We introduce a new incremental 2-manifold surface reconstruction method. Compared to the previous works, its input is a sparse 3D point cloud estimated by a Structure-from-Motion (SfM) algorithm instead of a more common dense input. The main argument against such a method is that the lack of points implies an inaccurate scene surface. However, the advantages like point quality (thanks to the SfM machinery including bundle adjustment and interest point detection) and simplified resulting surface makes it worth of exploration.

Our algorithm is incremental since the surface is locally updated for every new camera pose (and its 3D points) estimated by SfM. This is an advantage compared to global methods like [5] or [2] for applications which require a surface while reading the video sequence. Compared to [6], our method avoids prohibitive time complexity in presence of loops in the camera trajectory. Last but not least, unlike other incremental methods like [3] the output surface is a 2-manifold, i.e. it is a list of triangles in 3D such that the neighborhood of every surface point is topologically a disk. This property is needed to define surface normal and curvature [1] and thus is used by many mesh processing and computational geometry algorithms.

Now we introduce notations. Let \( P \) be a set of 3D points on the unknown scene surface. The 3D Delaunay triangulation of \( P \) is a list \( T \) of tetrahedra which partition the convex hull of \( P \). A list \( O \) of tetrahedra \( (O \subseteq T) \) represents the reconstructed object whose volume is \( |O| \), the union of the \( O \) tetrahedra. Border \( \partial O \) is the list of triangles (tetrahedra faces) which are included in exactly one tetrahedra of \( O \). The union of triangles \( |\partial O| \) is our target surface and should be a 2-manifold. SfM also provides visibility knowledge \( R_v \): every point \( p \in P \) is computed from camera locations \( c_j \) where \( j \in R_v \). This implies that \( |\partial O| \) should not intersect the rays (line segments) \( c_j p \). \( j \in R_v \) except at \( p \). The tetrahedra intersected by a ray are labeled free-space, the others are matter. Let \( F \) be the set of free-space tetrahedra. In practice, \( |\partial F| \) is not a 2-manifold.

One iteration of our algorithm is shown in Fig. 1. At image (or time) \( t+1 \), we have the following input

- 3D Delaunay triangulation \( T_t \);
- list \( F_t \) of freespace tetrahedra such that \( F_t \subseteq T_t \);
- list \( O_t \) such that \( O_t \subseteq F_t \) and \( |\partial O_t| \) is manifold;
- list \( P_{t+1} \) of new SfM points, camera locations \( c_j \), where \( t' \leq t + 1 \).

and output

- 3D Delaunay triangulation \( T_{t+1} \) by adding \( P_{t+1} \) in \( T_{t+1} \);
- list \( F_{t+1} \) of freespace tetrahedra (by raytracing in \( T_{t+1} \)) such that \( F_{t+1} \subseteq T_{t+1} \);
- list \( O_{t+1} \) such that \( |\partial O_{t+1}| \) is manifold and \( O_{t+1} \) is largest as possible in \( P_{t+1} \).

Moreover, three assumptions are introduced for complexity reasons. First, \( p_j \in P_i \) and \( t' \in R_t \) imply \( t \in R_t \) and \( t' \leq t \) (SfM does not update a reconstructed point once it is a vertex of \( T \)). Second, the length of all rays \( c_j p_i \) is bounded by \( r > 0 \) (a consequence of standard SfM point filtering). Third, the diameters of the tetrahedra are bounded by \( l > 0 \) thanks to a large Cartesian grid of Steiner (extra) vertices. Then the addition of \( P_{t+1} \) is a local update of \( T \) included in ball \( B_{r+1} \) centered at \( c_i \) with radius \( r + l \).

Assume that we add \( P_{t+1} \) in the Delaunay \( T_t \) as soon as the \( O_t \) computation is done. This destroys a list \( D \) of tetrahedra, which can contain tetrahedra in \( O_t \). In Fig. 1f the problem is the following: if we initialize \( O_{t+1} = O_t \backslash D \), \( |\partial O_{t+1}| \) can be non manifold, and there is no obvious method to update \( O_{t+1} \) such that \( |\partial O_{t+1}| \) becomes manifold again.

Our idea is to compute \( D \) without updating \( T \) to ensure that \( |\partial O_{t+1}| \) is always a manifold. In practice, the results are better if we replace \( D \) by list \( E \) of the tetrahedra included in \( B_{r+1} \) (then \( D \subseteq E \subseteq T_t \)). We initialize \( O_{t+1} = O_t \) (Fig. 1a) and we progressively remove tetrahedra from \( O_{t+1} \) such that \( |\partial O_{t+1}| \) is maintained manifold. This shrinking process is stopped if \( O_{t+1} \cap E = \emptyset \) (Fig. 1b) or if \( O_{t+1} \) can not decrease.

To add the points \( P_{t+1} \) to \( T_t \), we initialize \( T_{t+1} = T_t \) and \( F_{t+1} = F_t \). For every point \( p \in P_{t+1} \), we compute the list \( D(p) \) of tetrahedra in \( T_{t+1} \) which would be destroyed if we add \( p \) to \( T_{t+1} \). Point \( p \) is added to \( T_{t+1} \) and \( D(p) \) is removed from \( F_{t+1} \), if and only if \( D(p) \cap O_{t+1} = \emptyset \) (Fig. 1c). Since \( O_{t+1} \) is unchanged, \( |\partial O_{t+1}| \) is still a 2-manifold.

Finally, \( O_{t+1} \) progressively grows in freespace \( F_{t+1} \) by adding tetrahedra such that \( |\partial O_{t+1}| \) is maintained manifold (Fig. 1c). As the finishing touch, incremental post-processing (handle removal and Laplacian smoothing) is performed on \( O_{t+1} \) to improve the output surface quality.

We experiment on a large scale real urban scene using the Ladybug camera, as well as on synthetic images with ground truth. The trajectory of the real sequence is 2.5 km long and includes a large loop (2.3 km). 483k points are reconstructed in [4] in 1306 keyframes, then the loop is closed thanks to a global bundle adjustment. The final surface has 528k triangles (http://www.youtube.com/watch?v=w1AQfvhGx5I or joint video). The synthetic sequence has 1553 images generated by ray-tracing with textures extracted from real images taken in a city. The trajectory is 621 m long and is a closed loop. 145k points are reconstructed in 346 keyframes. The estimated error between the final surface (182k triangles) and the ground truth is 0.76 m for 70% of the pixels.

Several improvements of the described method are subjects of future works. First, the choice of a better (smaller) enclosing area \( A \) would decrease the computation time. Second, our implementation and matching method should be improved. Last, several lines of investigation exist to enhance the quality of the reconstructed surface (curve reconstruction and integration in the Delaunay triangulation, surface denoising methods that make a better usage of the SfM properties and scene priors).