This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2018.2839911, IEEE Transactions on Multimedia

Distributed consolidation of highly-incomplete dynamic point clouds based on rank minimization

Evangelos Vlachos, Aris S. Lalos, Aristotelis Spathis-Papadiotis and Konstantinos Moustakas
Electrical and Computer Engineering Department, University of Patras, GR-26500, Rion-Patras, Greece.
Emails: {vlaxose, aris.lalos, agspathis, moustakas}@ece.upatras.gr

***Abstract*—Recently, there has been increasing interest for easy and reliable generation of 3D animated models facilitating several real-time applications (like immersive tele-presence, motion capture and gaming). In most of these applications, the reconstruction of soft body animations is based on time-varying point clouds which are non-uniformly sampled and highly incomplete. To overcome these significantly challenging imperfections without any additional information, first we introduce a novel reconstruction technique based on rank minimization theory, which can result into a unique solution to the otherwise ill-posed problem. This technique is further extended to exploit the spatial coherence which usually characterizes the soft-body animations. Based on the developed tools, we propose a distributed consolidation technique where the reconstruction is performed by working simultaneously on several group of frames. To achieve this, we impose temporal coherence between successive frame clusters by constraining the rank minimization problem. We validate the proposed techniques via experimental evaluation under different configurations and animated models, where we show that the high-frequency details of the models can be adequately recovered from a highly incomplete geometry dataset.**

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

I. Introduction

NOWADAYS, a considerably increasing attention has been attracted to geometry reconstruction of real-world scenes. Several techniques have been developed for the acquisition, e.g. motion compensated structured light [1], active space-time stereo [2], passive multi-view stereo [3]. Although the resolution and accuracy of the new generation image sensors and scanning techniques are constantly improving, due to the imperfect scanning conditions the acquired point data are often corrupted by severe noise, outliers, high variations in point density, misalignment and missing data. All these imperfections pose dramatic challenges to the reconstruction of dynamic shapes, of which one of the most significant is to deal with the multiple reflections and the object occlusion which usually result into highly-incomplete data.

The output of such an acquisition process is a sequence of unstructured point clouds, which is one of the most primitive and fundamental manifold representation. The enhancement process of these low-quality point cloud data is usually called *consolidation* [4]. Consolidation techniques [5]–[7] work on the captured data and output a new point set which more faithfully represents the *underlying shape*. They are considered as an essential pre-processing step before other point cloud operations (e.g., surface reconstruction, normal estimation), since improvements of the point cloud quality will have a major impact into the quality of the subsequent operations. Specifically, decoupling these processes can effectively avoid premature and erroneous decisions.

In this work, we consider the problem of dynamic point reconstruction given only a small number of geometry data, which have been sampled non-uniformly from the entire shape. Usually, the process of the recovery of the missing data requires some additional information, e.g., normals or scanner information, template models. As far as we know, there are very few techniques in the area that deal with the problem of missing data under a generic framework [8]. This is a consequence of the fact that the point cloud reconstruction problem can be very *ill-posed* when prior information is not being provided, and thus, an infinite number of surfaces can pass through (or near) a given set of data points.

Recently, a new signal processing framework termed as *matrix completion* (MC) [9] has been extensively used with great success in several applications. MC techniques aim to recover the missing entries of a matrix by minimizing its *rank* via the solution of an optimization problem [10]. The rank minimization problem [11] corresponds to the task of finding the simplest model that fit to the given data. One of the strong aspects of MC theory is the provided *universal* 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. MC has been successfully applied to several computer graphics and vision problems, such as the recovery of occluded faces [12], the face image alignment [13], and the fusion of point clouds from multiview images of the same object [14].

In computer graphics, the modeling of the geometry of dynamic scenes is obtained via *animation meshes*. When the sequences have fixed connectivity but the vertices' positions vary over time, then the sequences of meshes exhibit temporal coherence, apart from the spatial [15]. This *spatio-temporal* coherence of the meshes indicate that the underlying data reside on or near a low-dimensional subspace in a higher-dimensional space. In matrix form representation, the animation matrix can be constructed by stacking together the 3D frames of the static point clouds as row vectors. The resulting matrix can be accurately represented by keeping only a subset of the principal components, since it has a low-rank property. However, when the scanning operation results into missing point positions, outliers and noise, several entries of this matrix can be zero or erroneous, thus the computed

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2018.2839911, IEEE Transactions on Multimedia

principal components may contain severe errors. In this case, matrix completion is a viable reconstruction solution for robust recovery of the missing geometry data.

A. Related work

In this subsection we provide a brief overview of related work in the recovery of the geometry for static and time-varying meshes.

Surface reconstruction: Deriving a new point set from a given point cloud can also be considered in the context of defining point set surfaces. Conventionally, the area of surface reconstruction has been categorized in combinatorial methods, e.g, Delaunay triangulation [16] and implicit methods [17]. The Delaunay methods reconstruct a triangulated surface providing guarantees in the geometric and sometimes topological quality of the reconstruction. However, if the input points are non-uniformly distributed or incomplete, they do not work well. On the other hand, implicit techniques construct indicator functions for the underlying surface and perform isosurfacing to generate a mesh. They are more suitable for data that are noisy, incomplete or non-uniformly distributed. Recently, a unified framework has been proposed which jointly optimizes geometry and connectivity for surface reconstruction [18]. However, both the aforementioned approaches either perform well provided that the sampling of the input surface is sufficiently dense, or may require the estimation of the normals, the construction of the level set function and isosurfacing.

Static point cloud reconstruction: A huge number of prior works has investigated the problem of completion in static geometries, resulting in excellent filled static meshes. In [19] the least-squares meshes (LSM) algorithm has been proposed where the geometry of the mesh is reconstructed by solving a sparse linear system given only a small number of control points. However, the direct application of LSM to every frame separately is usually causing incorrect topologies and temporally incoherent surfaces. Another common approach to produce a temporally consistent dynamic mesh is to use a template prior [20]. 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.

Animation mesh reconstruction: A large number of existing approaches for animation reconstruction are based on a priori knowledge, which is represented by a template model. This template model is either provided by the user [21] or is being reconstructed by the data [22]. Specifically, in [22] a sequence of point clouds sampled at different time instances are used as input. The proposed approach automatically assembles them to a common shape that best fits all of the input data frames, without requiring prior knowledge of the template model. In [23], a method for the automatic reconstruction of spatial and temporal coherent animated meshes from raw real-time high resolution scanner data.

PCA reconstruction: Completion of missing data based on principal components analysis (PCA) approaches has been extensively used in several applications, e.g., for reconstructing an image of a face from a few pixels. However, there are very few published works concerning the case of 3D surface reconstruction, [24], while the majority of the works related with PCA of the point cloud are tackling mesh compression [25], [26].

B. Overview and contributions

In this work, we address the highly challenging problem completing the vertex positions missing from a time-varying point cloud describing an arbitrary shape by developing a consolidation framework based on rank minimization theory. Specifically, first we introduce a novel, geometry-myopic reconstruction technique based on matrix completion theory, which is able to provide a unique solution to the otherwise ill-posed problem. This technique is extended in order to exploit the spatial coherence of the geometry data by implicitly forcing row and column proximities. Building on this framework, a novel distributed technique for the reconstruction of massive data has been proposed, where the computational burden can be distributed over a network of interconnected machines via a light communication protocol.

It is important to note that the proposed framework does not depend on previous works in animation reconstruction and consolidation, thus develops novel geometry processing techniques using modern optimization tools. It can be considered as a part of a surface reconstruction pipeline, which would otherwise fail to recover the animation mesh due to the missing data and the lack of any prior information. To the best of our knowledge, matrix completion algorithms have never been applied to the animation reconstruction problem, despite their wide success on a large range of computer graphics and vision applications. Our motivation can be justified by the low-rank property of the animation matrix, which permits the employment of matrix completion techniques.

In summary, the main contributions of this work are the following:

- We describe a *generic* reconstruction technique based on rank minimization for the recovery of the missing data of dynamic point clouds. This technique does not use any prior information concerning the connectivity of the meshes.
- We extend the proposed technique exploiting the *spatial coherence* which usually characterizes the soft-body animations, which is captured via a Laplacian matrix. We prove that this extended framework is a generalization of the well-known least-square meshes (LSM [19]) for the reconstruction of dynamic point clouds.
- A practical scenario where the Laplacian matrix is unknown has been also considered. In order to recover the connectivity of the soft-rigid models in the case of highly-incomplete geometry data, we propose a robust approach based on the *average point cloud.*
- We introduce a novel *distributed* consolidation technique where the reconstruction is divided and performed in

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2018.2839911, IEEE Transactions on Multimedia

TABLE I
SUMMARY OF NOTATION.

$a, \mathbf{a}$ and $\mathbf{A}$	Scalar, vector and matrix variables
$\mathbf{A}^T$ and $\mathbf{A}^H$	Matrix transpose and Hermitian transpose
$[\mathbf{A}]_{i,j}$	Matrix element at the i-th row and j-th column
$\mathbf{M}$ and $\hat{\mathbf{M}}$	Real and reconstructed animation matrix
$\mathbf{L}$	Laplacian matrix
$\mathbf{I}_N$	the $N \times N$ identity matrix
$\mathbf{I}_{N\times K}$	the $N \times K$ identity matrix
$\mathbf{0}_{N\times K}$	the $N \times K$ matrix with zeros
Ω	Set containing matrix positions of observed vertices
$\mathbf{E}_\Omega$	Matrix containing 1 at positions $\in \Omega$, 0 elsewhere
$\mathbf{E}$	An all ones matrix
$\|\cdot\|_*$, $\|\cdot\|_F$	Nuclear and Frobenius norms of matrix
$\circ$	Element-wise (Hadamard) matrix product
$\otimes$	Kronecker product
$\langle \mathbf{A}, \mathbf{B} \rangle$	Inner product of two matrices equal to $\mathrm{trace}(\mathbf{A}^H\mathbf{B})$
$\mathrm{vec}(\mathbf{A})$	Vectorization of matrix $\mathbf{A}$
$\mathrm{unvec}(\mathbf{A})$	The inverse operation of $\mathrm{vec}(\cdot)$
$\mathrm{diag}(\mathbf{a})$	Diagonal matrix with vector $\mathbf{a}$ on the diagonal
SVT_t	Singular Value Thresholding with threshold t
$\mathcal{L}(\mathbf{X}, \mathbf{Y}, \mathbf{Z})$	Lagrangian with primal ($\mathbf{X}, \mathbf{Y}$) and dual ($\mathbf{Z}$) variables
$\mathbf{V}_{c,\ell}$	Auxiliary variable for pairing neighbouring clusters c, ℓ

successive parts of the dynamically generated animation point cloud. To achieve this, the rank minimization problem is further extended by introducing constraints which impose temporal coherence between successive group of frames.

II. PRELIMINARIES:

A. Point-cloud reconstruction based on principal components analysis

Subspace approximation techniques such as PCA have been exploited with great success for the compression and reconstruction of dynamic point clouds [25]–[27]. However, the consolidation of the animation matrix, due to the low-quality of the scanned data does not permit its utilization. In particular, it is known that singular value decomposition (SVD) is not robust to outlier noise and incomplete data. To support this, let us first formulate the PCA-based point cloud reconstruction technique. A summary of the notation used throughout the text can be found in Table I.

We consider that for each frame $f \in \{1, \dots, F\}$, the xyz-coordinates of N points are concatenated to a matrix,

$$\mathbf{M}_f = \begin{bmatrix} m_{f,x}(1) \dots m_{f,x}(N) \\ m_{f,y}(1) \dots m_{f,y}(N) \\ m_{f,z}(1) \dots m_{f,z}(N) \end{bmatrix} \in \mathbb{R}^{3\times N}. \quad (1)$$

Stacking all the F frames, we construct the animation matrix $\mathbf{M} = \begin{bmatrix}\mathbf{M}_1^T \dots \mathbf{M}_F^T\end{bmatrix}^T \in \mathbb{R}^{3F\times N}$. Let the SVD of this matrix be written as

$$\mathbf{M} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^H \quad (2)$$

where $\mathbf{U} \in \mathbb{C}^{3F\times 3F}$ and $\mathbf{V} \in \mathbb{C}^{N\times N}$ are unitary matrices, while $\mathbf{\Sigma} \in \mathbb{R}^{3F\times N}$ diagonal matrix with the singular values $\sigma_i, i = 1, \dots, \min(3F, N)$, on the diagonal.

Due to the underlying spatio-temporal coherence which typically exists in geometric animations [15], we can expect that $\mathbf{M}$ can be adequately approximated by using fewer components, i.e. by keeping only $K < \min(3F, N)$ principal components of $\mathbf{M}$, thus the reconstructed animation matrix $\hat{\mathbf{M}}$ is defined as:

$$\mathbf{M} \approx \hat{\mathbf{M}} = \mathbf{U}\hat{\mathbf{\Sigma}}_K\mathbf{V}^H \quad (3)$$

where $\hat{\mathbf{\Sigma}}_K$ is the diagonal matrix with $\hat{\sigma}_i = \sigma_i$, for $i = 1, \dots, K$ and $\hat{\sigma}_i = 0$ for $i = K+1, \dots, \min(3F, N)$. For instance, Fig. 1 shows the reconstruction result for the 39-th frame of the Handstand animation model [28], where only the 10% of the principal components have been used, i.e., $K = 50$ with $\mathbf{M} \in \mathbb{R}^{525\times 10002}$.

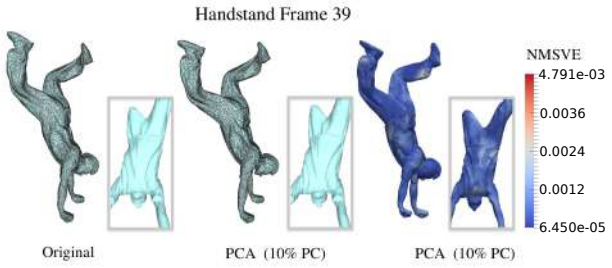

Fig. 1. Left: The original mesh of the 39-th frame of the Handstand animation model $\mathbf{M}$. Middle: The PCA-based reconstruction $\hat{\mathbf{M}}$ using 10% of the principal components. Right: Heatmap visualization of the normalized mean square visual error (NMSVE) with maximum error equal to $4.79\mathrm{e}{-3}$.

In general, an animation matrix $\mathbf{M} \in \mathbb{R}^{3F\times N}$ described by $3FN$ values, it has only $(2N-K)K$ degrees of freedom. This fact can be revealed by counting parameters in the SVD (the number of degrees of freedom associated with the description of the singular values and of the left and right singular vectors). This behaviour is standard for many models of soft-body animation, thus, by imposing small rank to $\hat{\mathbf{M}}$ we usually sacrifice a small level of the reconstruction quality.

B. Matrix completion

Matrix completion (MC) is a powerful tool that permits the recovery of a matrix given only a subset of its entries, and it is based on the low-rank property of the given matrix. Formally, can be expressed as

$$\min_{\mathbf{X}} \mathrm{rank}(\mathbf{X}) \text{ subject to } \mathbf{E}_\Omega \circ (\mathbf{X} - \hat{\mathbf{M}}) = \mathbf{0} \quad (4)$$

where $\hat{\mathbf{M}} \in \mathbb{R}^{3F\times N}$ is the known matrix, Ω is the set with the matrix indices of the non-zero entries, $\mathbf{X}$ is the unknown matrix and $\mathrm{rank}(\mathbf{X})$ provides the rank of the matrix $\mathbf{X}$. The matrix $\mathbf{E}_\Omega$ is composed by ones and zeros, i.e.,

$$[\mathbf{E}_\Omega]_{i,j} = \begin{cases} 1 & \text{if } (i,j) \in \Omega \\ 0 & \text{otherwise} \end{cases} \quad (5)$$

In [29], it was proposed that the matrix completion problem (4) results into the following equivalent unconstrained problem,

$$\min_{\mathbf{X}} \tau\|\mathbf{X}\|_* + \frac{1}{2}\|\mathbf{X} - \mathbf{Y}\|_F^2 \quad (6)$$

where $\tau \geq 0$ is a weighting parameter and $\mathbf{Y} = \mathbf{E}_\Omega \circ (\mathbf{X} - \hat{\mathbf{M}})$. It has been proved that the solution of (6) is provided by the singular-value-thresholding (SVT) operator $\mathrm{SVT}_\tau(\mathbf{Y})$ [29, Theorem 2.1]. Specifically, let $\mathbf{Y} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^H$ be the SVD of

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2018.2839911, IEEE Transactions on Multimedia

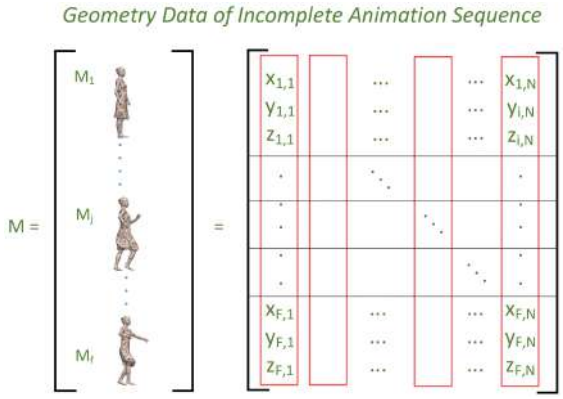

Fig. 2. An example of a highly-incomplete time-varying point cloud matrix.

a matrix $\mathbf{Y}$, where $\mathbf{U}$ and $\mathbf{V}$ are matrices with orthonormal columns. Then, the SVT operator is defined as

$$\mathcal{D}_\tau(\mathbf{Y}) = \mathbf{U}\text{diag}\big(\{(\sigma_i - \tau)_+\}_{1\leq i\leq r}\big)\mathbf{V}^H \tag{7}$$

e.g., SVT applies the τ-thresholding rule to the singular values of the operand matrix shrinking those below τ towards zero.

III. Consolidation based on Rank Minimization

In this Section, we introduce an efficient solution to the problem of point cloud reconstruction based on matrix completion techniques. Afterwards, we improve the reconstruction quality by using geometry constraints provided by the Laplacian matrix.

A. Exploitation of the underlying subspace properties

Our aim is to reconstruct the matrix $\hat{\mathbf{M}}$ given a reduced number of 3D points and assuming that there is no other information available, e.g., edge lengths, faces. Essentially, the incomplete animation matrix has zeros at several positions, as shown in Fig. 2 and can be formally expressed as the element-wise (Hadamard) product $\mathbf{E}_\Omega \circ \hat{\mathbf{M}}$ of the animation matrix and the matrix $\mathbf{E}_\Omega$. Ω is the set with the matrix $\hat{\mathbf{M}}$ positions (i, j) which are known and correspond to the known 3D points of the point cloud, with $M \triangleq |\Omega| \ll 3F \cdot N$ known entries. Since the matrix $\hat{\mathbf{M}}$ has rank equal to $K \ll N$, a rank minimization problem [11], [29] can be formulated in order to retrieve the missing data, i.e.,

$$\min_{\mathbf{X}} \|\mathbf{X}\|_* \text{ subject to } \mathbf{E}_\Omega \circ (\mathbf{X} - \hat{\mathbf{M}}) = \mathbf{0} \tag{8}$$

where $\|\cdot\|_*$ denotes the nuclear norm defined as $\|\mathbf{X}\|_* = \sum_{i=1}^{\min(3F,N)} \sigma_{X,i}^2$ with $\sigma_{X,i}$ denotes the i-th singular value of $\mathbf{X}$. Eq. (8) can be approximated by the following unconstrained optimization problem:

$$\min_{\mathbf{X}} \frac{1}{2}\|\mathbf{E}_\Omega \circ (\mathbf{X} - \hat{\mathbf{M}})\|_F^2 + \tau\|\mathbf{X}\|_* \tag{9}$$

where the first term of (9) minimizes the error between the known points and the recovered with $\|\cdot\|_F$ 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. Note that the choice for τ is determined specifically for each animation matrix, thus it is further investigated in Section V.

To proceed, let us introduce an auxiliary matrix $\mathbf{Y} \in \mathbb{R}^{3F\times N}$ and formulate the optimization problem (9) as follows:

$$\min_{\mathbf{X}} \frac{1}{2}\|\mathbf{E}_\Omega \circ (\mathbf{Y} - \mathbf{M})\|_F^2 + \tau\|\mathbf{X}\|_* \text{ s.t. } \mathbf{X} = \mathbf{Y}. \tag{10}$$

Although expression (10) seems to be more complex than (9) due to the introduced constraint, essentially it permits the employment of efficient algorithms, such as the Alternating Method of Multipliers (ADMM) [10], due to the decomposed cost function. In this work, the proposed techniques are based on the ADMM, since it is a simple but powerful technique, based on a solid theory with performance guarantees for its convergence.

In order to solve (10), first let us express its augmented Lagrangian form: $\mathcal{L}(\mathbf{X}, \mathbf{Y}, \mathbf{Z}) = \frac{1}{2}\|\mathbf{E}_\Omega \circ (\mathbf{Y} - \mathbf{M})\|_F^2 + \tau\|\mathbf{X}\|_* + \frac{\rho}{2}\|\mathbf{X}-\mathbf{Y}\|_F^2 + \langle \mathbf{Z}, \mathbf{X}-\mathbf{Y}\rangle$, where $\mathbf{X}, \mathbf{Y}$ are the primal variables while $\mathbf{Z}$ is the dual variable, and $\rho > 0$ is the penalty parameter. Then, ADMM is composed by the following steps which are executed for each iteration $i = 1, \ldots, i_{\max}$:

$$\mathbf{X}(i+1) = arg\min_{\mathbf{X}} \mathcal{L}(\mathbf{X}, \mathbf{Y}(i), \mathbf{Z}(i)) \tag{11}$$

$$\mathbf{Y}(i+1) = arg\min_{\mathbf{Y}} \mathcal{L}(\mathbf{X}(i+1), \mathbf{Y}, \mathbf{Z}(i)) \tag{12}$$

$$\mathbf{Z}(i+1) = \mathbf{Z}(i) + \rho(\mathbf{X}(i+1) - \mathbf{Y}(i+1)). \tag{13}$$

The minimization of (11) is equivalent with $\min_{\mathbf{X}} \tau\|\mathbf{X}\|_* + \frac{\rho}{2}\|\mathbf{X} - \mathbf{Y} + \rho^{-1}\mathbf{Z}\|_F^2$, which is known that it can be solved via the SVT operator, i.e.,

$$\mathbf{X}(i+1) = \text{SVT}_{\tau/\rho}\big(\mathbf{Y}(i) - \rho^{-1}\mathbf{Z}(i)\big). \tag{14}$$

In particular, the SVT is a non-linear function which applies a soft-thresholding rule at level $\frac{\tau}{\rho}$ to the singular values of the input matrix $\mathbf{Y}(i) - \rho^{-1}\mathbf{Z}(i)$.

Now, to obtain the optimality condition of (12) we set the partial derivative of the augmented Lagrangian with respect to $\mathbf{Y}$ equal to zero, i.e.,

$$\begin{aligned} &\frac{\partial}{\mathbf{Y}}\mathcal{L}\big(\mathbf{X}(i+1), \mathbf{Y}, \mathbf{Z}(i))\big) = \mathbf{0} \\ \Rightarrow\ &\mathbf{E}_\Omega \circ (\mathbf{Y} - \hat{\mathbf{M}}) - \rho\big(\mathbf{X}(i+1) - \mathbf{Y}\big) - \mathbf{Z}(i) = \mathbf{0} \end{aligned} \tag{15}$$

which has solution expressed as follows (c.f. Appendix A):

$$\mathbf{Y}(i+1) = \text{unvec}\Big(\mathbf{A}^{-1}\big(\mathbf{E}_\Omega \circ \hat{\mathbf{M}} + \rho\mathbf{X}(i+1) + \mathbf{Z}(\ell)\big)\Big) \tag{16}$$

where $\mathbf{A} = \text{diag}(\text{vec}(\mathbf{E}_\Omega)) + \rho\mathbf{I}_{3F\cdot N}$. The ADMM-based matrix completion technique for the reconstruction of the partially known animation matrix $\mathbf{E}_\Omega \circ \hat{\mathbf{M}}$ is summarized in Algorithm 1.

Convergence speed: It is known that when high quality reconstruction is required, the sequential implementation ADMM may suffer from low convergence speed. To optimize the speed of the convergence, we utilize the *over-relaxation* technique [10], where the quantity $\rho\mathbf{X}$ in (15) is replaced by the following one: $\rho(\alpha\mathbf{X} - (1-\alpha)\mathbf{Y})$, where α is the relaxation parameter.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2018.2839911, IEEE Transactions on Multimedia

Algorithm 1 CoPC: Proposed consolidation algorithm for the completion of the point cloud matrix

Input: $\Omega, \mathbf{E}_\Omega \circ \mathbf{M}, \rho, i_{\max}$
Output: $\mathbf{X}(i_{\max})$
Computation of $\mathbf{A}$ via $\mathbf{A} = \text{diag}(\text{vec}(\mathbf{E}_\Omega)) + \rho \mathbf{I}_{3F \cdot N}$
for $i = 1, \ldots, i_{\max}$ **do**
  {*Step 1: Soft-thresholding of the singular values*}
  $\mathbf{X}(i+1) \leftarrow \text{SVT}_{\tau/\rho}\big(\mathbf{Y}(i) - \rho^{-1}\mathbf{Z}(i)\big)$
  {*Step 2: Computation of* (15) *with the over-relaxation*}
  $\mathcal{V} \leftarrow \alpha \mathbf{X}(i+1) - (1-\alpha)\mathbf{Y}(i)\big)$
  $\mathbf{b} \leftarrow \text{vec}(\mathbf{E}_\Omega \circ \mathbf{M} + \rho\mathcal{V} - \mathbf{Z}(i))$
  Solve the system $\mathbf{A}\mathbf{y} = \mathbf{b}$
  $\mathbf{Y}(i+1) \leftarrow \text{unvec}(\mathbf{y})$
  {*Step 3: Update the dual variable*}
  $\mathbf{Z}(i+1) \leftarrow \mathbf{Z}(i) + \rho(\mathbf{X}(i+1) - \mathbf{Y}(i+1))$
end for

Computational Complexity:: The costly step of the Algorithm 1 is the computation of the SVT operator to sparse matrix $\mathbf{Y}(i) - \rho^{-1}\mathbf{Z}(i)$. In general, the complexity of the SVD is $\mathcal{O}((3F)^2 N)$ per iteration for $N > 3F$ [30, Chapter 8.6, pp 486-493]. However, the number of known entries is usually much lower than the total number of entries of the matrix $\hat{\mathbf{M}}$, 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, thus the complexity can be reduced over to $\mathcal{O}(NFM)$.

B. Incorporation of the spatial coherence

Spatial coherence can provide additional information about the geometry of the 3D model, improving the reconstruction quality. Thus, apart of the underlying structure of the data, the proximity of the vertices can be incorporated into the problem formulation. In the previous subsection, we have developed Algorithm 1 as a generic approach for point cloud reconstruction. To incorporate the available geometry structure, here, we introduce a graph-based matrix completion technique for the reconstruction of time-varying point clouds, exploiting the spatial coherence of the shapes.

The proximity of the vertex positions of a static point cloud can be encoded by a graph, where an edge between two graph vertices (where each one represents a different point in the 3D space) denotes that these points are close in the 3D space. To be more specific, let us consider the graph $\mathbf{G} = (V, E, W)$ where $V = \{\mathbf{v}_1, \ldots, \mathbf{v}_N\}$ is the vertex set with each vertex represents a 3D point. $E \subseteq V \times V$ is the edge set, where the edge e_{ij} connects the two adjacent vertices $\mathbf{v}_i \sim \mathbf{v}_j$ depending on their proximity in the 3D space. $W = \{w_{ij}, i, j = 1, \ldots, N\}$ is the set with the non-negative weights. The binary Laplacian matrix $\mathbf{L}$ can be employed in order to provide a proximity metric for the point cloud geometry, with $\mathbf{L} = \mathbf{D} - \mathbf{A}$. The matrix $\mathbf{A} \in \mathbb{R}^{N \times N}$ represents the connectivity of the vertices of $\mathbf{M}_f$, with $\mathbf{A}_{(i,j)} = 1$ when $(i,j) \in E$ and 0 otherwise, while $\mathbf{D}$ is the diagonal matrix with $\mathbf{D}_{(i,i)} = |N(i)|$ and $N(i) = \{j \mid (i,j) \in E\}$ is a set with the immediate neighbours for node i.

It is important to note that the spatial coherence has been previously employed for the reconstruction of a static point cloud, in the well known technique of LSM [19]. Specifically, LSM is described as the solution of the following extended system of equations,

$$\mathbf{X}_{lsm} \begin{bmatrix} \delta\mathbf{L} & \mathbf{I}_{N\times K} \end{bmatrix} = \begin{bmatrix} \mathbf{0}_{3F\times N} & \mathbf{R} \end{bmatrix} \tag{17}$$

where $\mathbf{X}_{lsm} \in \mathbb{R}^{3F\times N}$, $\mathbf{L} \in \mathbb{R}^{N\times N}$, δ is a weighting parameter and $\mathbf{R} \in \mathbb{R}^{3F\times K}$ are the known $3FK$ *anchor* points from $\mathbf{M}$. Let $\mathbf{L}_e = \begin{bmatrix} \delta\mathbf{L} & \mathbf{I}_{N\times K} \end{bmatrix}$ and $\mathbf{R}_e = \begin{bmatrix} \mathbf{0}_{3F\times N} & \mathbf{R} \end{bmatrix}$, then the least-squares solution of (17) is expressed as: $\mathbf{X}_{lsm} = \mathbf{R}_e\mathbf{L}_e^T(\mathbf{L}_e\mathbf{L}_e^T)^{-1}$.

To proceed, let us first note that for the case of soft-body animations, while the relative vertex distance may vary from frame to frame, the adjacency of the vertices remains consistent, thus providing the spatial coherence property. Therefore, all the frames could share a common Laplacian matrix, which can be obtained based on one arbitrary frame $\mathbf{M}_f$. Therefore, the proximity metric can be embedded into problem (9) according to the following formulation:

$$\min_{\mathbf{X}} \frac{1}{2}\|\mathbf{E}_\Omega \circ (\mathbf{X} - \hat{\mathbf{M}})\|_F^2 + \tau\|\mathbf{X}\|_* + \frac{\gamma}{2}\|\mathbf{X}\mathbf{L}\|_F^2 \tag{18}$$

where via the last term of (18), we demand that the neighbouring node positions of the reconstructed frames to be close to each other, and γ represent the associated regularization parameter of the graph Laplacian. Before proceeding to the solution of (18) let us note that:

Proposition 1: Given that the connectivity of the animation matrix $\mathbf{M}$ is described via the binary Laplacian matrix $\mathbf{L}$, then the LSM [19] technique (17) can be viewed as a special case of the optimization problem (18) for $\tau = 0$ and $\gamma = \delta$.

Proof 1: The proof is given at the Appendix C.

Since the solution of (18) exploits the spatial coherence as well as the underlying coherence of the geometry data, it is expected to improve the performance over the LSM.

Now, let us express the augmented Lagrangian of the equivalent splitting version of the optimization problem (18) as $\mathcal{L}(\mathbf{X}, \mathbf{Y}, \mathbf{Z}) = \frac{1}{2}\|\mathbf{E}_\Omega \circ (\mathbf{Y} - \hat{\mathbf{M}})\|_F^2 + \tau\|\mathbf{X}\|_* + \frac{\gamma}{2}\|\mathbf{Y}\mathbf{L}\|_F^2 + \frac{\rho}{2}\|\mathbf{X} - \mathbf{Y}\|_F^2 + \langle \mathbf{Z}, \mathbf{X} - \mathbf{Y}\rangle$, where, as previously, $\mathbf{X}, \mathbf{Y}$ are the primal variables, $\mathbf{Z}$ the dual variable and ρ is the penalizing factor. Based on the analysis of the previous subsection, it can be seen that only the second step of ADMM is essentially different in this case. The optimality condition with respect to the $\mathbf{Y}$ variable is expressed as:

$$\mathbf{E}_\Omega \circ (\mathbf{Y} - \hat{\mathbf{M}}) + \gamma\mathbf{Y}(\mathbf{L}^T\mathbf{L}) + \rho\big(\mathbf{Y} - \mathbf{X}(i+1)\big) - \mathbf{Z}(i) = \mathbf{0} \tag{19}$$

The minimizer of (19) is obtained by solving the following generalized Sylvester equation [31] (c.f. Appendix B):

$$\sum_{i=1}^{3F+2} \mathbf{A}_i\mathbf{Y}\mathbf{B}_i - \mathbf{C} = \mathbf{0} \tag{20}$$

with

$$\mathbf{A}_i = \begin{cases} \mathbf{E}_{ii} & i = 1, \ldots, 3F \\ \gamma\mathbf{L}^T\mathbf{L} & i = 3F+1 \\ \rho\mathbf{I}_N & i = 3F+2 \end{cases} \tag{21}$$

$$\mathbf{B}_i = \begin{cases} \text{diag}(\mathbf{E}_{\Omega,i}) & i = 1, \ldots, 3F \\ \mathbf{I}_N & i = 3F+1, 3F+2 \end{cases} \tag{22}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2018.2839911, IEEE Transactions on Multimedia

and $\mathbf{C} = \mathbf{E}_\Omega \circ \hat{\mathbf{M}} + \rho\mathbf{X}(i+1) + \mathbf{Z}(i)$, which has a unique solution for $\rho > 0$. Eq. (20) is equivalent with the following $3FN \times 3FN$ linear system:

$$\left[\sum_i \mathbf{B}_i^T \otimes \mathbf{A}_i\right] \mathrm{vec}(\mathbf{Y}) = \mathrm{vec}(\mathbf{C}) \Rightarrow \mathbf{R}\mathbf{y} = \mathbf{r} \qquad (23)$$

where $\mathbf{R} \triangleq \sum_i \mathbf{B}_i^T \otimes \mathbf{A}_i$, $\mathbf{y} \triangleq \mathrm{vec}(\mathbf{Y})$ and $\mathbf{r} \triangleq \mathrm{vec}(\mathbf{C})$, thus the solution of (23) is straightforward.

The proposed algorithm in this case (termed as sCoPC) is described by the same steps with Algorithm 1, except for the Step 2 which is replaced by the procedure of the computation of (23).

Computational Complexity:: Unlike Algorithm 1, the major computational bottleneck in sCoPC lies at the update procedure of the ADMM variable $\mathbf{Y}$ in eq. (23), since the matrix that has to be inverted is no longer diagonal. The direct approach of matrix inversion for (23) requires complexity of the order $\mathcal{O}\big((3FN)^3\big)$, thus it cannot be applied for this case, due to the potentially huge dimensions of this system. Even with the employment of iterative algorithms (e.g. GMRES [32]) the complexity remains impractical, i.e., $\mathcal{O}\big((3FN)^2 i_{max}\big)$ where $i_{max} \ll 3FN$ is the number of algorithm iterations. Instead, techniques that exploit the multiple right-hand-sides of the equations are required to approximate efficiently the solution of (20) [33], [34].

IV. Distributed consolidation

In the previous section, the proposed techniques applied on the whole set of the animation frames. However, due to many practical limitations the operations on the points have to be performed in parts, i.e., by dividing the data and processing at each time a small portion. In our case, the dynamic point cloud data can be divided by grouping several frames into (*clusters*) and their processing can be obtained in parallel using interconnected devices. From matrix completion perspective, this problem is highly ill-conditioned since the number of the available entries are much lower than the required one. To overcome this problem, additional information has to be incorporated to the problem formulation. On this premise, we propose the exploitation of temporal coherence between the successive frames, thus, we introduce an spatio-temporal coherence exploitation technique.

To proceed, let us consider the case where $\mathbf{M}$ is divided into C disjoint clusters i.e., $\hat{\mathbf{M}} = \begin{bmatrix} \hat{\mathbf{M}}_1^T & \dots & \hat{\mathbf{M}}_C^T \end{bmatrix}^T$, where the c-th cluster has f_c frames, i.e., $\hat{\mathbf{M}}_c \in \mathbb{R}^{3f_c \times N}$ for $c = 1, \dots, C$ and $f_c < F$. In analogy with the problem (18), we express the cluster-based rank minimization as follows:

$$\min_{\{\mathbf{X}_c\}_{c=1}^C} \sum_{c=1}^{C} \tau\|\mathbf{X}_c\|_* + \frac{1}{2}\|\mathbf{E}_{\Omega_c} \circ \big(\mathbf{X}_c - \hat{\mathbf{M}}_c\big)\|_F^2 + \frac{\gamma}{2}\|\mathbf{X}_c\mathbf{L}_c\|_F^2 \qquad (24)$$

where the minimization is obtained jointly for all clusters $\mathbf{X}_c \in \mathbb{R}^{3f_c \times N}$. Note that the sum of the nuclear norms of the partitioned matrices $\hat{\mathbf{M}}_c$ can be larger than the nuclear norm of $\mathbf{M}$, i.e., $\|\mathbf{M}\|_* \leq \sum_{c=1}^C \|\hat{\mathbf{M}}_c\|_*$. This fact combined with that the number of available data decreases for the same under-sampling ratio, i.e., $|\Omega_c| \leq |\Omega|$, indicate the bad condition of the clustered case. To mitigate this problem, we propose to utilize the available information of the *temporal* coherence, incorporated as additional constraints into the rank minimization problem (24). Recall that in the previous subsection, the structure of the average point cloud has been encoded into the rank minimization problem, however, the temporal coherence of the animation sequence has not been explicitly utilized.

To achieve this, first let us introduce a suitable metric to evaluate the proximity of the vertex positions between the successive frames. In particular, assume that ϵ denotes the coherence between the successive frames $\mathbf{M}_f$ and $\mathbf{M}_{f+1}$, then the expression

$$\|\mathbf{E}_{\Omega_{f,f+1}} \circ \mathbf{M}_f - \mathbf{E}_{\Omega_{f+1,f}} \circ \mathbf{M}_{f+1}\|_F^2 \leq \epsilon \qquad (25)$$

can be employed to impose temporal coherence into the problem formulation (18). Note that $\mathbf{E}_{\Omega_{f,f+1}} \circ \mathbf{M}_f$ selects the vertices of $\mathbf{M}_f$ frame which are adjacent with $\mathbf{M}_{f+1}$, while $\mathbf{E}_{\Omega_{f+1,f}} \circ \mathbf{M}_{f+1}$ selects the vertices of $\mathbf{M}_{f+1}$ frame which are adjacent with $\mathbf{M}_f$. Note that $\Omega_{f,f+1}$ and $\Omega_{f+1,f}$ are the sets which indicate the neighbouring vertices and $\mathbf{E}_{\Omega_{f,f+1}}, \mathbf{E}_{\Omega_{f+1,f}} \in \mathbb{R}^{3f_c \times N}$

Then, we can formulate the following temporally-coherent graph-based rank minimization problem, where the new introduced constraints force the neighbouring vertices to consent to the close values, i.e.,

$$\min_{\{\mathbf{X}_c\}} \sum_{c=1}^{C} \tau\|\mathbf{X}_c\|_* + \frac{1}{2}\|\mathbf{E}_{\Omega_c} \circ \big(\mathbf{X}_c - \mathbf{M}_c\big)\|_F^2 + \frac{\gamma}{2}\|\mathbf{X}_c\mathbf{L}_c\|_F^2 + \frac{\xi}{2}\sum_{\ell \in \mathcal{N}_c} \|\mathbf{E}_{\Omega_{c,\ell}} \circ \mathbf{X}_c - \mathbf{E}_{\Omega_{\ell,c}} \circ \mathbf{X}_\ell\|_F^2 \qquad (26)$$

where $\mathcal{N}_c$ denotes the neighbouring areas of the c-th cluster. Since the clusters are obtained successively, it is that $\mathcal{N}_c = \{c-1, c+1\}$, where $\mathbf{M}_{c-1}$ and $\mathbf{M}_{c+1}$ denote the previous and the next cluster respectively. However, the fourth term of this optimization problem couples the problem among the clusters. To overcome this issue, let us introduce an auxiliary variable per pair of neighbouring clusters c, ℓ, denoted by $\mathbf{V}_{c,\ell}$ with the property that $\mathbf{V}_{c,\ell} = \mathbf{V}_{\ell,c}$. Then, the splitting version of can be expressed as:

$$\begin{aligned} \min_{\{\mathbf{X}_c\},\{\mathbf{Y}_c\},\{\mathbf{V}_{c,\ell}\}} & \sum_{c=1}^{C} \tau\|\mathbf{X}_c\|_* + \frac{1}{2}\|\mathbf{E}_{\Omega_c} \circ \big(\mathbf{Y}_c - \mathbf{M}_c\big)\|_F^2 \\ & + \frac{\gamma}{2}\|\mathbf{Y}_c\mathbf{L}_c\|_F^2 + \frac{\xi}{2}\sum_{\ell \in \mathcal{N}_c} \|\mathbf{E}_{\Omega_{c,\ell}} \circ \mathbf{Y}_c - \mathbf{V}_{c,\ell}\|_F^2 \\ & \text{s.t. } \mathbf{Y}_c = \mathbf{X}_c \end{aligned} \qquad (27)$$

where the cost function has been separated based on the three variables, $\mathbf{X}_c, \mathbf{Y}_c$ and $\mathbf{V}_{c,\ell}$, thus it can be efficiently solved by the ADMM algorithm (c.f. Appendix D). The obtained solution of (27) is described in Algorithm 2 (dCoPC).

Computational Complexity: The proposed algorithm overcomes the high complexity issues of Algorithm 2, since it solves multiple smaller instances of each subproblem, one per input cluster (cluster size being a user-selected parameter), as shown in Fig. 3. Note that the parallelism concerns the clusters, i.e., the inner "for loops" in Algorithm 2.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2018.2839911, IEEE Transactions on Multimedia

Algorithm 2 dCoPC: Proposed distributed consolidation algorithm

Input: $\Omega_c \forall c, \Omega_{c,\ell} \forall \ell \in \mathcal{N}_c, \mathbf{E}_\Omega \circ \mathbf{M}, \gamma, \xi, i_{\max}$
Output: $\mathbf{X}(i_{\max})$

for $i = 1, \ldots, i_{\max}$ **do**
  {*Step 1: Soft-thresholding of the singular values*}
  for $c = 1, \ldots, C$ **do**
    $\mathbf{X}_c(i+1) \leftarrow \mathrm{SVT}_{\tau/\rho}\big(\mathbf{Y}_c(i) - \rho^{-1}\mathbf{Z}_c(i)\big)$
  end for
  {*Step 2: Computation of* $\mathbf{Y}$}
  for $c = 1, \ldots, C$ **do**
    $\mathbf{A}_c \leftarrow \gamma\big((\mathbf{L}_c^T\mathbf{L}_c) \otimes \mathbf{I}_{f_c}\big) + \mathrm{diag}\big(\mathrm{vec}(\xi\mathbf{E}_{\Omega_{c,\ell}} + \mathbf{E}_{\Omega_c} + \rho\mathbf{E})\big)$
    $\mathbf{b}_c \leftarrow \mathrm{vec}\big(\xi\mathbf{V}_{c,\ell} + \mathbf{Z}_c + \rho\mathbf{X}_c + \mathbf{E}_{\Omega_c} \circ \mathbf{M}_c\big)$
    $\mathbf{Y}_c(i+1) \leftarrow \mathbf{A}_c^{-1}\mathbf{b}_c$
  end for
  {*Step 3: Computation of the average value of the neighboring 3D points*}
  for $c = 1, \ldots, C$ **do**

$$\mathbf{V}_{c,\ell}(i+1) \leftarrow \frac{\mathbf{E}_{\Omega_{c\ell}} \circ \mathbf{Y}_c(i+1) + \mathbf{E}_{\Omega_{\ell c}} \circ \mathbf{Y}_\ell(i+1)}{2}, \quad \forall \ell \in \{\Omega_{c,c-1}, \Omega_{c+1,c}\}$$

  end for
  {*State 4: Update the dual variable*}
  for $c = 1, \ldots, C$ **do**
    $\mathbf{Z}_c(i+1) \leftarrow \mathbf{Z}_c(i) + \rho\big(\mathbf{X}_c(i+1) - \mathbf{Y}_c(i+1)\big)$
  end for
end for

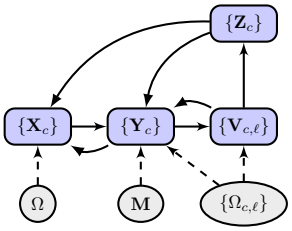

Fig. 3. Flow chart of the distributed consolidation algorithm. The notation $\{\cdot\}$ indicates that quantities for each frame cluster are computed in a parallel manner for $c = 1, \ldots, K$.

Moreover, we can take advantage of the natural separability of the problem to describe a distributed protocol with low communication overhead between the clusters, as it is depicted in Fig. 4 for the case of the c-th cluster. For instance, the input for the Sylvester equation (23) solver is locally prepared according to the rest of the ADMM variables, then deployed to remote processes for computation. After all updates are completed (ensured by the presence of a synchronization barrier), the lighter updates for $\mathbf{Z}_c$, $\mathbf{V}_{c,\ell}$ and $\mathbf{X}_c$ can be carried out locally. In this manner, the dCoPC framework becomes readily parallelizable, lending itself well towards integration with distributed computing systems.

V. Experimental results

In this section, we present an experimental analysis of the proposed framework for the reconstruction of highly-incomplete time-varying 3D point clouds. We categorize the developed algorithms into *global* (CoPC, sCoPC) and *distributed* (dCoPC) techniques, where the former considers that the processing is conducted globally over the whole set of data at once, while the later considers that the data are separated and distributed into different interconnected devices. All algorithms have been implemented using the Julia scientific language [35].

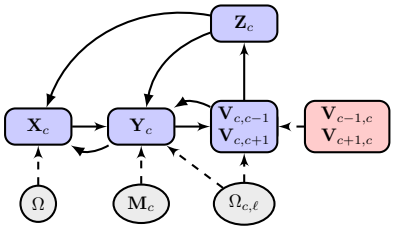

Fig. 4. Distributed protocol for low communication overhead between clusters. The red box indicates the required input from the next and the previous clusters.

A. Setup

1) Datasets: For our experiments we have used three animation models, namely, the "Handstand", "Samba" and "Squat" models from [28] which have been produced based on scanned images. In order to represent the imperfections of the scanned animated models, we produce the missing point positions based on the random uniform distribution, where the xyz-coordinates share the same under-sampling pattern. The reconstruction quality of the evaluated techniques is obtained by using the normalized mean root square error for all frames, $\mathrm{NMSE} = \frac{\|\mathbf{X}-\mathbf{M}\|_F}{\|\mathbf{M}\|_F}$, and the KG error [36], which is defined as $\mathrm{KGE} = 100 \cdot \frac{\|\mathbf{X}-\mathbf{M}\|_F}{\|\mathbf{X}-\mathcal{E}(\mathbf{M})\|_F}$, where $\mathcal{E}(\mathbf{M})$ denotes a matrix whose columns consist of the average vertex positions for all frames.

2) Parameter selection: The results of the proposed iterative techniques have been obtained upon the convergence of the algorithms, i.e., $i_{\max}$ has been set accordingly for each case. The convergence behavior of the proposed algorithms is determined by the parameters τ, γ and ξ, where τ (used in CoPC, sCoPC, dCoPC) determines the rank of the recovered matrix and thus the convergence speed, γ (used in sCoPC, dCoPC) determines the spatial coherence via the Laplacian matrix, and ξ (used in dCoPC) determines the temporal coherence between the successive clusters of frames. Specifically, large values of τ (or equivalently imposing low-rank to the reconstructed matrix) will result into fast convergence but also lower reconstruction quality [29, Theorem 1]. On the contrary, small values of τ may result into better reconstruction quality but the algorithm could diverge due to the small sampling ratio M. Moreover, the parameters γ and ξ are weighting the prior information concerning the spatial and temporal coherence respectively. The parameter of the dual variables ρ determines the convergence speed of the ADMM techniques; after experimental evaluation we have selected the value $\rho = 1/10$ for the derived results. The cluster size for obtaining the results of dCoPC was set to 15.

3) Robust estimation of the Laplacian: The weighted graphs are constructed based on the following two techniques: the $\epsilon \in \mathbb{R}$ neighbourhoods (ϵ-N) and the $k \in \mathbb{N}$ nearest

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2018.2839911, IEEE Transactions on Multimedia

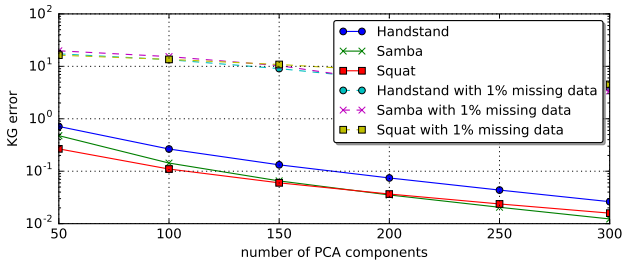

Fig. 5. KG error of the PCA-based reconstruction w.r.t the number of the principal components.

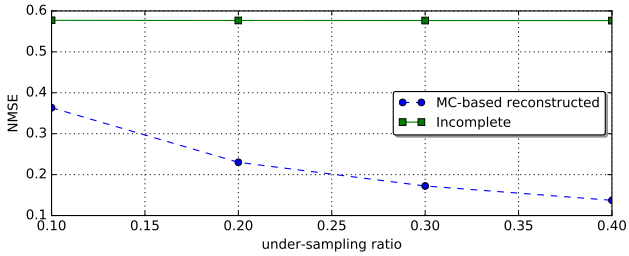

Fig. 6. Normalized mean-square-error (NMSE) for the reconstruction of the Laplacian matrix w.r.t. the under-sampling ratio of the animation matrix.

neighbours (k-NN). It is known that ϵ-N graphs are symmetric and are more geometric 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 $w_{i,j} = 1$ 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 $w_{i,j} = e^{\frac{-\|x_i - x_j\|_2^2}{2\theta^2}}$ if $(i,j) \in E_c$, and 0 otherwise, where θ is the variance which sets the threshold for edge existence. In this work we employ k-NN binary weighted graphs for the derivations of the Laplacian matrix, however from simulation results which are not presented in this work, we get similar results when thresholded Gaussian weighting is used.

In the case of incomplete geometry and no other prior information, the direct computation of the Laplacian matrix is not possible. A straightforward approach would be first to reconstruct the missing data based on Algorithm 1, and then compute the Laplacian. Although, this is a plausible approach, it could result into a noisy estimation for high under-sampling ratios. To overcome this problem, we propose to construct and exploit the graph of the *average point cloud*. The average point cloud is defined as $\bar{\mathbf{M}} = \sum_{f=1}^{F} \alpha_f \mathbf{M}_f \in \mathbb{R}^{3\times N}$, where α_f is the weighting factor for the f-th frame. Therefore, a *robust* technique for the reconstruction of the Laplacian is the following:

1) first reconstruct the incomplete point cloud sequence via Algorithm 1,
2) then build an average point cloud of all the reconstructed frames, and
3) finally compute the Laplacian matrix of this average point cloud.

Although it is known that the simple averaging of the positions of the points could result into visually implausible shape, the resulting mean shape retains the information about the proximity of the points in average, as it is justified by the experimental results. A more appropriate approach would be to compute the Laplacian based on [27], however this is out of the scope of this work.

4) Included techniques: In the next subsection, we investigate the quality performance of the proposed techniques, namely, **CoPC**: Matrix completion technique (Algorithm 1) which works on the animation matrix $\mathbf{M} \in \mathbb{R}^{3F\times N}$. This technique represents the geometry-myopic approach for the reconstruction of $\mathbf{M}$, where no prior information is required, and it belongs to the global case, where the processing is conducted at all the available data at once; **sCoPC**: We extend the CoPC technique by exploiting the spatial point cloud geometry, represented by the Laplacian matrix $\mathbf{L}$, and it also belongs to the global case; **dCoPC**: This technique represents the distributed case where the processing is conducted by different devices by dividing the animation set into clusters. These devices are assumed to be interconnected into a network.

Apart from the proposed techniques, we also consider the LSM technique [19] for the global and the distributed cases (dLSM), which exploits only the spatial coherence of the dynamic meshes. Also, we employ with two methods of Laplacian interpolation, one emploiting spatial (SLI) and the other both spatial and temporal coherence (STLI) [37]. SPI uses spatial Laplacian constraints to aid the completion of the animated mesh, while STLI complements the constraints imposed by the weigthed Laplacian with temporal information. More specifically, the motion vectors of known points are used to estimate the motion vectors of the neighbouring ones that are missing by solving a weighted Laplacian-based optimization problem.

B. Evaluation results

1) PCA-based consolidation: Before proceeding to the evaluation of the proposed techniques, let us investigate the performance of the PCA-based reconstruction. In Fig. 5, we depict the KG error of the reconstructed animation matrix (3) with respect to (w.r.t.) the number of the principal components that are being retained. Obviously, the number of the components determines the quality of the reconstruction, while the performance of the technique depends on the underlying correlations of the animation data, thus, the number of the required principal components is model dependent. Note that even in the case where the missing data is only 1%, the PCA-based technique fails to reconstruct the models. This result verifies the fact that the conventional PCA technique is not robust to outlier noise and thus not suitable for reconstruction based on missing data.

Obviously, the method for the construction of the graph Laplacian matrix (e.g. ϵ-N or k-NN) has major impact at

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2018.2839911, IEEE Transactions on Multimedia

the reconstruction performance. Due to the missing data, the constructed Laplacian may contain severe errors, even in the case of few (e.g. 10%) missing entries. However, the graph Laplacian of the average point cloud mitigates this effect via the averaging over the available frames. To justify these statements, in Fig. 6 we show the normalized square error

$$\text{NMSE}(u) = \frac{\|\mathbf{L} - \tilde{\mathbf{L}}_u\|_F^2}{\|\mathbf{L}\|_F^2}$$

where $\mathbf{L}$ is the true Laplacian matrix and $\tilde{\mathbf{L}}_u$ is the reconstructed based on the CoPC technique for different under-sampling ratios. We can observe that when the Laplacian matrix is being constructed based on the average of the incomplete point cloud, then it does provide any information even for the 10% case. On the contrary, the construction of the Laplacian matrix upon the CoPC provides an estimate of the average point cloud $\bar{\mathbf{M}}$ connectivity.

2) Reconstruction quality: Figure 7 shows the heatmap visualization of the normalized mean square visual error (NMSVE) [38] for the three animation models, the "Handstand", "Samba" and "Squat" for the case of 70% missing points (i.e. 30% under-sampling ratio). We can observe that sCoPC preserves the details and achieves small KG errors. The results of NMSE and KG error for highly incomplete under-sampling ratios, i.e., 10%, 20% and 30%, are shown in Table II. Comparing among the techniques, we observe that sCoPC exhibits the smaller errors leverage from the underlying structure of the data and the average point cloud geometry. The matrix completion techniques, CoPC and sCoPC perform better for the "Squat" animation model, which is justified by the higher number of frames ($F = 250$). Note that working on square matrices ($3F = N$) favours the performance of the matrix completion techniques. Also note that for under-sampling ratios larger than 30%, the exploitation of the spatial coherence via LSM performs worse than CoPC, which only exploits the underlying low-rank structure of the animation matrix.

3) Estimated Laplacian: In the previous results, we have assumed that the Laplacian matrix is perfectly known, being constructed based on the complete animation matrix. Obviously, this is not the case in a more real case setup where only the 3D points are available. On this premise, we investigate the performance of the techniques when the Laplacian matrix is computed based on the reconstructed animation matrix. In particular, for its reconstruction we use the CoPC technique which does not need any other prior information about the geometry. In Table II we show the results for the "noisy" Laplacian case for the cases of LSM and sCoPC. Recall that in our results the Laplacian matrix which is used by the algorithms has been obtained by the MC algorithm. The performance of LSM is constrained by the reconstruction quality of the Laplacian matrix, hence even some small noise it may cause very large errors. Fig. 8 shows three frames (1, 88 and 109) of the "Handstand" animation model, comparing the original, the LSM and the sCoPC techniques for the case of 30% under-sampling ratio. We can observe that sCoPC achieves much better reconstruction quality compared to LSM, reaching lower than the half of the LSM KG error in the case of "Squat" model. However, for the case of 10% under-sampling ratio, the techniques exhibit large KG errors (> 2) which it is usually not acceptable.

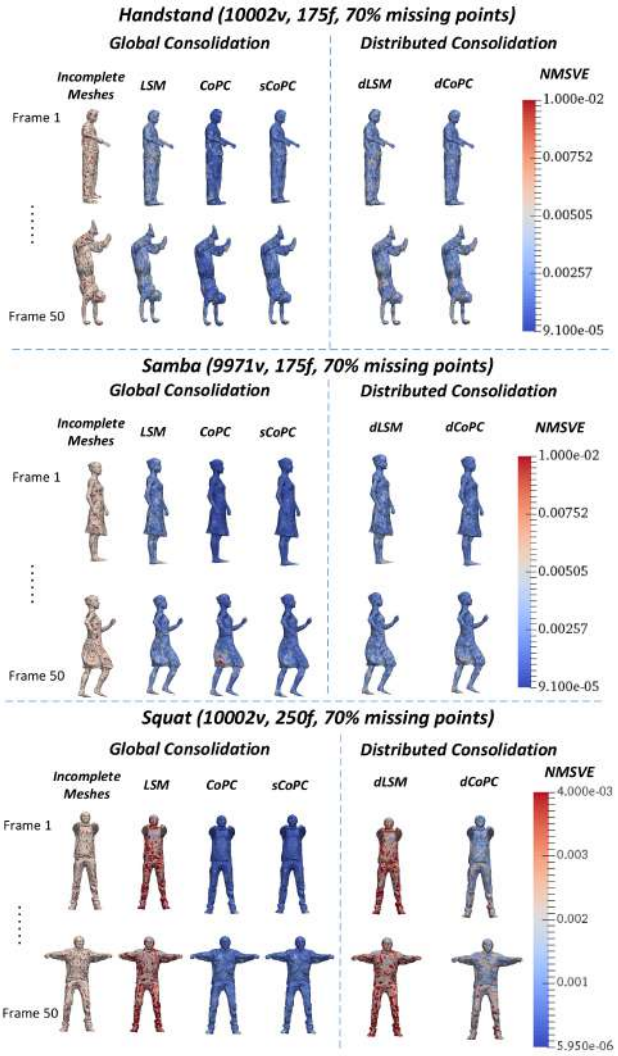

Fig. 7. Heatmap visualization of the normalized mean square visual error (NMSVE) for the three animation models. Comparison between the proposed techniques CoPC, sCoPC and the LSM [19] for global and distributed consolidation.

Fig. 9 depicts the obvious advantages of CoPC-based methods to spatial Laplacian interpolation. Spatiotemporal Laplacian interpolation's first frames similarly suffer (since completion on the first frame is performed using only spatial Laplacian constraints), but recovers quality as the animation unfolds. Still, points in regions with fast motion (such as the feet in the handstand animation) fail to be reconstructed reliably, resulting in relatively large errors. Conversely, the proposed methods are shown to be more robust with respect to abrupt changes and fast motion, simultaneously preserving low and consistent error rates across the whole dataset.

4) Computational complexity: Concerning the computational complexity of the clustered consolidation dCoPC, it is known that, the distributed techniques allow for parallelization of the computationally heavy part of the per-cluster processing,

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2018.2839911, IEEE Transactions on Multimedia

TABLE II
QUALITY MEASUREMENTS FOR THE PROPOSED CONSOLIDATION TECHNIQUES: COPC, SCOPC, DCOPC

Ratio	Table II.A (perfectly known Laplacian)						Table II.B (estimated Laplacian)				Table II.C (distributed)			
	Normalized MSE			KG error			Normalized MSE		KG error		Normalized MSE		KG error	
	LSM	CoPC	sCoPC	LSM	CoPC	sCoPC	LSM	sCoPC	LSM	sCoPC	dLSM	dCoPC	dLSM	dCoPC
Handstand [28]														
10%	0.0091	0.0217	0.0087	2.13	4.42	1.77	0.0533	0.0421	10.82	8.56	0.0164	0.0139	3.33	2.83
30%	0.0048	0.0034	0.0032	1.14	0.82	0.76	0.0062	0.0035	1.27	0.72	0.0064	0.0057	1.29	1.16
50%	0.0042	0.0024	0.0019	0.86	0.49	0.39	0.0042	0.0032	0.86	0.65	0.0044	0.0037	0.86	0.76
Samba [28]														
10%	0.0091	0.0152	0.0067	2.13	3.58	1.57	0.0712	0.0548	16.78	12.90	0.0102	0.0118	2.40	2.79
30%	0.0048	0.0035	0.0027	1.14	0.82	0.63	0.0049	0.0032	1.15	0.74	0.0049	0.0045	1.15	1.05
50%	0.0033	0.0012	0.0012	0.78	0.29	0.29	0.0033	0.0018	0.78	0.43	0.0033	0.0029	0.79	0.70
Squat [28]														
10%	0.0119	0.0047	0.0045	2.26	0.89	0.85	0.0446	0.0248	8.44	4.69	0.0120	0.0144	2.27	2.73
30%	0.0064	0.0015	0.0015	1.22	0.28	0.28	0.0064	0.0026	1.22	0.46	0.0064	0.0053	1.00	0.99
50%	0.0044	0.0009	0.0009	0.83	0.16	0.16	0.0044	0.0015	0.84	0.29	0.0044	0.0036	0.83	0.68

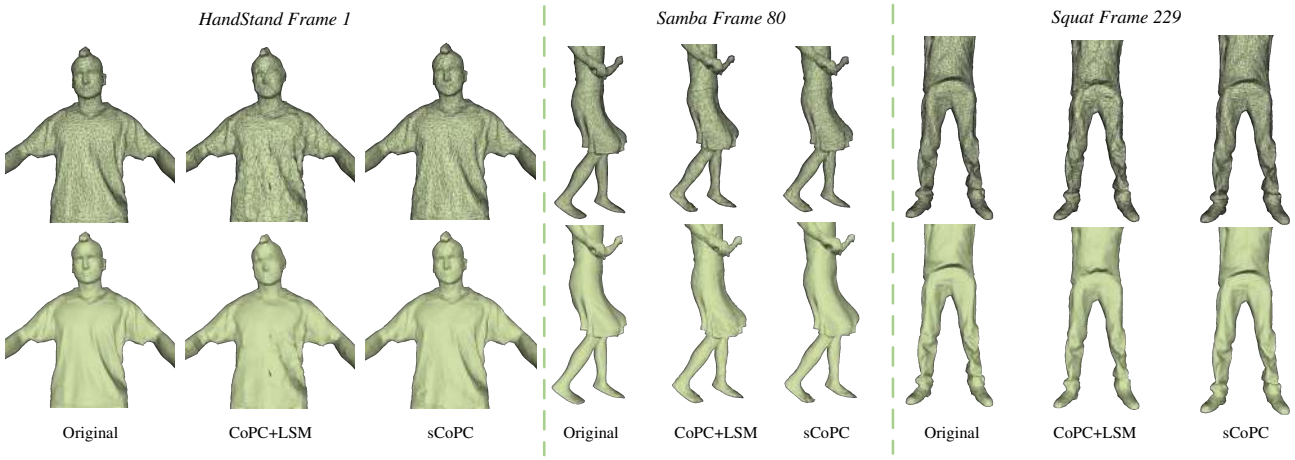

Fig. 8. Results for the noisy case with 30% under-sampling ratio (i.e., only 3000 vertices from 10002 are known). The Laplacian matrix has been estimated after averaging the reconstructed animation sequence provided by Algorithm 1.

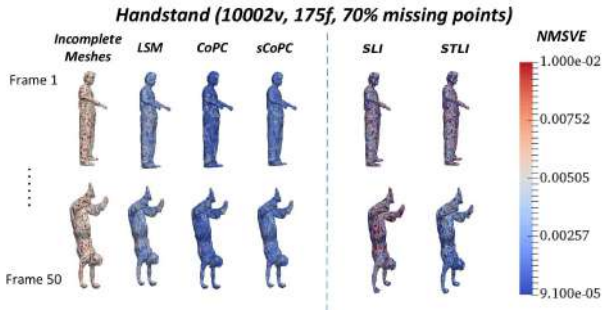

Fig. 9. Heatmap visualization of the normalized mean square visual error (NMSVE) for the three animation models. Comparison between the proposed techniques CoPC, sCoPC and the LSM [19], SLI [37] and STLI [37], showing NMSVE per vertex.

achieving significant reduction in computational complexity when compared to the global case of sCoPC which works on the whole dataset. The speed up for the case of 32 parallel streams is shown in Table III. The distributed consolidation provided at least 35% speed up for all three models, percentage that can be increased by increasing the computational streams.

TABLE III
COMPARISON OF INDICATIVE EXECUTION TIMINGS (IN SECONDS) FOR 32 PARALLEL STREAMS.

Technique	Model: Handstand	Samba	Squat
sCoPC - global (working on the whole dataset)	454s	496s	784s
dCoPC - distributed (working on dataset parts)	188s	187s	353s
Speedup	**40%**	**37%**	**45%**

VI. CONCLUSION

In this work, we have considered the consolidation of dynamic point clouds given only a small number of geometry data. First we have described a geometry-myopic technique which exploits only the structure of the underlying subspace of the geometry data. Then we have extended this technique so as to jointly exploit the spatial coherence of the 3D points. Since

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2018.2839911, IEEE Transactions on Multimedia

the animation data are usually massive, we have proposed a distributed technique with low communication overhead which is able to divide the animation matrix and perform in parallel manner. To overcome the ill-conditioning of this problem, we have proposed the exploitation of the time coherence between the successive frames. Through simulation results we have verified that the proposed techniques are able to faithfully recover the missing points.

APPENDIX

A. Proof of eq. (16):

Eq. (15) can be written as follows: $\mathbf{E}_\Omega \circ \mathbf{Y} - \rho\mathbf{Y} = \mathbf{E}_\Omega \circ \hat{\mathbf{M}} - \rho\mathbf{X}(i+1) - \mathbf{Z}(i) \Rightarrow \text{vec}(\mathbf{E}_\Omega \circ \mathbf{Y} - \mathbf{Y}) = \text{vec}\big(\mathbf{E}_\Omega \circ \hat{\mathbf{M}} - \rho\mathbf{X}(i+1) - \mathbf{Z}(i)\big) \Rightarrow \Big(\text{diag}\big(\text{vec}(\mathbf{E}_\Omega)\big) - \rho\mathbf{I}_{3FN}\Big)\text{vec}\big(\mathbf{Y}\big) = \text{vec}(\mathbf{E}_\Omega \circ \hat{\mathbf{M}} - \rho\mathbf{X}(i+1) - \mathbf{Z}(i)\big)$ where the last equation is obtained utilizing the properties of the vectorization function, i.e., $\text{vec}(\mathbf{AXB}) = (\mathbf{B}^T \otimes \mathbf{A})\text{vec}(\mathbf{X})$ and $\text{vec}(\mathbf{E}_\Omega \circ \mathbf{Y}) = \text{diag}\big(\text{vec}(\mathbf{E}_\Omega)\big)\text{vec}(\mathbf{Y})$ [39].

B. Proof of eq. (20):

Eq. (20) is in the form of the generalized Sylvester equation as it has been defined in [31]. To show this, let us write (19) as follows: $\mathbf{E}_\Omega \circ (\mathbf{Y} - \hat{\mathbf{M}}) + \gamma\mathbf{Y}\mathbf{L}^T\mathbf{L} + \rho(\mathbf{Y} - \mathbf{X}(i+1)) - \mathbf{Z}(i) = \mathbf{0} \Rightarrow \mathbf{E}_\Omega \circ \mathbf{Y} + \gamma\mathbf{Y}\mathbf{L}^T\mathbf{L} + \rho\mathbf{Y} + \mathbf{C} = \mathbf{0}$, where i indicates the ADMM iteration. Based on the properties of the Hadamard product [40], the first term of the last equation can be written as: $\mathbf{E}_\Omega \circ \mathbf{Y} = \sum_{i=1}^{3FN} \mathbf{E}_{ii}\mathbf{Y}\text{diag}([\mathbf{E}_\Omega]_i)$, where $\mathbf{E}_{ii}$ 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. Based on the definitions of (21) - (22), it is straightforward to obtain (20). Also, it is known that the necessary 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 (20) includes the term $\rho\mathbf{I}_{3FN}$, this condition is satisfied for $\rho > 0$.

C. Proof of Proposition 1

For $\tau = 0$ eq. (18) is expressed as:

$$\min_{\mathbf{X}} \frac{1}{2}\|\mathbf{E}_\Omega \circ (\mathbf{X} - \hat{\mathbf{M}})\|_F^2 + \frac{\gamma}{2}\|\mathbf{X}\mathbf{L}\|_F^2 \tag{28}$$

which has the minimizer: $\mathbf{E}_\Omega \circ \mathbf{X} - \mathbf{E}_\Omega \circ \hat{\mathbf{M}} + \gamma\mathbf{X}\mathbf{L}^T\mathbf{L} = \mathbf{0}$. Following the analysis of Appendix A, the minimizer of (28) can be obtained as the solution of the $3FN \times 3FN$ linear system:

$$\big(\text{diag}(\text{vec}(\mathbf{E}_\Omega)) + \gamma\mathbf{L}^T\mathbf{L} \otimes \mathbf{I}_{3F}\big)\text{vec}(\mathbf{X}) = \text{vec}\big(\mathbf{E}_\Omega \circ \hat{\mathbf{M}}\big) \tag{29}$$

It can be seen that the non zero values of the right hand side vector are the anchor points of the mesh $\mathbf{M}$, i.e., $\text{vec}(\mathbf{R}_e) = \Pi \cdot \text{vec}(\mathbf{E}_\Omega \circ \mathbf{M})$ where Π is a matrix that rearranges the entries of the vector $\text{vec}(\mathbf{E}_\Omega \circ \mathbf{M})$ by grouping first the zero and then the non zero values. Hence, by applying the rearrangement matrix to (29), we have:

$$\begin{aligned} &\Pi\big(\text{diag}(\text{vec}(\mathbf{E}_\Omega)) + \gamma\mathbf{L}^T\mathbf{L} \otimes \mathbf{I}_{3F}\big)\Pi\Pi\text{vec}(\mathbf{X}) = \mathbf{z} \\ \Rightarrow &\Big(\Pi\text{diag}(\text{vec}(\mathbf{E}_\Omega))\Pi + \gamma\Pi\big(\mathbf{L}^T\mathbf{L} \otimes \mathbf{I}_{3F}\big)\Pi\Big)\Pi\text{vec}(\mathbf{X}) = \mathbf{z} \end{aligned} \tag{30}$$

where $\Pi\Pi = \mathbf{I}$ and $\mathbf{z} \triangleq \Pi\text{vec}\big(\mathbf{E}_\Omega \circ \hat{\mathbf{M}}\big)$. Note that $\Pi\,\text{diag}(\text{vec}(\mathbf{E}_\Omega))\Pi = \begin{bmatrix} \mathbf{0}_{3F(N-K)\times 3FN}^T & \mathbf{I}_{3FK\times 3FN}^T \end{bmatrix}^T$ and $\Pi\,\text{vec}\big(\mathbf{E}_\Omega \circ \hat{\mathbf{M}}\big) = \begin{bmatrix} \mathbf{0}_{3F(N-K)\times 1}^T & \text{vec}(\mathbf{R})^T \end{bmatrix}^T$. Hence, (30) can be decomposed into the following two systems:

$$\begin{cases} (\gamma\mathbf{L}^T\mathbf{L} \otimes \mathbf{I}_{N-K})\text{vec}(\mathbf{X}) &= \mathbf{0}_N \\ (\mathbf{I}_{3FK} + \gamma\mathbf{L}^T\mathbf{L} \otimes \mathbf{I}_K)\text{vec}(\mathbf{X}) &= \text{vec}(\mathbf{R}) \end{cases}$$

or equivalently

$$\begin{cases} \mathbf{X}\mathbf{L}^T\mathbf{L} = \mathbf{0} \\ \mathbf{X}(\mathbf{I}_N + \gamma\mathbf{L}^T\mathbf{L}) = \mathbf{R} \end{cases} \Rightarrow \begin{cases} \mathbf{X}\mathbf{L}^T\mathbf{L} = \mathbf{0} \\ \mathbf{X}\mathbf{L}_e\mathbf{L}_e^T = \mathbf{R}_e\mathbf{L}_e^T \end{cases}$$
$$\Rightarrow \begin{cases} \mathbf{X}\mathbf{L} = \mathbf{0} \\ \mathbf{X}\mathbf{L}_e = \mathbf{R}_e \end{cases}$$

This can be written as:

$$\mathbf{X}\begin{bmatrix} \mathbf{I}_N + \gamma\mathbf{L}^T\mathbf{L} & \gamma\mathbf{L}^T\mathbf{L} \end{bmatrix} = \begin{bmatrix} \mathbf{R} & \mathbf{0}_{3F\times N} \end{bmatrix}$$

which is identical with to the solution of (17).

D. Solution of eq. (27)

The augmented Lagrangian of (27) is written as: $\mathcal{L}\big(\{\mathbf{X}_c\}_c, \{\mathbf{Y}_c\}_c, \{\mathbf{Z}_{c,\ell}\}_{c,\ell}, \{\mathbf{V}_c\}_c\big) = \sum_{c=1}^{C}\Big[\tau\|\mathbf{X}_c\|_* + \frac{1}{2}\|\mathbf{E}_{\Omega_c} \circ (\mathbf{Y}_c - \mathbf{M}_c)\|_F^2 + \sum_{\ell\in\mathcal{N}_c}\frac{1}{2}\|\mathbf{E}_{\Omega_{c,\ell}} \circ \mathbf{Y}_c - \mathbf{V}_{c,\ell}\|_F^2 + \langle\mathbf{Z}_c, \mathbf{X}_c - \mathbf{Y}_c\rangle + \frac{\rho}{2}\|\mathbf{X}_c - \mathbf{Y}_c\|_F^2\Big]$. The solution of (27) can be obtained by expressing the optimality conditions with respect to the primal and dual variables:

- The minimization of $\mathcal{L}$ with respect to $\mathbf{X}_c$ requires the computation of the subgradient $\frac{\partial\mathcal{L}}{\partial\mathbf{X}_c} = \tau\frac{\partial\|\mathbf{X}_c\|_*}{\partial\mathbf{X}_c} + \mathbf{V}_c + \rho(\mathbf{X}_c - \mathbf{Y}_c)$, which alternatively can be expressed as: $\mathbf{X}_c = arg\min_{\mathbf{X}_c}\frac{\tau}{\rho}\|\mathbf{X}_c\|_* + \frac{1}{2}\|\mathbf{X}_c - (\mathbf{Y}_c - \frac{1}{\rho}\mathbf{V}_c)\|_F^2$. Based on [29, Theorem 2.1], the minimizer of this expression is obtained using the SVT operator on the matrix $\mathbf{Y}_c - \frac{1}{\rho}\mathbf{V}_c$.
- The minimization of $\mathcal{L}$ with respect to $\mathbf{Y}_c$ is written as $\frac{\partial\mathcal{L}}{\partial\mathbf{Y}_c} = \mathbf{E}_{\Omega_c} \circ \mathbf{Y}_c + \gamma\mathbf{Y}_c(\mathbf{L}_c^T\mathbf{L}_c) - \mathbf{E}_{\Omega_c} \circ \mathbf{M}_c + \sum_{\ell\in\mathcal{N}_c}\big(\mathbf{E}_{\Omega_{c,\ell}} \circ \mathbf{Y}_c - \mathbf{Z}_{c,\ell}\big) - \mathbf{V}_c - \rho(\mathbf{X}_c - \mathbf{Y}_c) = 0 \Rightarrow \big(\mathbf{E}_{\Omega_c} + \sum_{\ell\in\mathcal{N}_c}\mathbf{E}_{\Omega_{c,\ell}} + \rho\mathbf{E}\big) \circ \mathbf{Y}_c = \mathbf{E}_{\Omega_c} \circ \mathbf{M}_c + \sum_{\ell\in\mathcal{N}_c}\mathbf{Z}_{c,\ell} + \mathbf{V}_c + \rho\mathbf{X}_c$. Following Appendix B analysis, the minimizer of the previous expression is provided by the solution of the following system of equations: $\mathbf{A}_c\text{vec}(\mathbf{Y}_c) = \mathbf{b}_c$, where $\mathbf{A}_c \triangleq \gamma\big((\mathbf{L}_c^T\mathbf{L}_c) \otimes \mathbf{I}_{fc}\big) + \text{diag}\big(\text{vec}(\xi\mathbf{E}_{\Omega_{c,\ell}} + \mathbf{E}_{\Omega_c} + \rho\mathbf{E})\big)$ and $\mathbf{b}_c \triangleq \text{vec}\big(\xi\mathbf{V}_{c,\ell} + \mathbf{Z}_c + \rho\mathbf{X}_c + \mathbf{E}_{\Omega_c} \circ \mathbf{M}_c\big)$
- The minimization $\mathcal{L}$ with respect to $\mathbf{V}_{c,\ell}$ requires the computation of the gradient $\frac{\partial\mathcal{L}}{\partial\mathbf{V}_{c,\ell}} = \mathbf{E}_{\Omega_{c,\ell}} \circ \mathbf{Y}_c - \mathbf{V}_{c,\ell} + \mathbf{E}_{\Omega_{\ell,c}} \circ \mathbf{Y}_\ell - \mathbf{V}_{\ell,c}$. Since $\mathbf{V}_{\ell,c} = \mathbf{V}_{c,\ell}$, we have that: $\mathbf{V}_{c,\ell} = \frac{\mathbf{E}_{\Omega_{c,\ell}} \circ \mathbf{Y}_c + \mathbf{E}_{\Omega_{\ell,c}} \circ \mathbf{Y}_\ell}{2}$.

REFERENCES

[1] T. Weise, B. Leibe, and L. V. Gool, "Fast 3d scanning with automatic motion compensation," in *2007 IEEE Conference on Computer Vision and Pattern Recognition*, Jun. 2007, pp. 1–8.

[2] J. Davis, D. Nehab, R. Ramamoorthi, and S. Rusinkiewicz, "Spacetime stereo: a unifying framework for depth from triangulation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 2, pp. 296–302, Feb. 2005.

[3] C. L. Zitnick, S. B. Kang, M. Uyttendaele, S. Winder, and R. Szeliski, "High-quality video view interpolation using a layered representation," *ACM Trans. Graph.*, vol. 23, no. 3, pp. 600–608, Aug. 2004. [Online]. Available: http://doi.acm.org/10.1145/1015706.1015766

[4] M. Alexa, J. Behr, D. Cohen-Or, S. Fleishman, D. Levin, and C. T. Silva, "Computing and rendering point set surfaces," *IEEE Transactions on Visualization and Computer Graphics*, vol. 9, no. 1, pp. 3–15, Jan. 2003. [Online]. Available: http://dx.doi.org/10.1109/TVCG.2003.1175093

[5] Y. Lipman, D. Cohen-Or, D. Levin, and H. Tal-Ezer, "Parameterization-free projection for geometry reconstruction," *ACM Trans. Graph.*, vol. 26, no. 3, Jul. 2007. [Online]. Available: http://doi.acm.org/10.1145/1276377.1276405

[6] S. Liu, K.-C. Chan, and C. C. L. Wang, "Iterative consolidation of unorganized point clouds," *IEEE Comput. Graph. Appl.*, vol. 32, no. 3, pp. 70–83, May 2012. [Online]. Available: http://dx.doi.org/10.1109/MCG.2011.14

[7] J. Wang, K. Xu, L. Liu, J. Cao, S. Liu, Z. Yu, and X. D. Gu, "Consolidation of low-quality point clouds from outdoor scenes," in *Proceedings of the Eleventh Eurographics/ACMSIGGRAPH Symposium on Geometry Processing*, ser. SGP '13. Aire-la-Ville, Switzerland, Switzerland: Eurographics Association, 2013, pp. 207–216. [Online]. Available: http://dx.doi.org/10.1111/cgf.12187

[8] M. Berger, A. Tagliasacchi, L. M. Seversky, P. Alliez, G. Guennebaud, J. A. Levine, A. Sharf, and C. T. Silva, "A survey of surface reconstruction from point clouds," *Computer Graphics Forum*, 2016. [Online]. Available: http://dx.doi.org/10.1111/cgf.12802

[9] E. J. Candès and B. Recht, "Exact matrix completion via convex optimization," *Foundations of Computational Mathematics*, vol. 9, no. 6, p. 717, 2009. [Online]. Available: http://dx.doi.org/10.1007/s10208-009-9045-5

[10] S. B. 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. [Online]. Available: http://dx.doi.org/10.1561/2200000016

[11] M. Fazel, "Matrix rank minimization with applications," *Elec Eng Dept Stanford University*, vol. 54, pp. 1–130, 2002.

[12] 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.

[13] Y. Peng, A. Ganesh, J. Wright, W. Xu, and Y. Ma, "Rasl: Robust alignment by sparse and low-rank decomposition for linearly correlated images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 11, pp. 2233–2246, Nov. 2012.

[14] Y. Deng, Y. Liu, Q. Dai, Z. Zhang, and Y. Wang, "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.

[15] M. Alexa and W. Muller, "Representing Animations by Principal Components," *Computer Graphics Forum*, 2000.

[16] F. Cazals and J. Giesen, "Delaunay triangulation based surface reconstruction: Ideas and algorithms," in *EFFECTIVE COMPUTATIONAL GEOMETRY FOR CURVES AND SURFACES*. Springer, 2006, pp. 231–273.

[17] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle, "Surface reconstruction from unorganized points," *SIGGRAPH Comput. Graph.*, vol. 26, no. 2, pp. 71–78, Jul. 1992. [Online]. Available: http://doi.acm.org/10.1145/142920.134011

[18] S. Xiong, J. Zhang, J. Zheng, J. Cai, and L. Liu, "Robust surface reconstruction via dictionary learning," *ACM Trans. Graph.*, vol. 33, no. 6, pp. 201:1–201:12, Nov. 2014. [Online]. Available: http://doi.acm.org/10.1145/2661229.2661263

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

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

[21] D. Anguelov, P. Srinivasan, H.-C. Pang, D. Koller, S. Thrun, and J. Davis, "The correlated correspondence algorithm for unsupervised registration of nonrigid surfaces," in *Proceedings of the 17th International Conference on Neural Information Processing Systems*, ser. NIPS'04. Cambridge, MA, USA: MIT Press, 2004, pp. 33–40. [Online]. Available: http://dl.acm.org/citation.cfm?id=2976040.2976045

[22] M. Wand, B. Adams, M. Ovsjanikov, A. Berner, M. Bokeloh, P. Jenke, L. Guibas, H.-P. Seidel, and A. Schilling, "Efficient reconstruction of nonrigid shape and motion from real-time 3d scanner data," *ACM Trans. Graph.*, vol. 28, no. 2, pp. 15:1–15:15, May 2009. [Online]. Available: http://doi.acm.org/10.1145/1516522.1516526

[23] J. Sussmuth, M. Winter, and G. Greiner, "Reconstructing animated meshes from time-varying point clouds," in *Proceedings of the Symposium on Geometry Processing*, ser. SGP '08. Aire-la-Ville, Switzerland, Switzerland: Eurographics Association, 2008, pp. 1469–1476. [Online]. Available: http://dl.acm.org/citation.cfm?id=1731309.1731332

[24] V. Blanz, A. Mehl, T. Vetter, and H. P. Seidel, "A statistical method for robust 3d surface reconstruction from sparse data," in *Proceedings. 2nd International Symposium on 3D Data Processing, Visualization and Transmission, 2004. 3DPVT 2004.*, Sept. 2004, pp. 293–300.

[25] A. S. Lalos, I. Nikolas, E. Vlachos, and K. Moustakas, "Compressed sensing for efficient encoding of dense 3d meshes using model-based bayesian learning," *IEEE Transactions on Multimedia*, vol. 19, no. 1, pp. 41–53, Jan 2017.

[26] A. 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.

[27] L. Vasa, S. Marras, K. Hormann, and G. Brunnett, "Compressing Dynamic Meshes with Geometric Laplacians," *Computer Graphics Forum*, 2014.

[28] D. Vlasic, I. Baran, W. Matusik, and J. Popović, "Articulated mesh animation from multi-view silhouettes," *ACM Trans. Graph.*, vol. 27, no. 3, pp. 97:1–97:9, Aug. 2008. [Online]. Available: http://doi.acm.org/10.1145/1360612.1360696

[29] J.-F. Cai, E. J. Candes, and Z. Shen, "A singular value thresholding algorithm for matrix completion," *SIAM Journal on Optimization*, vol. 20, no. 4, pp. 1956–1982, 2010. [Online]. Available: http://dx.doi.org/10.1137/080738970

[30] G. H. Golub and C. F. Van Loan, *Matrix Computations (4th Ed.)*. Baltimore, MD, USA: Johns Hopkins University Press, 2013.

[31] B. Zhou and G.-R. Duan, "On the generalized sylvester mapping and matrix equations," *Systems & Control Letters*, vol. 57, no. 3, pp. 200 – 208, 2008. [Online]. Available: //www.sciencedirect.com/science/article/pii/S0167691107001156

[32] Y. Saad and M. H. Schultz, "Gmres: A generalized minimal residual algorithm for solving nonsymmetric linear systems," *SIAM J. Sci. Stat. Comput.*, vol. 7, no. 3, pp. 856–869, Jul. 1986. [Online]. Available: http://dx.doi.org/10.1137/0907058

[33] T. F. Chan and W. L. Wan, "Analysis of projection methods for solving linear systems with multiple right-hand sides," *SIAM J. Sci. Comput.*, vol. 18, no. 6, pp. 1698–1721, Nov. 1997.

[34] A. Bouhamidi and K. Jbilou, "A note on the numerical approximate solutions for generalized Sylvester matrix equations with applications," *Appl. Math. Comput.*, vol. 206, no. 2, pp. 687–694, 2008. [Online]. Available: http://dx.doi.org/10.1016/j.amc.2008.09.022

[35] J. Bezanson, A. Edelman, S. Karpinski, and V. B. Shah, "Julia: A fresh approach to numerical computing," *CoRR*, vol. abs/1411.1607, 2014. [Online]. Available: http://arxiv.org/abs/1411.1607

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

[37] G. Arvanitis, A. S. Lalos, K. Moustakas, and N. Fakotakis, "Weighted regularized Laplacian interpolation for consolidation of highly incomplete time varying point clouds," in *3DTV Conference: The True Vision - Capture, Transmission and Display of 3D Video*, Jun. 2017, pp. 1–4.

[38] Z. Karni and C. Gotsman, "Compression of soft-body animation sequences," *Computers & Graphics*, vol. 28, no. 1, pp. 25 – 34, 2004.

[39] J. R. Magnus and H. Neudecker, *Matrix Differential Calculus with Applications in Statistics and Econometrics*, 2nd ed. John Wiley, 1999.

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