EFFICIENT GRAPH-BASED MATRIX COMPLETION ON INCOMPLETE ANIMATED MODELS

Evangelos Vlachos[1], *Aris S. Lalos*[1], *Konstantinos Moustakas*[1] *and Kostas Berberidis*[2]

[1] Electrical and Computer Engineering Department, [2] Computer Engineering and Informatics Department
University of Patras, GR-26500, Rion-Patras, Greece
Emails: {vlaxose, aris.lalos, moustakas}@ece.upatras.gr, berberid@ceid.upatras.gr

ABSTRACT

Recently, there has been increasing interest for easy and reliable generation of 3D animated models facilitating several real-time applications. In most of these applications, the reconstruction of soft body animations is based on time-varying point clouds which are irregularly sampled and highly incomplete. To overcome these imperfections, we introduce a novel reconstruction technique, using graph-based matrix completion approaches. The presented method exploits spatio-temporal coherences by implicitly forcing the proximity of the adjacent 3D points in time and space. The proposed constraints are modeled by using the weighted Laplacian graphs and are constructed from the available points. Extensive evaluation studies, carried out using a collection of different highly-incomplete dynamic models, verify that the proposed technique achieves plausible reconstruction output despite the constraints posed by arbitrarily complex and motion scenarios.

Index Terms— incomplete dynamic point clouds, point cloud reconstruction, animated models, matrix completion

1. INTRODUCTION

Nowadays, commodity real-time 3D capturing devices are widely used for acquisition of dynamically deforming shapes at sustained video rates. Therefore, there has been increasing interest for easy and reliable generation of 3D animated meshes facilitating several real-time applications in fields ranging from film making and gaming to engineering (e.g., Microsoft holoportation) and medicine (e.g., in radiation therapies, the pre-identified malignant tumors need to be constantly updated using 3D surface capture). Although the resolution and accuracy of the new generation image sensors and scanning techniques are constantly improving the generated dynamic meshes usually contain a variety of imperfections that pose significant challenges for reconstruction of dynamic shapes, like noise, outliers, sampling density, misalignment and missing data [1].

A large number of prior works [2] has investigated the problem of completion in static geometries, resulting in excellent filled static meshes. However, their direct application to every frame separately is usually causing incorrect topologies and temporally incoherent surfaces. A common approach to produce a temporally consistent dynamic mesh is to use a template prior [3]. While the general animation can be captured adequately, geometric details are limited to those in the template. Hence, the deformation of a generic or a user specified template fail in modeling fine-scale dynamics that could be captured by exploiting the embedding low dimensional structure that exist in the deformable animated models [4].

Due to hardware impairments and/or environmental conditions, usually the geometry data have several missing points. In this case, when the frames of static meshes are stacked together as row vectors, the resulting matrix contains several zero entries. Recently, a new signal processing technique termed as matrix completion (MC) [5] has been successfully applied to several computer vision problems. One of the strong aspects of this theory is the provided performance guarantees, which state that the missing information can be *exactly* recovered provided that the incomplete matrix is low-rank and that a lower bound for the known entries is satisfied. Applications of the matrix completion techniques include the recovery of occluded faces [6], and the face image alignment [7]. Moreover, they have been successfully used for the fusion of point clouds from multiview images of the same object [8].

Building on this direction, in this work, we introduce a novel technique for reconstruction of highly incomplete dynamic models. Usually, the matrices with the dynamic geometry data of the animated models exhibit a low-rank due to spatio-temporal coherences. Therefore matrix completion is a viable reconstruction solution. To the best of our knowledge, matrix completion algorithms have never been applied to the animation completion problem, despite their wide success on a large range of computer vision applications. In summary, the main contributions of this work are the following:

- We formulate the reconstruction of the animated model graphical structure as a convex optimization problem, and we provide a uniqueness guarantee for the solution.
- We provide an efficient graph-based matrix completion algorithm which is able to recover the missing entries

978-1-5090-6067-2/17/$31.00 ©2017 IEEE

of highly incomplete time-varying point clouds, capturing the underlying spatio-temporal structure of the geometry data.

- We show via experimental evaluation under different configurations and animation models, that the high-frequency details can be adequately recovered from a highly incomplete geometry dataset.

2. BACKGROUND ON MATRIX COMPLETION

In order to optimally follow the proposed framework, in this section we briefly review some background information regarding the matrix completion theory.

Matrix completion [5] refers to the procedure of recovering a low-rank matrix from a sampling of its entries, which formally, can be written as:

$$\min_{\mathbf{X}} rank(\mathbf{X}) \text{ subject to } [X]_{ij} = [M]_{ij}\ (i,j) \in \Omega \qquad (1)$$

where $\mathbf{M} \in \mathbb{R}^{3F \times N}$ is the complete matrix, $\mathbf{X} \in \mathbb{R}^{3F \times N}$ is the optimization variable and $[X]_{ij}$ denotes the element of the matrix $\mathbf{X}$ at the i-th row and j-th column. $\Omega = \{(i_k, j_k), k = 1, \ldots, K\}$ is the set with the matrix indices of the K non-zero entries, where $i_k = 1, \ldots, 3F$ and $j_k = 1, \ldots, N$. In [5] it has been proven that most matrices $\mathbf{M}$ of rank R can be perfectly recovered by solving the nuclear-norm optimization problem

$$\min_{\mathbf{X}} \tau \|\mathbf{X}\|_* \text{ subject to } [X]_{ij} = [M]_{ij}\ (i,j) \in \Omega \qquad (2)$$

where $\|\mathbf{X}\|_* = \sum_{k=1}^{R} \sigma_k(\mathbf{X})$ denotes the nuclear norm of the matrix $\mathbf{X}$ with $\sigma_k(\mathbf{X})$ the k-th singular value of the matrix, and $\tau \geq 0$ a weighting parameter which depends on the rank of the matrix $\mathbf{M}$. Note that, a large value of τ will impose low rank to the recovered matrix. Theorem 1.1 in [5] states that given that the matrix $\mathbf{M}$ with rank R is randomly sampled, keeping only $3F \geq c_1 N^{5/4} R log(N)$ entries, then with probability $1 - c_2 N^{-3} logN$ the minimizer of (2) is equal to $\mathbf{M}$, where c_1, c_2 are numerical constants. This minimizer is provided by the singular value thresholding (SVT) algorithm [9], denoted as $\mathcal{D}_\tau(\mathbf{Y})$, where $\mathbf{Y} \in \mathbb{R}^{3F \times N}$ is the dual variable of the Lagrangian of (2). Specifically, let $\mathbf{Y} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$ be the singular value decomposition (SVD) of a matrix $\mathbf{Y}$, where $\mathbf{U}$, $\mathbf{V}$ are orthonormal matrices. Then, the SVT operator is

$$\mathcal{D}_\tau(\mathbf{Y}) = \mathbf{U} diag\big(\{(\sigma_i - \tau)_+\}_{1 \leq i \leq R}\big)\mathbf{V}^T \qquad (3)$$

where $(x)_+ = \max(0, x)$ via successive application of the SVT operator the singular values with $\sigma_i < \tau$ are replaced by zero. Therefore, the minimizer is provided after L_{max} iterations of the SVT algorithm, i.e.

1: **for** $\ell = 1, \ldots, L_{max}$ **do**
2: $\quad \mathbf{X}_\ell = \mathcal{D}_\tau(\mathbf{Y}_{\ell-1})$
3: $\quad \mathbf{Y}_\ell = \mathbf{Y}_{\ell-1} + \mathbf{E}_\Omega \circ (\mathbf{M} - \mathbf{X})$
4: **end for**

with $\mathbf{E}_\Omega$ denotes the matrix where its (i,j)-th component is equal to 1 if $(i,j) \in \Omega$ and zero otherwise, and $\circ$ denotes the Hadamard (element-wise) product.

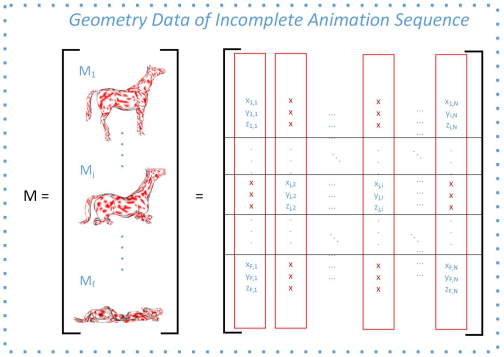

Fig. 1. An example of a highly incomplete dynamic mesh (only the 30% of the total points is known).

3. PROPOSED APPROACH

In this section, first we introduce the graph-based representation of the spatio-temporal coherence of the geometry data, which is exploited afterwards for the introduction of the proposed graph-based matrix completion technique.

3.1. Representation of the spatio-temporal structure

The geometric structure of the soft-body animation models provides that the adjacent 3D points in space and time will have "close" geometry data. This prior information can be incorporated into the matrix completion problem formulation so as to jointly exploit the low-rank property of the point cloud matrix as well as the geometric structure of the points in space and time. Usually, the underlying graphical structure is expressed using the graph Laplacians of the rows and the columns.

Let us assume that, the time-varying point clouds is represented by the matrix $\mathbf{M} \in \mathbb{R}^{3F \times N}$, which is a collection of a number of F 3D frames with N vertices for each frame, as shown in Fig. 1. Specifically, each triad of rows corresponds to the x, y, z- coordinates for time frame. The undirected weighted row graph $\mathbf{G}_r = (V_r, E_r, W_r)$ where $V_r = \{1, \ldots, 3F\}$ is the set of vertices and $E_r \subseteq V_r \times V_r$ is the set of edges, while $\mathbf{W}_r$ represents a $3F \times 3F$ matrix with non-negative weights. The graph Laplacian $\mathbf{L}_r$ for the rows of $\mathbf{M}$ is defined as $\mathbf{L}_r = \mathbf{I}_{3F} - \mathbf{D}^{-1}\mathbf{A}$, where $\mathbf{I}_{3F}$ is the $3F \times 3F$ identity matrix, $\mathbf{A} \in \mathbb{R}^{3F \times 3F}$ is the connectivity matrix of the mesh with elements $\mathbf{A}_{(i,j)} = 1$ when $(i,j) \in E_r$ and 0 otherwise, while $\mathbf{D}$ is the diagonal matrix with $\mathbf{D}_{(i,i)} = |N(i)|$ and $N(i) = \{j \,|\, (i,j) \in E_r\}$ is a set with the immediate neighbors for node i. The graph of the

columns of $\mathbf{M}$, $\mathbf{G}_c = (V_c, E_c, W_c)$ and the Laplacian $\mathbf{L}_c$ are defined respectively.

For the construction of these weighted graphs, we consider the techniques: $\epsilon \in \mathbb{R}$ neighborhoods (ϵ-N) and $k \in \mathbb{N}$ nearest neighbours (k-NN). It is known that, ϵ-N graphs are symmetric and are more meaningful, although depending on the choice of the parameter ϵ they could lead to heavy or disconnected graphs. On the other hand, the choice of the parameter for k-NN graphs is more straightforward, usually leading to connected graphs. For the weighting of the graphs, two different approaches can be applied: either the *binary*, where the weight between the i-th and the j-th nodes is 1, i.e., $w_{i,j} = 1$, if and only if vertices i and j are connected by an edge, or the *thresholded Gaussian kernel*, where if the nodes i and j are connected, then $w_{i,j} = e^{\frac{-\|x_i - x_j\|_2^2}{2\theta^2}}$ and 0 otherwise, where θ is the variance which sets the threshold for edge existence.

To express the spatial coherence of the structure, we regard $\mathbf{M}$ as a collection of $3F$-dimensional column vectors $\mathbf{m}_j \in \mathbb{R}^{3F\times 1}, j = 1, \ldots, N$, where $\mathbf{m}_j \approx \mathbf{m}_k$ if $(j,k) \in E_c$. More formally, this can be expressed as follows:

$$\|\mathbf{L}_c\mathbf{M}^T\|_F^2 = \sum_{j,k} w_{j,k}^c \|\mathbf{m}_j - \mathbf{m}_k\|_2^2. \tag{4}$$

Essentially, the expression $\|\mathbf{m}_j - \mathbf{m}_k\|_2^2$ measures the spatial distance of the time trajectories of the j and k 3D vertices, since the 3F vectors $\mathbf{m}_j, j = 1, \ldots N$ represent one 3D vertex for all time frames. Accordingly, to express the time coherence of the structure in space, we regard $\mathbf{M}^T$ as a collection of N-dimensional column vectors $\mathbf{v}_j \in \mathbb{R}^{N\times 1}, j = 1, \ldots, 3F$, where $\mathbf{v}_j \approx \mathbf{v}_k$ if $(j,k) \in E_r$. More formally, this can be expressed as:

$$\|\mathbf{L}_r\mathbf{M}\|_F^2 = \sum_{j,k} w_{j,k}^r \|\mathbf{v}_j - \mathbf{v}_k\|_2^2. \tag{5}$$

Note that, the expression $\|\mathbf{v}_j - \mathbf{v}_k\|_2^2$ measures the distance of the whole point cloud between two different time frames.

3.2. Graph-based matrix completion

The underlying structure can be embedded into problem (2) according to the following formulation [10]:

$$\min_{\mathbf{X}} \frac{1}{2}\|\mathbf{E}_\Omega \circ (\mathbf{X} - \mathbf{M})\|_F + \tau\|\mathbf{X}\|_* + \frac{\gamma_c}{2}\|\mathbf{L}_c\mathbf{X}^T\|_F^2 + \frac{\gamma_r}{2}\|\mathbf{L}_r\mathbf{X}\|_F^2 \tag{6}$$

where the first term of (6) minimizes the error between the known points and the recovered with $\|\mathbf{X}\|_F$ denoting the Frobenius norm of the matrix, while the term with the nuclear-norm imposes low rank to the recovered matrix, depending on the weighting parameter τ. Via the last two terms of (6), we demand that the columns corresponding to neighbouring nodes and the rows corresponding to the node movement to be "close" (in some sense) to each other; where λ_r and λ_c represent the associated regularization parameters of the graph Laplacians.

The Lagrangian of the equivalent splitting version of the optimization problem (6) can be expressed as $\mathcal{L}(\mathbf{X}, \mathbf{Y}, \mathbf{Z}) = \frac{1}{2}\|\mathbf{E}_\Omega \circ (\mathbf{Y} - \mathbf{M})\|_F^2 + \tau\|\mathbf{X}\|_* + \frac{\gamma_r}{2}\|\mathbf{Y}\|_{L_r} + \frac{\gamma_c}{2}\|\mathbf{Y}\|_{L_c} + \frac{\rho}{2}\|\mathbf{X} - \mathbf{Y}\|_F^2$, where $\mathbf{Y}, \mathbf{Z}$ are the dual variables and $\rho > 0$ is the penalty parameter.

A simple but powerful algorithm for solving large-scale constrained optimization problems, such as (6), is the method of Alternating Direction Method of Multipliers (ADMM) [11]. ADMM is based on a solid theory which provides performance guarantees for its convergence. It consists of the following iterations:

$$\mathbf{X}(\ell+1) = arg\min_{\mathbf{X}} \mathcal{L}(\mathbf{X}, \mathbf{Y}(\ell), \mathbf{Z}(\ell)) \tag{7}$$

$$\mathbf{Y}(\ell+1) = arg\min_{\mathbf{Y}} \mathcal{L}(\mathbf{X}(\ell+1), \mathbf{Y}, \mathbf{Z}(\ell)) \tag{8}$$

$$\mathbf{Z}(\ell+1) = \mathbf{Z}(\ell) + \rho(\mathbf{X}(\ell+1) - \mathbf{Y}(\ell+1)). \tag{9}$$

Considering these steps of ADMM for our case, the computation of eq. (9) is straightforward, while the minimizer of eq. (7) is given by the SVT operator (3), e.g.

$$\mathbf{X}(\ell+1) = \mathcal{D}_\tau\big(\mathbf{Y}(\ell) - \rho^{-1}\mathbf{Z}(\ell)\big). \tag{10}$$

Proposition 1. *The minimizer of* (8) *is obtained by the solution of the following system:*

$$\mathbf{E}_\Omega \circ (\mathbf{Y} - \mathbf{M}) + \gamma_r(\mathbf{L}_r^T\mathbf{L}_r)\mathbf{Y} + \gamma_c\mathbf{Y}(\mathbf{L}_c^T\mathbf{L}_c) + \rho(\mathbf{Y} - \mathbf{X} - \rho^{-1}\mathbf{Z}) = \mathbf{0}. \tag{11}$$

which is a unique solution for $\rho > 0$

Proof. The proof is given at the Appendix. □

The solution of (11) is obtained according to the steps: (a) First, (11) is expressed into the form

$$\mathbf{A}\mathbf{y} = vec(\mathbf{E}_\Omega \circ \mathbf{M} + \rho\mathbf{X} + \mathbf{Z}) \tag{12}$$

where $\mathbf{A} = \mathcal{D}(vec(\mathbf{E}_\Omega)) + \gamma_r(\mathbf{L}_r^T\mathbf{L}_r) \otimes \mathbf{I}_N + \gamma_c\mathbf{I}_{3F} \otimes (\mathbf{L}_c^T\mathbf{L}_c) + \rho\mathbf{I}_{3F\times N}$. (b) Then, (12) is solved and the estimated vector $\mathbf{y} \in \mathbb{R}^{3F\cdot N\times 1}$ is reshaped to the matrix $\mathbf{Y} \in \mathbb{R}^{3F\times N}$. (c) Finally, the vector $\mathbf{y}$ is reshaped back to the matrix form, i.e. $\mathbf{Y} = \mathcal{P}(\mathbf{y})$, where $\mathcal{P}$ denotes the transformation operator.

The graph-based matrix completion technique is summarized in Algorithm 1.

Computational Complexity: The costly steps of the Algorithm 1 are: i) the computation of the SVT operator (line 3) to sparse matrix $\mathbf{H}(\ell)$, and ii) the solution of a sparse system of dimensions $3F\cdot N \times 3F\cdot N$ (line 5). In general, the complexity of the SVD is $\mathcal{O}(NM^2)$ per iteration for $N > 3F$ [12]. However, the number of known entries is usually much lower than the total number of entries of the matrix $\mathbf{M}$, hence its dominant singular values and singular vectors can be efficiently computed via incomplete SVD methods, e.g. Lanczos bidiagonalization algorithm, or via subspace tracking techniques,

Algorithm 1 Graph-based matrix completion (GMC)

$\mathbf{A} \leftarrow \mathcal{D}(vec(\mathbf{E}_\Omega)) + \gamma_r(\mathbf{L}_r^T\mathbf{L}_r) \otimes \mathbf{I}_N + \gamma_c \mathbf{I}_{3F} \otimes (\mathbf{L}_c^T\mathbf{L}_c) + \rho\mathbf{I}_{3F \cdot N}$
for $\ell = 1, \ldots, \ell_{max}$ **do**
  $\mathbf{X}(\ell+1) \leftarrow \mathcal{D}_\tau\big(\mathbf{Y}(\ell) - \rho^{-1}\mathbf{Z}(\ell)\big)$
  $\mathcal{V} \leftarrow \alpha\mathbf{X}(\ell+1) - (1-\alpha)\mathbf{Y}(\ell)\big)$
  $\mathbf{b} \leftarrow vec(\mathbf{E}_\Omega \circ \mathbf{M} + \rho\mathcal{V} + \mathbf{Z}(\ell))$
  Solve the system $\mathbf{A}\mathbf{y} = \mathbf{b}$
  $\mathbf{Y}(\ell+1) \leftarrow \mathcal{P}(\mathbf{y})$
  $\mathbf{Z} \leftarrow \mathbf{Z}(\ell) + \rho(\mathbf{X}(\ell+1) - \mathbf{Y}(\ell+1))$
end for

thus the complexity can be reduced over to $\mathcal{O}(NMT)$, where $T \ll 3F, N$ is the truncation parameter. Considering the complexity of (12), the direct approach of matrix inversion is usually not suitable for this case, due to the potentially huge dimensions of this system and the high density of the inverse matrix $\mathbf{A}^{-1}$. Instead, iterative solvers can leverage the sparsity and achieve better computational performance (e.g., conjugate gradient (CG) algorithm can achieve $\mathcal{O}(kMN)$ for k-NN graphs). However, in the case where the Laplacian matrices have smaller sparsity (e.g. they correspond to graphs with more edges), the cost for the computation and storage of (8) can dominate the overall computational complexity.

Convergence speed: It is known that, when high quality reconstruction is required, the sequential implementation ADMM may suffer from low convergence speed [11]. To optimize the speed of the convergence, we utilize the *over-relaxation* technique [11], where the quantity $\rho\mathbf{X}$ in (12) is replaced by the following one: $\rho(\alpha\mathbf{X} - (1-\alpha)\mathbf{Y})$, where α is the relaxation parameter.

4. PERFORMANCE EVALUATION

In this section, we present the experimental evaluation of the proposed graph-based matrix completion approach on time-varying 3D point clouds. The algorithms have been implemented using the Julia scientific language. For our experiments we have used three animation models: the "Horse Collapse" model [13], with 50 3D frames ($3F = 150$) and $N = 8431$ vertices, the "Handstand" model [14] with 175 3D frames and $N = 10002$ vertices, and the "Ocean" model (a highly-deformed wave simulation), with with 1500 3D frames and $N = 2500$ vertices. We assume that the missing points have been obtained in a uniform random manner, and the xyz coordinates share the same sampling pattern. The reconstruction quality of the techniques is evaluated using the normalized mean root square error for all frames, $NMSE = \frac{\|\mathbf{X}-\mathbf{M}\|_F}{\|\mathbf{M}\|_F}$, and the KG error [15], which is defined as $KGE = 100 \cdot \frac{\|\mathbf{X}-\mathbf{M}\|_F}{\|\mathbf{X}-\mathcal{E}(\mathbf{M})\|_F}$, where $\mathcal{E}(\mathbf{M})$ denotes a matrix whose columns consist of the average vertex positions for all frames. For comparison purposes, we have also employed a conventional technique for the reconstruction of the animation models, namely the least-square meshes (LSM) algorithm [16]. LSM is described as the solution of an extended system of equations, $[\mathbf{L}_c^T\mathbf{I}_{N\times K}^T]^T\mathbf{x}_f = [\mathbf{0}^T, \mathbf{x}_{a,f}]$ for $f = 1, \ldots, 3F$, where $\mathbf{x}_{a,i}$ are the K known (*anchor*) points for the f-th frame. Note that, it can be shown that LSM is equivalent with the solution of (6) with $\tau = 0$.

Parameters selection: The convergence behaviour of the Algorithm 1 is determined by the parameters τ and ρ. Specifically, τ parameter determines the rank R of the reconstructed matrix. Hence, large values of τ will result into fast convergence but also lower reconstruction quality. On the contrary, small values of τ may result into better reconstruction quality but the algorithm could diverge due to the small sampling ratio $K(\leq N^{1.2}R\log N)$. Considering the parameter ρ, which influences the convergence speed of the ADMM technique, after experimental evaluation we have selected the value $\rho = 1/6$ for the following results.

Construction of the graph Laplacians: Obviously, the method for the construction of the spatio-temporal graph Laplacian matrices (e.g. ϵ-n or k-nn) has a major impact at the reconstruction performance. In particular, depeding on the ratio of the known entries different weights for the graph Laplacians are required. When the ratio is high, then the graph constraints has small impact, hence small weights are used. The opposite is applied for the cases of low ratio of known entries. Also it is important to note that, the estimation of the graph Laplacians from incomplete data may contain severe errors, even in the case of 10% missing entries. To overcome this problem and provide estimates of $\mathbf{L}_c$ and $\mathbf{L}_r$ for each animation matrix, we have employed the matrix completion algorithm (i.e. $\gamma_r = \gamma_c = 0$). Note that, since the graph constraints of (4) and (5) take the average distances in space and time, the estimation errors of this approach will have small impact to the overall performance.

Let us first evaluate the reconstruction quality of the MC techniques with respect to the ratio of the known points. In Fig. 2, we consider three reconstruction techniques: (a) the MC-based with $\gamma_c = \gamma_r = 0$, where the underlying structure of the meshes is not exploited, (b) the LSM technique, which exploits spatial coherence of the meshes, and (c) the proposed technique, which exploits the spatio-temporal coherence of the dynamic meshes. The results of the proposed technique have been obtained at the steady-state, which is reached after 1000 iterations of the Algorithm 1. For the cases of low and medium ratio of the known entries, the proposed technique outperforms both the MC-based and the LSM. For ratios larger than 0.6, the three techniques achieve almost the same reconstruction performance. Note that the LSM technique is based on the spatial coherence only, completely ignoring the temporal structure. Moreover, its performance is constrained by the reconstruction quality of the Laplacian matrix, hence even some small noise may cause very large errors. Note that, in our results, the Laplacian which is used by the

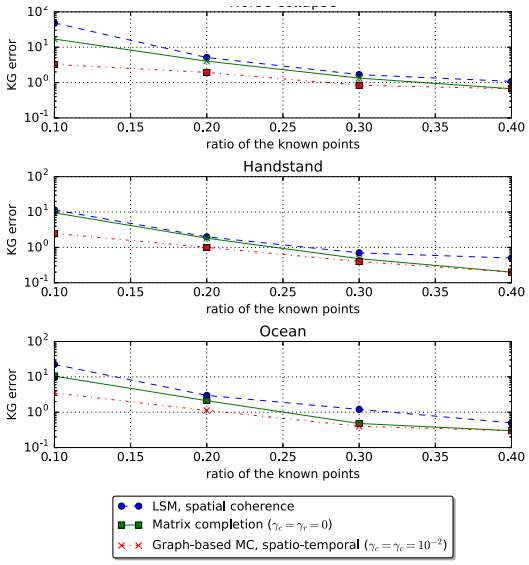

Fig. 2. KG error versus the ration of the known points for the case of the "Horse collapse" animation model. The steady state of the algorithms is taken after 2000 iterations.

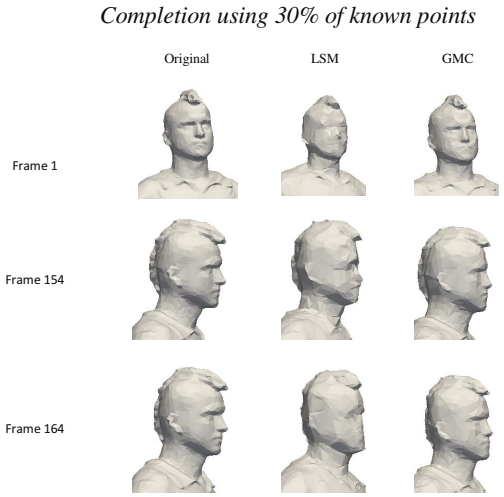

Fig. 3. Comparison of the techniques w.r.t. the detail perservation.

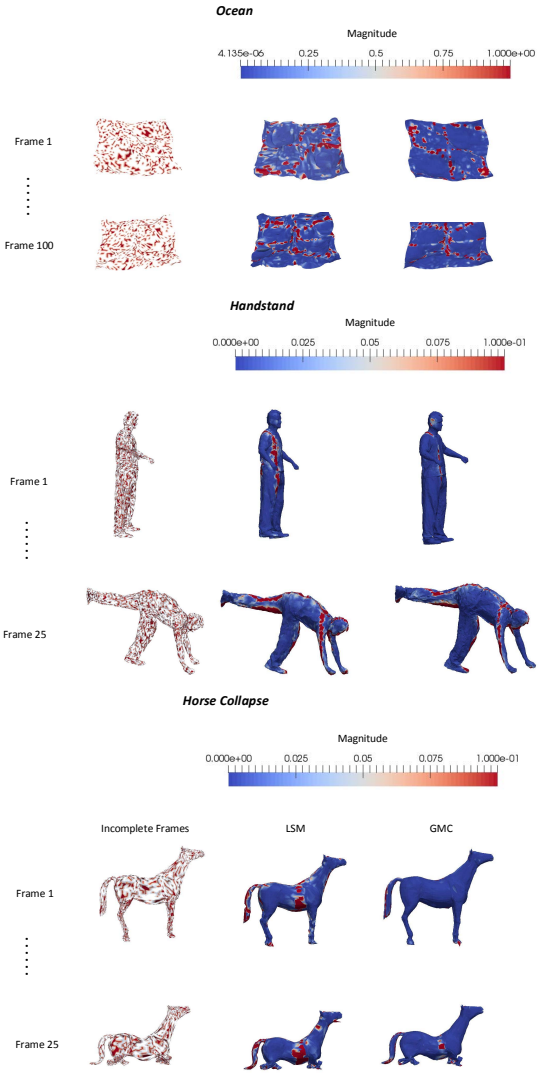

Fig. 4. Heatmap visualization of the normalized square error between the original model and the graph-based reconstruction for the case of 30% known entries.

LSM algorithm has been obtained by the MC algorithm.

In Fig. 5 we show the convergence curves of the MC techniques for two different cases of known entries (30% and 60%). Through experimental results, we have observed that for a larger number of known data ($> 40\%$), the matrix completion-based algorithms exhibit almost the same performance. This can also be justified based on the theoretic bounds of the MC techniques. Specifically, when the number of the known entries is larger than $K \geq N^{1.2} logN \approx 0.37 \cdot 3F \cdot N$, then MC techniques recover, with very high probability the exact matrix. Considering the convergence speed, we can observe that the graph-based techniques converge faster than the MC for all the cases. Moreover, for lower number of known entries ($< 30\%$) the graph-based MC exhibits lower error floor in NMSE.

5. CONCLUSION

In this work, we have provided a generic spatio-temporal coherent animation completion method based on matrix com-

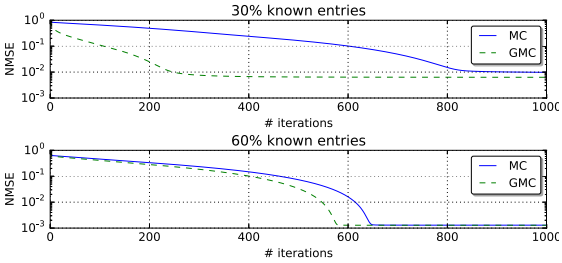

Fig. 5. Convergence curves of the matrix completion techniques for the case of the "Horse collapse" animation model.

pletion techniques, where the embedding of the mesh graphical structure has been formulated as a constrained convex optimization problem. The proposed algorithm achieves fast converge as compared to the conventional matrix completion technique. An extensive evaluation study using different 3D animation models verified that the proposed technique achieves plausible reconstruction output even with only 30% known geometry data, recovering the high frequency details of the original models.

Appendix

Proof of Proposition 1: Eq. (11) can be written as follows: $\mathbf{E}_\Omega \circ (\mathbf{Y}-\mathbf{M}) + \gamma_r \mathbf{L}_r^T \mathbf{L}_r \mathbf{Y} + \gamma_c \mathbf{Y} \mathbf{L}_c^T \mathbf{L}_c + \rho(\mathbf{Y}-\mathbf{H}) = \mathbf{0} \Rightarrow \mathbf{E}_\Omega \circ \mathbf{Y} + \gamma_r \mathbf{L}_r \mathbf{Y} + \gamma_c \mathbf{Y} \mathbf{L}_c + \rho \mathbf{Y} + \mathbf{C} = \mathbf{0}$ where $\mathbf{C} = -\rho \mathbf{H} - \mathbf{E}_\Omega \circ \mathbf{M}$. Based on the properties of the Hadamard product [17], the first term of (11) can be written as $\mathbf{E}_\Omega \circ \mathbf{Y} = \sum_{i=1}^{3FN} \mathbf{E}_{ii} \mathbf{Y} diag(\mathbf{E}_{\Omega,i})$ where $\mathbf{E}_i i$ is an all zero matrix except one entry in position (i,i) which is one, and $\mathbf{E}_{\Omega,i}$ is the i-th column of the $\mathbf{E}_\Omega$ matrix. Therfore, (11) can be written as $(A1): \sum_{i=1}^{3F+3} \mathbf{A}_i \mathbf{Y} \mathbf{B}_i - \mathbf{C} = \mathbf{0}$, where for $i = 1, \ldots, 3F$ we have $\mathbf{A}_i = \mathbf{E}_{ii}$ and $\mathbf{B}_i = diag(\mathbf{E}_{\Omega,i})$, for $i = 3F+1$ we have $\mathbf{A}_i = \gamma_r \mathbf{L}_r$ and $\mathbf{B}_i = \mathbf{I}_N$, for $i = 3F+2$ we have $\mathbf{A}_i = \mathbf{I}_{3F}$ and $\mathbf{B}_i = \gamma_c \mathbf{L}_c$, and for $i = 3F+3$ we have $\mathbf{A}_i = \rho \mathbf{I}$ and $\mathbf{B}_i = \mathbf{I}$. It is known that, the nessecary condition for the uniqueness of the solution is the non-singularity of the matrix $\sum_i \mathbf{B}_i^T \otimes \mathbf{A}_i$. Since $(A1)$ contains the term $\rho \mathbf{I}_{3FN}$, this condition is satisfied for $\rho > 0$.

6. REFERENCES

[1] M. Berger et al., "State of the Art in Surface Reconstruction from Point Clouds," in *Eurographics - State of the Art Reports*, 2014.

[2] P. Liepa, "Filling holes in meshes," in *ACM SIGGRAPH Symposium on Geometry Processing*, 2003, SGP '03, pp. 200–205.

[3] L. Zhang et al., "Spacetime faces: High resolution capture for modeling and animation," *ACM Trans. Graph.*, vol. 23, no. 3, pp. 548–558, Aug. 2004.

[4] Aris S. Lalos, A. A. Vasilakis, A. Dimas, and K. Moustakas, "Adaptive compression of animated meshes by exploiting orthogonal iterations," in *Computer Graphics International, Yokohama, Japan*, Jun. 2017.

[5] E. J. Candès and B. Recht, "Exact matrix completion via convex optimization," *Foundations of Computational Mathematics*, vol. 9, no. 6, pp. 717, 2009.

[6] Y. Deng, Q. Dai, and Z. Zhang, "Graph laplace for occluded face completion and recognition," *IEEE Transactions on Image Processing*, vol. 20, no. 8, pp. 2329–2338, Aug. 2011.

[7] Y. Peng et al., "Rasl: Robust alignment by sparse and low-rank decomposition for linearly correlated images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 11, pp. 2233–2246, Nov. 2012.

[8] Y. Deng et al., "Noisy depth maps fusion for multiview stereo via matrix completion," *IEEE Journal of Selected Topics in Signal Processing*, vol. 6, no. 5, pp. 566–582, Sep. 2012.

[9] Jian-Feng Cai et al., "A singular value thresholding algorithm for matrix completion," *SIAM Journal on Optimization*, vol. 20, no. 4, pp. 1956–1982, 2010.

[10] V. Kalofolias et al., "Matrix completion on graphs," *CoRR*, vol. abs/1408.1717, 2014.

[11] S. Boyd et al., "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Found. Trends Mach. Learn.*, vol. 3, no. 1, pp. 1–122, Jan. 2011.

[12] Gene H. Golub and Charles F. Van Loan, *Matrix Computations (3rd Ed.)*, Johns Hopkins University Press, Baltimore, MD, USA, 1996.

[13] R. W. Sumner and J. Popović, "Deformation transfer for triangle meshes," *ACM Transactions on Graphics (TOG)*, vol. 23, no. 3, pp. 399–405, 2004.

[14] D. Vlasic et al., "Articulated mesh animation from multi-view silhouettes," *ACM Trans. Graph.*, vol. 27, no. 3, pp. 97:1–97:9, Aug. 2008.

[15] Z. Karni and C. Gotsman, "Compression of soft-body animation sequences," *Computers & Graphics*, vol. 28, pp. 25–34, 2004.

[16] O. Sorkine and D. Cohen-Or, "Least-squares meshes," in *Proceedings Shape Modeling Applications*, Jun. 2004, pp. 191–199.

[17] C. Johnson, *Matrix Theory and Applications*, vol. 40, Proceedings of Symposia in Applied Mathematics, American Mathematical Society, 1989.