the source contour curve, we start with the first point and continue to its neighbor recursively, until algorithm reaches the end point. It is executed simultaneously the process on the target contour curve.

Interpolating two contour curves. Assumed λ ($\lambda = 0,1,2...,\lambda_{max} - 1$, λ_{max} stands for the maximum number of interpolation) stands for the interpolation index; $P_A^i \in C_{Source}$ $\left(i = 0,1,2...,n_1 - 1\right)$, $P_B^j \in C_{Target}$ $\left(j = 0,1,2...,n_1 - 1\right)$, n_1 is the point-pair number. P_A^i , P_B^j stands for the high confidence point, respectively; i, j is the index of the point, respectively; Q stands for a set of expansion point; Q_{sub} stands for a sub-set of expansion point; P_{inter} stands for the interpolation point ; γ stands for the Euclidean distance of neighboring points in Q_{sub} , γ_{max} stands for the maximum distance of neighboring points $\left(P_A^i, P_B^j\right)$. We obtain Q by the following steps:

Firstly, we obtain the interpolation point P_{inter} by Eq.(3):

$$P_{inter} = P_A^i + \alpha\left(P_B^j \text{-} P_A^i\right) \cdot \qquad \textbf{(3)}$$

where $\alpha \in [0,1]$. And then P_{inter} is inserted into S_{sub} .

Secondly, we let $Q = Q \cup Q_{sub}$.

The two steps are repeated until the end condition is reached. Q is obtained by Eq.(4):

$$Q = C_{Source} + \beta\left(C_{Target} \text{-} C_{Source}\right) \cdot \qquad \textbf{(4)}$$

where $\beta \in [0,1]$.

Note that we can find reverse curvature points of contour more precise than the current one-to-one pairing approach. But we aim at obtaining a good efficiency; our framework does not adopt these approaches, such as Bilinear Interpolation and Cubic Spline Interpolation.

λ_{max} and γ_{max} together determine the denseness of the model. The process of interpolation between two points will stop when the condition is satisfactory. The condition is if γ is less than the maximum distance γ_{max} or if λ exceeds the maximum number λ_{max} . Therefore the number of interpolation is different for the different point-pair $\left(P_A^i, P_B^j\right)$.

After interpolating all the contour curve of the sparse model, it turns into the quasi-dense one.

Removing the outliers. To remove the outliers, some points need to be removed. For the purpose, we adopt the method mentioned in section 3.1.3. In addition, it is important to distinguish high confidence point or expansion point from all point of point cloud during removing the outliers. The reason is that if the point is the high confidence point, we did not need to judge the location relationship between the point and the model.

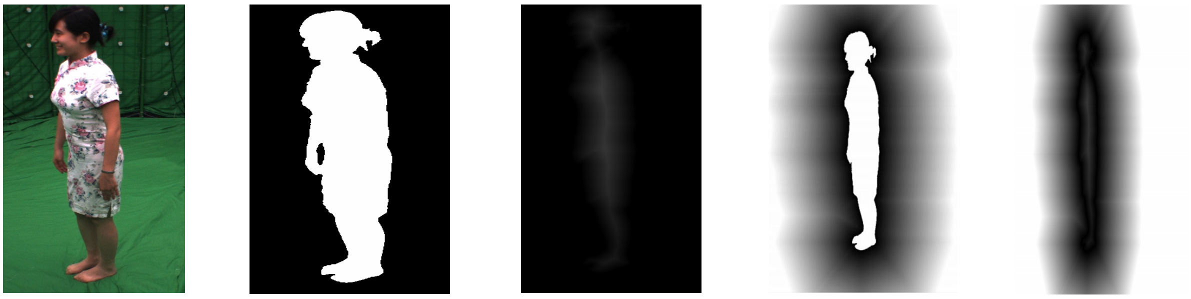
At last, we project the quasi-dense point cloud onto corresponding multi-view image so as to attain the quasi-dense point cloud with colored information. Refer to the paper [26] for its details.

3.3 Tracking Quasi-dense Point Cloud Sequence Based on Temporal-Contour Consistency

3.3.1 Obtaining Interpolation Silhouette Images

In order to increase the temporal coherence of model sequence, silhouette images are interpolated between key frames for each view. We obtain the interpolation silhouette images by the Distance Field images (DF-images). The basic procedure is as follows:

Obtaining DF-image. Firstly, we obtain the silhouette images from multi-view video at frame *t*, and then extract silhouette contour. Secondly, we convert the images which have silhouette contour to the binary silhouette image, and then compute the DF-image based on the distance transform. The result is shown in Fig. 4.



(a) the original input image

(b) the silhouette image

(c) the inner distance field

(d) the outer distance field

(e) the combined distance field

Fig. 4.

Note that the binary silhouette image must have good quality; otherwise the quality of DF-images will be affected.

Interpolating distance field. Interpolating distance field is an integral and important part of our shape tracking framework pipeline. We interpolate distance field between neighboring frames to get interpolation silhouette image, and then use interpolated DF-image to compute point cloud that are the in-between model between key models.

We get two DF-images which correspond to the same view, namely source DF-image and target one. Assumed $I_k(i,j)$ is source DF-image and $I_{k+1}(i,j)$ is target one. Our method implements the interpolation between source pixels and target ones by Eq.(5):

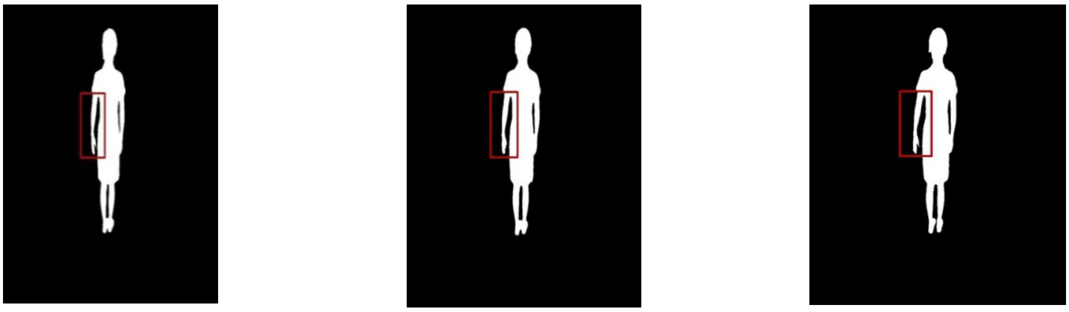
$$I_\eta(i,j) = \left[(1-\eta)\times I_k(i,j)\right] + \left[\eta \times I_{k+1}(i,j)\right]. \quad \textbf{(5)}$$

where $\eta = \left(k/\left(\lambda_{max}-1\right)\right)$, η stands for the progress of transport, it is transition rates, k stands for the interpolation index. The source distance field and target one are warped one towards the other, and then produce the interpolation distance field. In order to decrease the computation time and alleviate numerical instabilities, our method records the pixel value of DF-image using the interpolation matrix, instead of interpolation DF-image.

After the procedure, we obtain the interpolation silhouette image and the interpretation matrix.

Obtaining interpolation silhouette contour. Once the interpolation DF-image is available, the interpolation silhouette image can be determined by the zero points of the distance field or, in the discretized version, by the boundary between the positive and negative valued lattice points, whereas the pixels itself (its interior) is defined as the set of all negative valued points.

As is shown in Fig. 5, the interpolation silhouette contour is obtained. The in-between point cloud is computed from these contours by the method [24].



(a) the silhouette at frame t

(b) the interpolation silhouette between (a) and (c)

(c) the silhouette at frame $t+1$

Fig. 5.

3.3.2 Temporal-Contour-Consistency theory

The Temporal-Contour-Consistency is similar to the frame consistency which is used in many fields. In the video sequences, there is strong correlation in the same inter-frame scene. Motion estimation technology used the time redundancy to improve compressing efficiency in video compression field [24].

3.3.3 Tracking Quasi-dense Point Cloud Sequence

Our shape tracking framework is based on interpolating neighboring quasi-dense point cloud. It consists of two steps: finding corresponding point-pair and interpolating between neighboring models. Given two point cloud O_{Source}, O_{Target} consists of M contour curve, respectively; Assumed $C^{i}_{Source} \subseteq O_{Source}$ $(i = 0,1,2...M-1)$, $C^{j}_{Target} \subseteq O_{Target} (j = 0,1,2...N-1)$; τ stands for the progress of transport. The tracking method is similar to the method mentioned in section 3.2.2. Note that the source point clouds or target ones includes the computed point cloud from the interpolated silhouettes.

After interpolating all point-pairs of the neighboring point cloud, the Spatio-temporally coherent point cloud sequence is obtained. The main distinction of our shape tracking framework is that it does not complicatedly and strictly match between any two models.

4 Experimental Results and Analysis

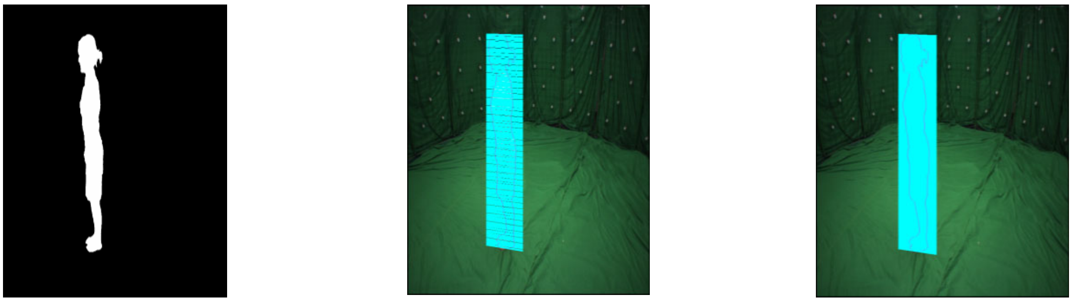
For the purposes of testing, we have conducted experiments to demonstrate the performance of our Spatio-Temporal-Contour-Consistency-Based shape tracking framework on the dataset publicly available datasets “Cheongsam”, “Redskirt”, “Redshirt” [13] and “Lady Dance” [1]. “Cheongsam” is the video of real-life performances and has 20 views. The image (video) sequence is collected at 25 fps. The resolution of images used for reconstruction is 1024 by 768. It is a short sequence (20 frames) with various types of actions such as standing, turning, and squatting. “Redskirt” and “Redshirt” are similar to “Cheongsam”. The dataset “Lady Dance” of 3D photography collection has 8 images from 8 viewpoints. We select 6 viewpoints in order to illustrate our method.

All the experiments are run on a desktop. Hardware for the experiments is: Intel(R) Core Duo CPU E8500 with frequency of 3.16 GHz, graphics card of ATI Radeon HD 3450-Dell Optiplex, memory of 14 GB. Software for the experiments is: OpenCV, C++. The exact camera parameters are known a priori. Therefore we can assess the performance of our framework in approximately ideal conditions.

[1] http://4drepository.inrialpes.fr/public/datasets

4.1 Expanding Sparse Point Cloud

In order to improve the accuracy of the intersection between epipolar line and contour, we adopt the method mentioned in section 3.1.1. As is illustrated in Fig. 6, our proposed method is more accurate than traditional one.



(a) the silhouette image

(b) the intersection (blue) using traditional method

(c) the intersection (blue) our proposed method

Fig. 6.

To assess the efficiency of expanding sparse point cloud by Spatio-Contour-Consistency, we complete the interpolation between neighboring contours on the sparse point cloud. The result is shown in Fig. 7.

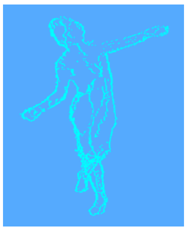
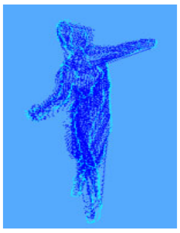
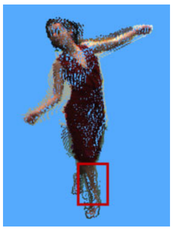
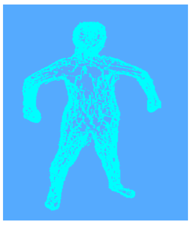
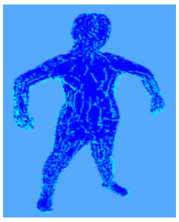
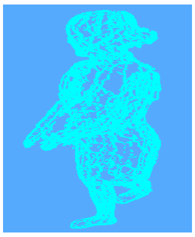
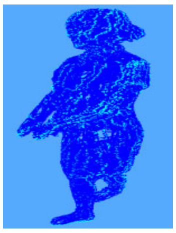
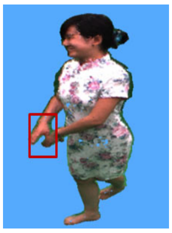
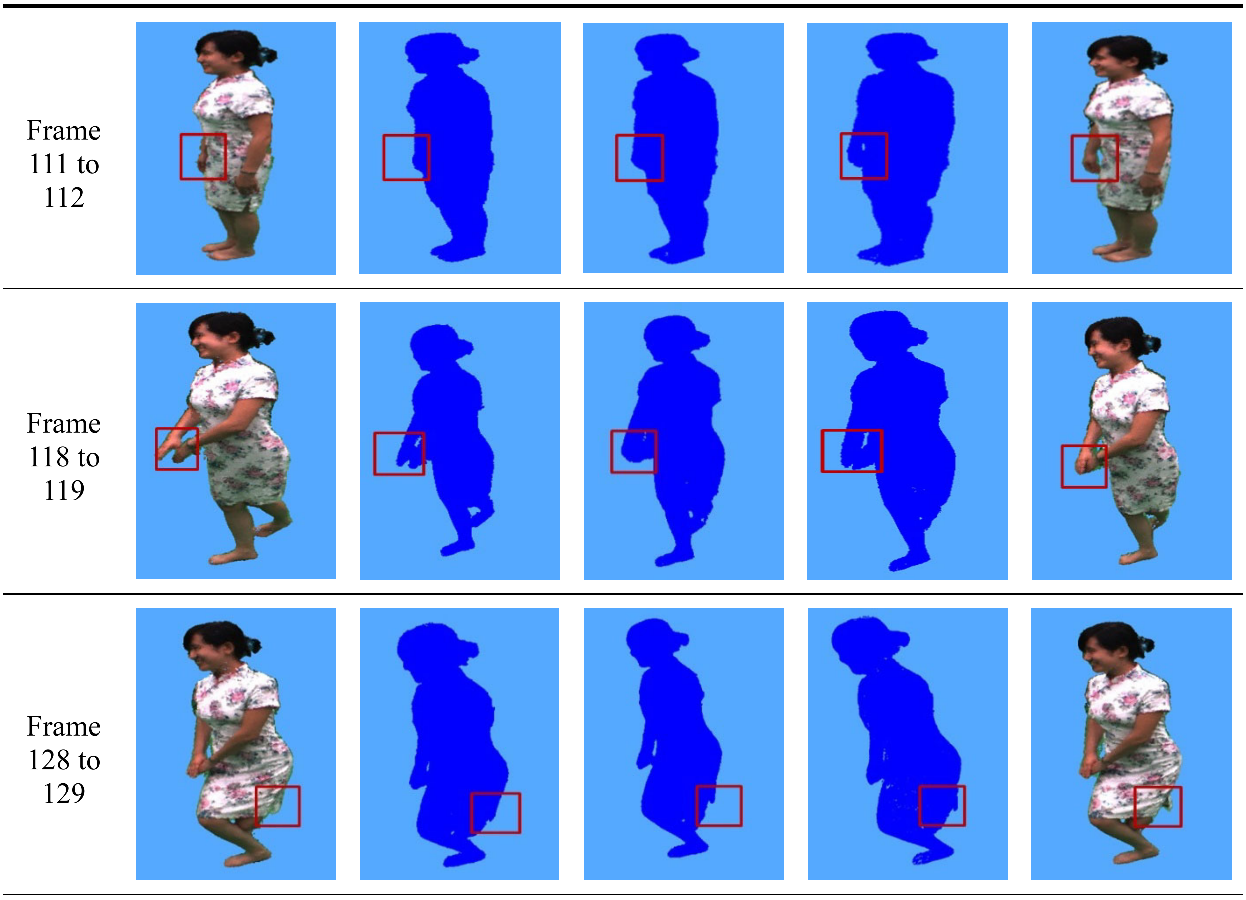
Dataset	λ_{max}	γ_{max}	Sparse point cloud	Quasi-dense point cloud	Coloured quasi- dense point cloud
Lady Dance	5	0.05			
Redshirt	5	0.01			
Cheongsam	10	0.05			
			(a) the sparse point cloud which consists of these high confidence points (Green)	the quasi-dense point cloud which consists of these high confidence points (Green) and these expansion points (Blue)	(c) the quasi-dense point cloud with coloured information. The denseness is different on the red rectangle area

Fig. 7.

As is shown in Fig. 7, the quality of the model of the dataset “Lady Dance” is poor. The reason is that the consistency between neighboring views is poor because we select only 6 views from the dataset. But the consistency will rise along with the increase of the view number. The quality of the model of the dataset “Cheongsam” and “Redshirt” testify the conclusion.Note that the reconstruction time increases with the number of pixels in all the contours. The demand about the denseness determines the value λ_{max} and γ_{max}. For example, the denseness of the point cloud will rise along with decreasing the value γ_{max}; but the reconstruction is more time-consuming. As is stated in Section 3.2.2, the number of interpolation is different for the different point-pair $\left(P^i_{Source}, P^j_{Target}\right)$. In other words, it is adaptive.

4.2 Tracking Quasi-dense Point Cloud

In our experiment, we set $\lambda_{max} = 5$, $\gamma_{max} = 0.01$, $\tau = 0.5$. The result is shown in Fig. 8. We display samples from the point cloud sequence. Point cloud transports from frame 111 to 112, frame 118 to 119 and frame 128 to 129, respectively. These frames stand for the typical deformation, such as standing, turning and squatting.



(a) the source quasi-dense point cloud

(b) the interpolation point cloud between (a) and (c)

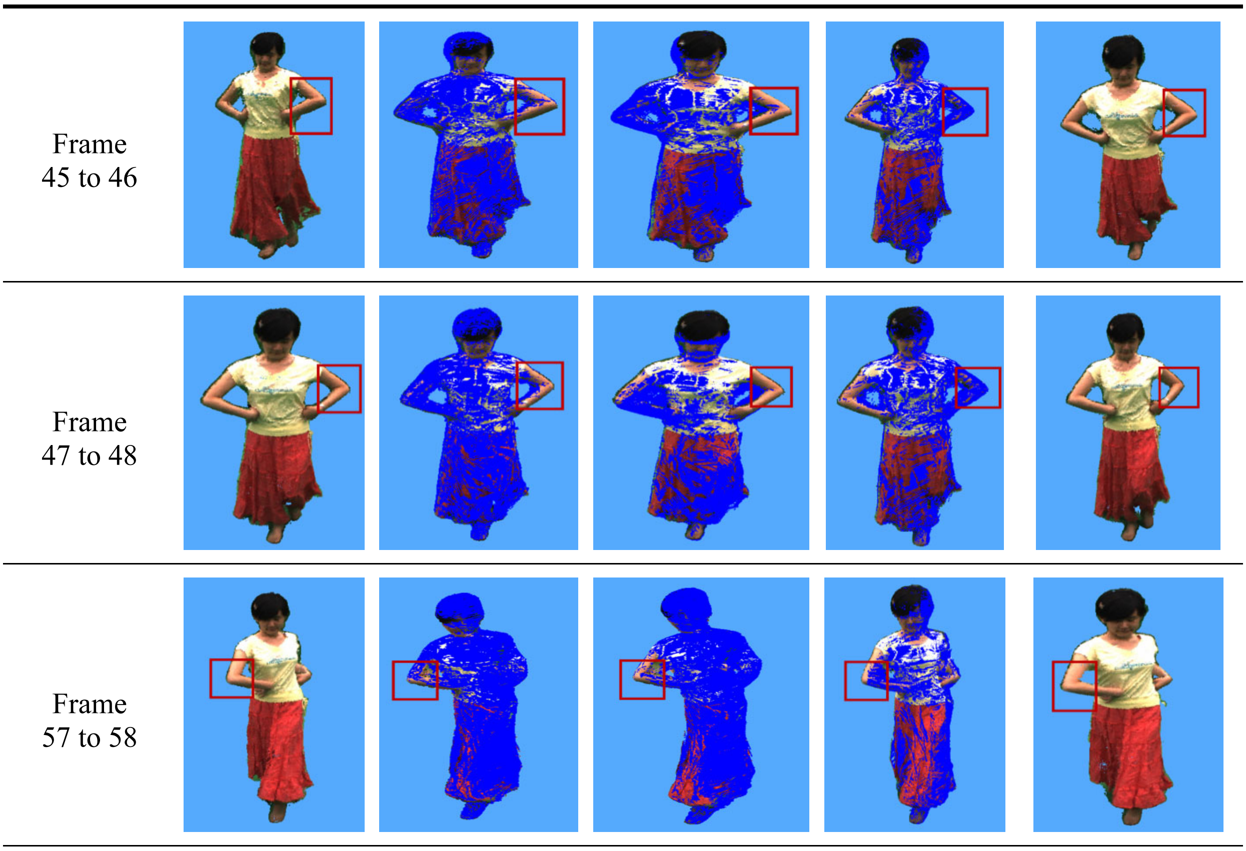
(c) the point cloud based on interpolation silhouette image

(d) the interpolation point cloud between (c) and (e)

(e) the target quasi-dense point cloud

Fig. 8.

In order to test the performance on different video sequences, we have conducted experiments on the dataset “Redskirt”. The datasets is a short sequence showing medium speed dance moves, and thus offers good opportunity to verify the tracking stability. The transportation process between neighboring models is illustrated in Fig. 9.



(a) the source quasi-dense model

(b) the overlap between interpolation quasi-dense model (Blue) and source one

(c) the overlap between quasi-dense model based on interpolation silhouette (Blue) and source one

(d) the overlap between interpolation quasi-dense model (Blue) and target one

(e) the target quasi-dense model

Fig. 9. Point cloud transport from Frame 45 to 46, 47 to 48, and 57 to 58

As is shown in Fig. 8 and Fig. 9, the source shape gradually transports to maximize its similarity with target one. Some readers may observe that the result from Frame 45 to 46 is similar to the one from Frame 47 to 48. The reason is that the four frames are neighboring. If they observe carefully the red rectangle area, they will find that the result is slightly different. The result reflects the performance of our framework.

4.3 Qualitative Assessment

We now analyze the average reconstruction time through the sequence. A favorable comparison to state of the art methods is listed in Table 1. It shows quantitative evaluation on the reconstruction time per frame. Our method shows competitive performance on through the sequence.

Table 1. Reconstruction time per frame through the sequence

Dataset	Method	Time (s)
Lady Dance	Bilir, et al. [7]	9.32
	Allain, et al. [27]	9.61
	Mustafa, et al. [11]	10.75
	Our method	6.43
Cheongsam	Liu, et al. [13]	142.68
	Bilir, et al [7]	67.41
	Allain, et al. [27]	80.85
	Our method	33.16
Redskirt	Liu, et al. [13]	145.45
	Bilir, et al. [7]	68.44
	Allain, et al. [27]	80.78
	Our method	34.65

Note that the temporal coherence will rise with increasing the value τ between neighboring models.

To obtain a quantitative analysis of the quality of the model, an experiment on the datasets “Cheongsam”, “Redskirt”, and “Lady Dance” was performed, respectively. Table 2 lists the accuracy (Acc.) and completeness (Comp.) with respect to the ground truth model. Note that we set $\lambda_{max}=5$, $\gamma_{max}=0.01$, and $\tau=0.5$. The index of the model is shown in parentheses in the table header.

A suitable choice of the parameter λ_{max}, γ_{max} and τ is very important. They affect the Spatio-temporal coherence of point cloud sequence as well as the reconstruction time. Generally speaking, as λ_{max} increases, τ increases or γ_{max} decreases, the Spatio-temporal coherence of the model sequence will increase. To obtain balance between the speed and the accuracy, one should select an appropriate value for these parameters.

Table 2. Accuracy and completeness for different datasets and methods

Method	Cheongsam (111)		Redskirt (45)		Lady Dance (00)	
	Acc.	Comp.	Acc.	Comp.	Acc.	Comp.
Liu, et al. [13]	0.65	98.1%	0.70	97.9%	0.79	86.2%
Bilir, et al. [7]	0.72	97.8%	0.79	98.3%	0.85	86.8%
Allain, et al. [27]	0.59	99.0%	0.64	99.2%	0.77	87.1%
Our method	0.91	95.2%	0.96	94.1%	1.01	83.4%

5 Conclusions

We have proposed and tested the automatic Temporal-Spatio-Contour-Consistency-based framework to track the time-varying shape of a dynamic object from its multi-view video. The main distinction of the framework is that it does not complicatedly and strictly match between any two contours on the same model or the neighboring model. The shape of the dynamic object is tracked correctly via point cloud transports based solely on image cues. We can obtain a higher frame rate, Spatio-temporal-coherence and quasi-dense point cloud sequence with color information. The advantage of the framework can be summarized as follows:

(1) It expands the sparse point cloud using Spatio-Contour consistency. By this means, one can rapidly obtain a quasi-dense point cloud sequence.

(2) It interpolates the quasi-dense point cloud sequence between consecutive models using Temporal-Contour consistency. By this means, one can obtain the point cloud sequence with the better Spatio-temporal coherence.

(3) It obtains robustly the sub-pixel intersection between the epipolar line and the contour edge. By this means, one laid the foundation for higher reconstruction quality, especially the dynamic object exhibiting complex topology.

For the algorithm to work successfully, there is generally a compromise between the frame rate, the speed of the motion and the accuracy of the shape details. Note that our framework is unsuitable for the application in which the model sequence with the high accuracy must be met. By implementing our approach in parallel, the execution time will be decreased. Therefore, in future work we will do it.

Acknowledgements

Thanks are specially given to the anonymous reviewers for their helpful suggestions. This work is supported by the Science and Technology Joint Foundation of Guizhou Province of China (No. 20147639), the Science and Technology Foundation of Guizhou Province of China (No. 20132094), the National Natural Science Fund of China (No. 61562009) and the National Natural Science Fund of China (No. 61462013).

References

[1] N. Ahmed, I.-N. Junejo, A system for 3D video acquisition and spatio-temporally coherent 3D animation reconstruction using multiple RGB-D cameras, International Journal of Signal Processing, Image Processing and Pattern Recognition 6(2)(2013) 113-128.

[2] N. Ahmed, C. Theobalt, C. Rossl, S. Thrun, H.-P Seidel, Dense correspondence finding for parametrization-free animation reconstruction from video, in: Proc. 26th IEEE Conference on Computer Vision and Pattern Recognition, CVPR, 2008.

[3] Z. Zhang, H.-S. Seah, C.-K. Quah, A multiple camera system with real-time volume reconstruction for articulated skele- ton pose tracking, in: Proc.17th International Multimedia Modeling Conference, 2011.

[4] J. Li, Z. Cheng, H. Li, Y. Chen, W. Jiang, G. Dang, S. Jin, A non-rigid registration algorithm of large-scale deformable models, Jisuanji Xuebao/Chinese Journal of Computers 34(3)(2011) 539-547.

[5] D. Vlasic, I. Baran, W. Matusik, J. Popovic, Articulated mesh animation from multi-view silhouettes, in: Proc. SIGGRAPH'08, 2008.

[6] K. Li, Q. Dai, W. Xu, Markerless shape and motion capture from multiview video sequences, IEEE Transactions on Circuits and Systems for Video Technology 21(3)(2011) 320-334.

[7] S.-C. Bilir, Y. Yemez, Non-rigid 3D shape tracking from multiview video, Computer Vision and Image Understanding 116(2012) 1121-1134.

[8] J. Sucmuth, M. WinterG. Greiner, Reconstructing animated meshes from time-varying point clouds, Computer Graphics Forum 27(5)(2008) 1469-1476.

[9] R.-A. Newcombe, D. FoxS.-M. Seitz, Dynamic fusion: reconstruction and tracking of non-rigid scenes in real-time, in: Proc. Conference on Computer Vision and Pattern Recognition, 2015.

[10] M. Vo, S.-G. Narasimhan, Y. Sheikh, Spatiotemporal bundle adjustment for dynamic 3D reconstruction, in: Proc. International Conference on Computer Vision and Pattern Recognition, 2016.

[11] A. Mustafa, H. Kim, J. Guillemaut, A. Hilton, Temporally coherent 4D reconstruction of complex dynamic scenes, in: Proc. International Conference on Computer Vision and Pattern Recognition, 2016.

[12] H. Nakajima, Y. Makihara, H. Hsu, I. Mitsugami, M. Nakazawa, H. Yamazoe, H. HabeY. Yagi, Point cloud transport, in: Proc. 21st International Conference on Pattern Recognition, 2012.

[13] Y. Liu, Q. Dai, W. Xu, A point-cloud-based multiview stereo algorithm for free-viewpoint video, IEEE Transactions on Visualization and Computer Graphics 16(3)(2010) 407-418.

[14] L.-Ç.-C. Ik, M. Uller, Point cloud morphing, in: Proc. Central European Seminar on Computer Graphics, 2003.

[15] M. Shinya, Unifying measured point sequences of deforming objects, in: Proc. 2nd International Symposium on 3D Data Processing, Visualization, and Transmission, 2004.

[16] H. Qiu, L. Chen, Research and development of point-based computer graphics, Computer Science 36(6)(2009) 10-15.

[17] Y. Zhao, G. Liu, Q. Liu, As-rigid-as-possible efficient deformation algorithm for point clouds, J. Comput.-Aided Des. Comput. Graph 25(7)(2013) 955-962.

[18] K. Guo, F. Xu, Y. Wang, Y. B. Liu, Robust non-rigid motion tracking and surface reconstruction using L0 regularization, IEEE Transactions on Visualization & Computer Graphics 99(2017) 1-14.

[19] D. Cohen-Or, D. Levin, A. Solomovici, Three-dimensional distance field metamorphosis, ACM Transactions on Graphics 17(2)(1998) 116-141.

[20] Y. Yemez, F. Schmitt, 3d reconstruction of real objects with high resolution shape and texture, Image and Vision Computing 22(13)(2004) 1137-1153.

[21] L. Yang, Q. Yan, Y. Fu, C. Xiao, Surface reconstruction via fusing sparse-sequence of depth images, IEEE Transactions on Visualization & Computer Graphics PP(99)(2017)1-1.

[22] M. Ji, J. Gall, H. Zheng, Y. B. Liu, F. Lu, SurfaceNet: an end-to-end 3D neural network for multi-view stereopsis, in: Proc. IEEE International Conference on Computer Vision, 2017.

[23] N. Matsushiro, New digital differential analyzer for circle generation, IEICE Transactions on Information and Systems E81-D (2)(1998) 239-242.

[24] C.H. Huang, D.M. Lu, C.Y. Diao, Accelerated visual hulls of complex objects using contribution weights, in: Proc. 2013 7th International Conference on Image and Graphics, 2013.

[25] J.-B. Franco, E. Boyer, Efficient polyhedral modeling from silhouettes, IEEE Transactions on Pattern Analysis and Machine Intelligence 31(3)(2009) 414-427.

[26] C.H. Huang, D.M. Lu, C.Y. Diao, Accelerated visual hulls of complex objects using contribution weights, J. Comput.-Aided Des. Comput. Graph. 26(8)(2014) 1297-1302.

[27] B. Allain, J. Franco, E. Boyer, An efficient volumetric framework for shape tracking, in: Proc. IEEE International Conference on Computer Vision and Pattern Recognition, 2015.