WEIGHTED REGULARIZED LAPLACIAN INTERPOLATION FOR CONSOLIDATION OF HIGHLY-INCOMPLETE TIME VARYING POINT CLOUDS

Gerasimos Arvanitis, Aris S. Lalos, Konstantinos Moustakas, Nikos Fakotakis

Dept. of Electrical and Computer Engineering
University of Patras, Rio, Patras, Greece

ABSTRACT

Recently, there has been increasing interest in real-time processing of time varying point clouds (TVPCs) ideally suited for real-time applications such as immersive tele-presence systems and gaming. Although the resolution and accuracy of the modern 3D scanners are constantly improving, the captured 3D point clouds are usually highly incomplete, stressing the need of completion approaches with low computational requirements. In this paper, we introduce a novel regularized Laplacian interpolation approach for a fast and effective consolidation of a TVPC. Extensive evaluation studies, carried out using a collection of different incomplete TVPCs, verify that the proposed technique achieves plausible reconstruction output despite the constraints posed by arbitrarily complex and motion scenarios.

Index Terms — Incomplete time varying 3D point clouds, missing vertices, weighted Laplacian interpolation

1. INTRODUCTION

In recent years, there has been increasing interest in the development of new generation image sensors and 3D scanning techniques. However, the new 3D acquisition systems are still unable to capture the full surface at once. Most scanned shapes are likely to exhibit large holes, even in scenarios where multiple sensors are placed around the subject, and these effects are attributed to occlusions, limited sensor range capabilities, high light absorption and low surface albedo [1]. Real time 3D captured data plays an important role in fields ranging from modeling dynamic scene geometry and holographic communication to plausible created avatars in games and others. In all these applications, the online consolidation of TVPC still remains a challenging issue since it requires the recovery of a large amount of missing data, increasing significantly the computational complexity and the execution time.

In literature, several works focus on 3D static point cloud consolidation, having presented excellent results applied to incomplete static point clouds. However, a little attention has been given to the consolidation of TVPCs. Most of the methods try to solve the problem by isolating each static point cloud of the sequence and handle it individually as a common 3D point cloud but without taking into account the temporal coherence between sequential frames [2]. Other approaches use a template prior in order to produce a temporal coherent TVPC, nonetheless, these approaches are not ideal for real-time applications because the entire captured point cloud sequence is required in advance, before the execution of the process [3]. These limitations motivated us to develop a fast TVPC consolidation approach that can be applied online, exactly when a new incomplete static point cloud is captured, taking advantage of the knowledge provided by the previous consolidated static point cloud.

Laplacian interpolation has been used extensively in image processing [4], [5] providing extremely good results even in cases when a large amount of data is missing. In the field of TVPC consolidation, Laplacian interpolation has been used mainly for morphing [6], [7]. Despite the computational effectiveness of Laplacian interpolation, the consolidated results are smoothed. Motivated by the aforementioned limitations, we introduce a novel consolidation technique that is based on a weighted regularized Laplacian interpolation approach that efficiently exploits the coherences between motion vectors of known vertices. To further enhance the recovery of the missing data, we suggest adding a regularization constraint that further exploits the sparse variations in the motion vector differences of a vertex and the average motion vector of its neighbors. The proposed approach is ideally suited for online settings where the point cloud sequence is not known a priori and is dynamically generated.

The rest of this paper is organized as follows: Section 2 presents an overview of our method. In this section we describe in detail the assumptions and we introduce proposed method workflow. Section 3 presents our experimental results and we compare them with other state-of-art methods. Section 4 draws the conclusions.

2. CONSOLIDATION OF HIGHLY-INCOMPLETE TVPC

2.1. Initial Assumptions and Preliminaries

Let us assume the existence of a sequence of static point clouds $M$ so that $\mathcal{M} = \{ M_1, M_2, \ldots, M_n \}$. Each static point cloud consists of $k$ vertices represented as a vector $v = [x, y, z]$ in a 3D coordinate space $x, y, z \in \mathbb{R}^3$ and $v \in \mathbb{R}^3$. We assume that the scanned TVPC is highly incomplete and denoted by $\mathcal{M}' = \{ M'_1, M'_2, \ldots, M'_n \}$ where $M'_i \subset M$, $\forall i = 2, n$. Fig. 1 depicts an example of a highly incomplete model assuming different number of known points.

![Figure 1: Missing areas are represented with white color while the known are represented with red. The percentage of considered known points are: (a) 10%, (b) 30%, (c) 50% (Squat model).](image)

The connectivity of vertices in a static point cloud is unknown. For this reason, we construct it based on the following two techniques: the $\epsilon \in \mathbb{R}$ neighborhoods ($\epsilon - N$) and the $\kappa \in \mathbb{N}$ nearest
neighbors ($k - NN$). It is known that, $\epsilon - N$ graphs are symmetric and are more geometric meaningful, although depending on the choice of the parameter $\epsilon$ they could lead to heavy or disconnected graphs. On the other hand, the choice of the parameter for $k \in \mathbb{N}$ graphs is more straightforward, usually leading to connected graphs. Despite the fact that the position of vertices change from frame to frame, the connectivity remains the same. In this work we assume that the topology is global, meaning that every static point cloud of the same sequence has the same topology over time [8]. We define as $C \in \mathbb{R}^{k \times k}$ the binary adjacency matrix with the following elements:

$$c_{ij} = \begin{cases} 
1 & \text{if } i, j \in S \\
0 & \text{otherwise} \end{cases} \quad (1)$$

where $S$ is a matrix estimated by $k - NN$ or $\epsilon - N$ process and consists of the neighbors of each vertex (first ring area). The adjacency matrix is estimated once and it can be used during the consolidation of any other frame.

### 2.2. Overview of our Method

The point cloud consolidation process takes place sequentially starting from the $M_1'$ and continuing until all point clouds of the sequence have been consolidated. Only the previous (consolidated) and the currently incomplete frame are required for the process, making the proposed schema ideal for real-time applications. In this way, our approach can be directly applied in real time 3D scanning scenarios. On the contrary, other methods require the entire knowledge of the incomplete point cloud sequence before the execution of the consolidated process. Fig. 2 briefly illustrates the proposed schema for a real-time consolidation of a TVPC.

![TVPC consolidation schema.](Image)

**Figure 2:** TVPC consolidation schema.

#### 2.2.1. Weighted Laplacian Matrix

A binary Laplacian matrix provides information about the connectivity of vertices. Nevertheless, weighted Laplacian matrices are able to provide additional geometric information that can be efficiently exploited by a variety of processes. In this work, we suggest constructing a modified weighted Laplacian matrix that takes into account: (a) the spatial coherence between connected vertices and (b) a prioritization ranking of the vertices. The prioritization of a vertex $i$ is performed by assigning an integer factor $p_i \geq 1$ expressing the proximity to an already known vertex. Using this annotation rule, the highest value $N$ is assigned to known vertices, the value $N - 1$ is assigned to vertices that are directly connected with known vertices while 1 is assigned to the most remote vertices. Therefore the prioritization of each vertex is determined by the values $p = [p_1, p_2, \ldots, p_k]$, where $p_i \in \{1, N\}$. Fig. 3 presents an example of vertices with different prioritization values. We assume that adjacent vertices exhibit similar behavior because they share common topological characteristics. In this way, we use a Gaussian kernel for expressing this spatial relation. The weighted adjacency matrix takes into account the two mentioned parameters which are applied to the adjacency matrix according to:

$$C_{w_{ij}(t)} = \begin{cases} 
p_{i} \cdot e^{-\|v_{i(t-1)} - v_{j(t-1)}\|^2}, & \text{if } c_{ij} = 1 \\
0 & \text{otherwise} \end{cases} \quad (2)$$

where $(t)$ and $(t-1)$ denotes the current and previous time indices (frames). The proposed weighted Laplacian matrix is estimated by:

$$L_w = D - C_w \quad (3)$$

where $D = \text{diag}(D_1, \ldots, D_k)$ is a diagonal matrix with $D_i = \sum_{j=1}^{k} C_{w_{ij}}$.

#### 2.2.2. Weighted Regularized Laplacian Interpolation

In this paragraph, we present a novel Laplacian interpolation approach using the weighted Laplacian matrix described in Eq.(3). According to [9], a way to interpolate a triangulated 3D model with a curved surface in a three-dimensional space is by putting constraints on the Laplacian $L$. The proposed Laplacian matrix encloses all the necessary constrains in order to be efficiently used by weighted regularized Laplacian interpolation (WRLI). The process is applied to the motion vector of known vertices managing to overcome the smoothness limitation of the original algorithm. The motion vector $\delta_i$ represents the distance of a vertex $v_i$ between two sequential frames (time indices):

$$\delta_i = |v_{i(t-1)} - v_{i(t)}| \quad \forall i = 1, k \quad (4)$$

As a result, we define $d = [\delta_1, \delta_2, \ldots, \delta_k]$. The Laplacian of $d$ is written as:

$$\theta = L_w d \quad (5)$$

in which $\theta$ is the vector containing the elements $(\Delta \delta)$. The vector $d$ can be split into two parts: $d_1 \in \mathbb{R}^{k' \times 3}$ and $d_2 \in \mathbb{R}^{k'' \times 3}$, where the vector $d_3$ contains the motion vectors of known vertices, $d_2$ contains the unspecified values and $k'$ is the number of known vertices. In a similar way, we can partition $L_w$ into four parts:

$$L_w = \begin{bmatrix} L_{w_{11}} & L_{w_{12}} \\ L_{w_{21}} & L_{w_{22}} \end{bmatrix} \quad (6)$$

$L_{w_{11}} \in \mathbb{R}^{k' \times k'}$, $L_{w_{12}} \in \mathbb{R}^{k' \times k''}$, $L_{w_{21}} \in \mathbb{R}^{k'' \times k'}$, $L_{w_{22}} \in \mathbb{R}^{k'' \times k''}$. We minimize the Euclidean norm $|\theta|$ according to:

$$|\theta| = |L_w d| = \begin{bmatrix} L_{w_{11}} & L_{w_{12}} \\ L_{w_{21}} & L_{w_{22}} \end{bmatrix} \begin{bmatrix} d_1 \\ d_2 \end{bmatrix} \quad (7)$$

$$= \begin{bmatrix} L_{w_{11}} & L_{w_{12}} \\ L_{w_{21}} & L_{w_{22}} \end{bmatrix} \begin{bmatrix} d_1 \\ d_2 \end{bmatrix}$$

![Known points.](Image)
This is equivalent to finding the least squares solution to:

\[
\left( \begin{array}{c}
L_{w_{12}} \\
L_{w_{22}}
\end{array} \right) \mathbf{d}_2 = -\left( \begin{array}{c}
L_{w_{11}} \\
L_{w_{21}}
\end{array} \right) \mathbf{d}_1 
\]

which is a system of \( k \) equations with \( k - k' \) variables. The well-known least squares solution to this system of equations is given by:

\[
\mathbf{d}_2 = \left( \begin{array}{c}
L_{w_{12}} \\
L_{w_{22}}
\end{array} \right)^\top \left( \begin{array}{c}
L_{w_{12}} \\
L_{w_{22}}
\end{array} \right)^{-1} \left( \begin{array}{c}
L_{w_{12}} \mathbf{d}_1 \\
L_{w_{22}} \mathbf{d}_1
\end{array} \right) 
\]

In order to add further constraints to the problem, given that it is overdetermined, we suggest using the \( l_1 \) regularizer, that efficiently exploits the fact that there are small variations in the motion vectors of the given vertex and the mean motion vector of its neighbors. Thus, we suggest estimating the unknown motion vectors \( \mathbf{d}_2 \) by solving the following Lasso problem:

\[
\arg\min_{\mathbf{a}} 0.5\| \mathbf{P} - \mathbf{Rz} \|_2^2 + \gamma \| \mathbf{z} \|_1 
\]

where \( \mathbf{P} = \mathbf{Qd}_1 - \mathbf{R}\hat{\mathbf{d}}_2 \), \( \mathbf{Q} = \left( \begin{array}{c}
L_{w_{11}} \\
L_{w_{21}}
\end{array} \right) \) and \( \mathbf{R} = \left( \begin{array}{c}
L_{w_{12}} \\
L_{w_{22}}
\end{array} \right) \).

The coherence between motion vector of neighboring vertices enhance the recovery efficiency of \( \mathbf{z} \) by efficiently exploiting its sparsity. The evaluated motion vectors are then used for estimating the coordinates of the missing vertices by efficiently exploiting its sparsity. The evaluated motion vectors are then used for estimating the coordinates of the missing vertices by efficiently exploiting its sparsity.

\[
v_{2(i)} = v_{2(i-1)} + \mathbf{d}_2 
\]

where \( v_1 = [v_{11}, \ldots, v_{1k}] \) and \( v_2 = [v_{21}, \ldots, v_{2k}] \). Fig. 4 correspond to the known motion vectors and the estimated ones based on WRLI. The described process is briefly presented in the following Algorithm 1.

![Algorithm 1](image)

3. EXPERIMENTAL ANALYSIS AND RESULTS

In this section, we present the result of our approach using a variety of initial conditions for different incomplete TVPC. The evaluation of the consolidated results shows the effectiveness of our method even in complex motion scenarios with rapid changes in the point trajectories of sequential frames or in highly incomplete TVPC where only a small percentage of points are known.

![Figure 4: Motion vectors of known vertices (blue) and motion vectors estimated via WRLI (red).](image)

3.1. Experimental Setup

In all the experiments we have used a PC Intel core i7-4710HQ CPU @ 2.50GHz 2.50GHz, 8 GB RAM. The main core of the algorithms is written in Julia. The scanned models that are used are: Handstand (175 frames, 10002 vertices), Squat (250 frames, 10002 vertices) and Samba (175 frames, 9971 vertices). The quality of the consolidated results are evaluated using the normalized mean square visual error (NMSVE) as described in [11] and heatmap visualization to highlight the difference between consolidated and original static point clouds \( |M_i - \hat{M}_i| \) \( \forall i = 1, n \).

3.2. Experimental Results

In this subsection, we investigate the quality performance of the proposed technique compared with other SoA techniques, namely:

- **LSM**: Least-square meshes algorithm [12] is described as the solution of an extended system of equations \( \left[ L^T_{I_k} L^T_{I_k} \right] \mathbf{x} \) \( \forall t = 1, \ldots, n \), for \( k' \) known vertices in the \( t \)-th frame.
- **MC**: Matrix completion technique [13] which works on the animation matrix \( \mathbf{A} \in \mathbb{R}^{3k \times n} \). This technique represents the geometry-agnostic approach for the consolidation of \( \mathbf{A} \), where no prior information is required. It belongs to the offline case, where all the available data has already been captured.
- **GMC**: An extended technique of MC [13] which exploits the spatial static point cloud geometry, represented by the Laplacian matrix \( \mathbf{L} \). It also belongs to the offline case.

Fig. 6 presents the NMSVE per frame for each compared method. According to this figure, the proposed method provides better results than any other method. LMS and GMC always have a constant NMSVE value independently of the frame.

![Figure 6: NMSVE values per frame for different methods.](image)
specifically, the NMSVE value of the proposed method is $\approx -62 \text{ dB}$ when the temporal coherence is high. Theoretically, we can guarantee that if the sampling frequency of a captured TVPC is high then the consolidated result using our method does not have a difference that can be easily perceived, even in highly incomplete cases. In Fig. 5 (a), the consolidated results for the Handstand model are presented. The experiment is executed after assuming different numbers of known vertices (30% and 50%). For the sake of completeness, the NMSVE values of the consolidated 3D static point cloud are also shown. In Fig. 5 (b), a heatmap visualization is finally presented highlighting the difference between the original and the consolidated 3D static point cloud for each one of the considered approaches.

4. CONCLUSIONS

In this work, we have provided a temporally coherent TVPC completion approach based on a novel weighted regularized Laplacian interpolation method. The proposed approach efficiently exploits the coherences between motion vectors of known vertices and is ideally suited for online settings where the point cloud sequence is not known a priori and is dynamically generated. An extensive evaluation study using a collection of highly incomplete TVPCs verified the effectiveness of our approach in terms of both reconstruction quality and computational complexity.

5. REFERENCES


