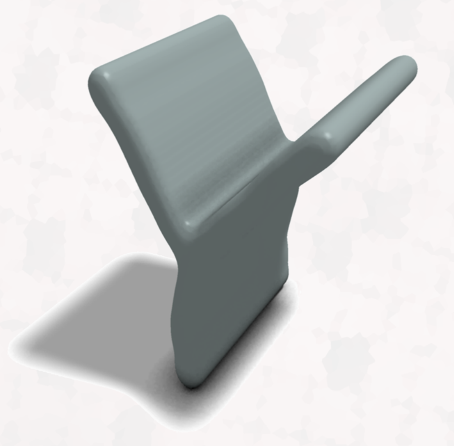
Surface intersection detection with topological guarantees

John Chaussard
Michel Couprie
ESIEE Paris, Departement Informatique

1. Objective of this work

Our goal is to **decompose a surfacic object into surface patches**.

A 3D object is considered as surfacic if one of its dimensions can be regarded as negligible with regard to the two other ones. However, when considering objects such as cracks in concrete or foam walls, we have to deal with (discrete) 3D objects which have a certain "thickness" (as the example on the right). In order to analyse such objects as compounds of surface patches, we first have to **perform a thinning step which must preserve essential topological and geometrical characteristics, and produce a thin object**.

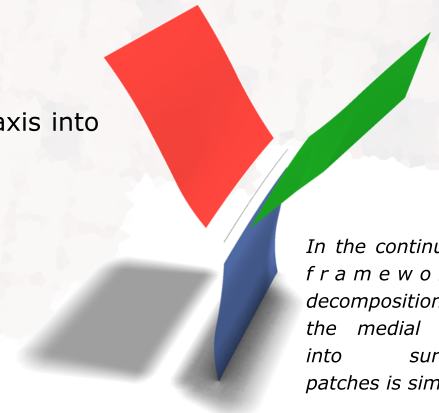


Surfacic objects may have a certain "thickness"

In the continuous framework, the medial axis of an object is thin.

In the continuous framework, the **medial axis** (the set of the centers of the object's maximal balls) can be considered as the result we were looking for: it is thin, well centered in the object, and has the same topology than the original object (example on the left).

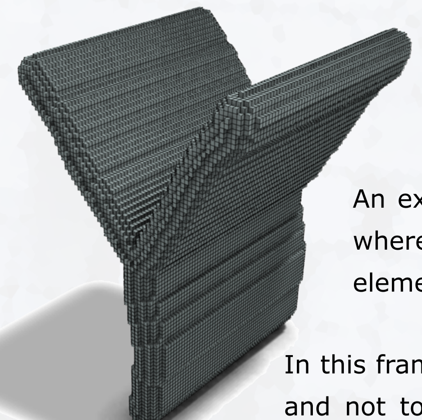
It is then possible to decompose the medial axis into surface patches (see on the right).



In the continuous framework, decomposition of the medial axis into surface patches is simple.

However, homotopic thinning and decomposition in the discrete framework is not so easy.

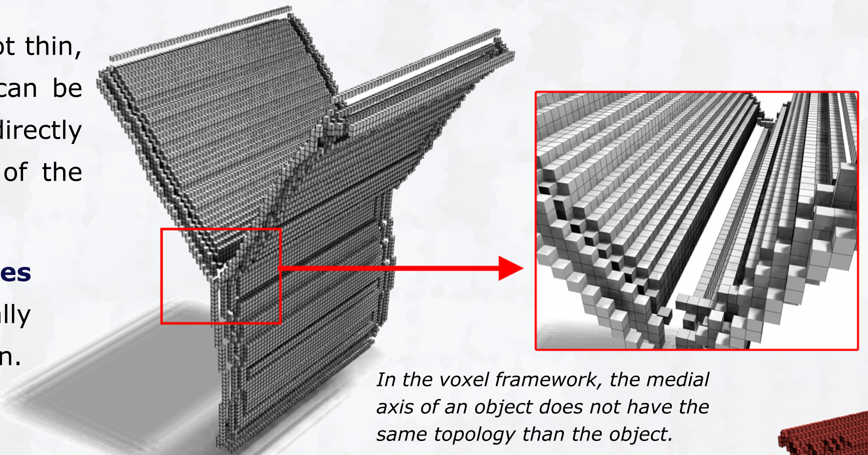
2. The voxel framework



An object made of voxels

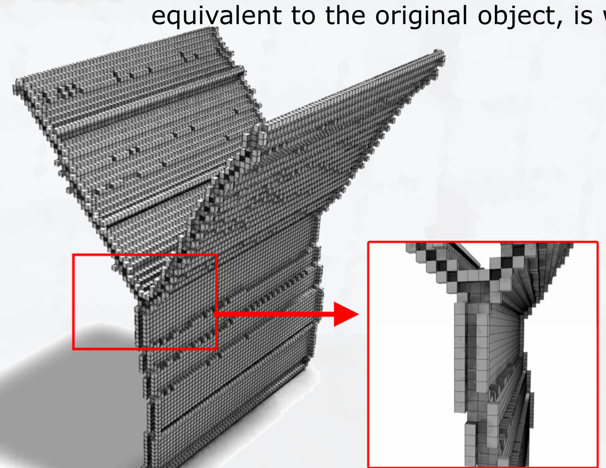
An example of discrete framework is the **voxel framework**, where every point of the object is represented by an elementary cube called **voxel** (example on the left).

In this framework, the medial axis of the object is not thin, and not topologically equivalent to the object (as can be seen on the example on the right). We cannot use directly the medial axis as a result of homotopic thinning of the object.



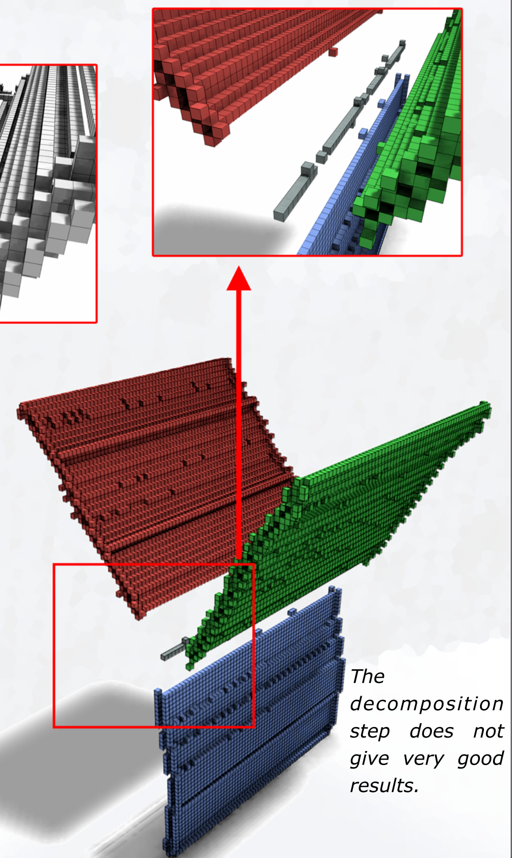
In the voxel framework, the medial axis of an object does not have the same topology than the object.

Instead, we perform a **skeletonisation of the object that preserves all points of the medial axis**. The result (down left) is topologically equivalent to the original object, is well centered, but is not thin.



In the voxel framework, the skeleton is not always thin.

In order to decompose the result into surface patches, we label each voxels of the object as a surface voxel, or as an intersection between surfaces voxel. **However, as previously said, the result of the skeletonisation is not thin**. Consequently, the **decomposition of the object into surface patches will lead to some errors** (some voxels may not be labeled as part of surfaces but as part of volumes). Moreover, it is very difficult to characterize the voxels which are part of intersections between surfaces: for example, until now, there is no method that guarantees that the voxels part of intersection between surfaces form a curve.

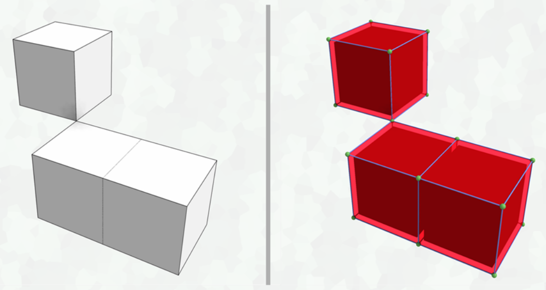


The decomposition step does not give very good results.

3. The cubical complex framework

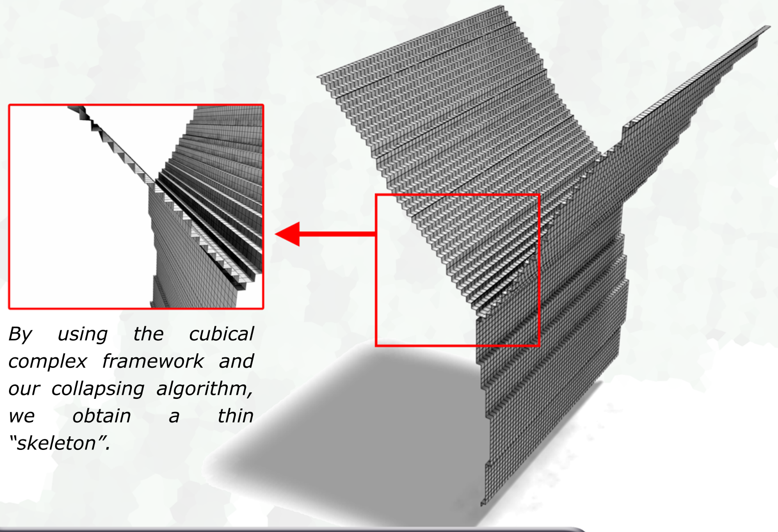
Another discrete framework is the **cubical complex framework**, where the objects are composed of elements of various dimensions: points (0-faces), segments (1-faces), squares (2-faces) and cubes (3-faces), glued together according to certain rules (example on the right). The fact that objects are made of elements of different dimensions will be very helpful to solve our problem.

In the cubical complex framework, homotopic thinning can be performed using the **collapse operation**. In an object X, when a face g of dimension m is contained inside only one face f of dimension m+1, we say that the pair (f,g) is free for X. Removing iteratively the free pairs of an object allow to perform collapsing in the cubical complex framework.



In the voxel framework, objects are made of voxels (on the left), whereas in the cubical complex framework (on the right), objects are made of points (green), segments (blue), squares (red) and cubes (black).

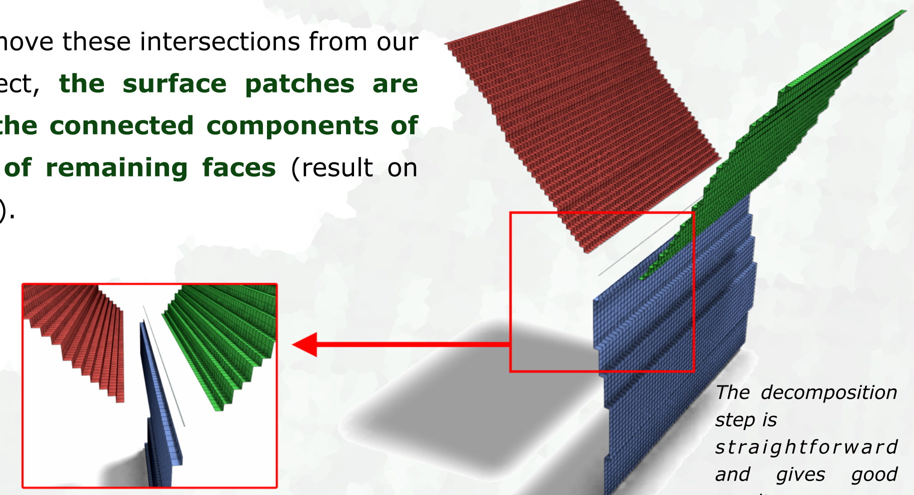
We have developed a **new collapsing algorithm** which looks at the dimension of the free pairs of an object, and at their orientation in space, in order to **produce a thin object**. Even if the result is not "as well centered in the object" than the medial axis, it contains the main geometrical information of the original object (example under).



By using the cubical complex framework and our collapsing algorithm, we obtain a thin "skeleton".

In the cubical complex framework, **decomposition of a thin object into surface patches is straightforward**. First, we consider all the segments of the object which are included in three or more faces: these segments allow us to extract the **intersections between the surface patches**. We have the property that the **resulting intersections are one-dimensional objects** (set of curves).

If we remove these intersections from our thin object, **the surface patches are simply the connected components of the set of remaining faces** (result on the right).

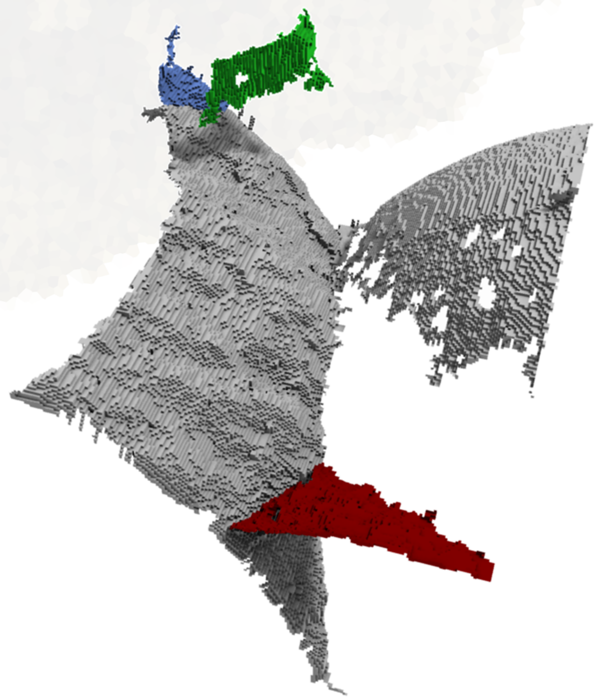


The decomposition step is straightforward and gives good results.

4. Results

We now present some results of our algorithm on "real life" images (cracks in concrete).

A "thick" surfacic object, representing cracks in concrete.



After homotopic thinning (with collapsing algorithm) and after removing isolated connected components, we obtain a decomposition of the object into surface patches.