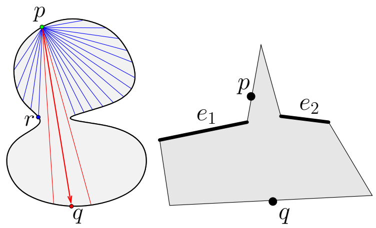
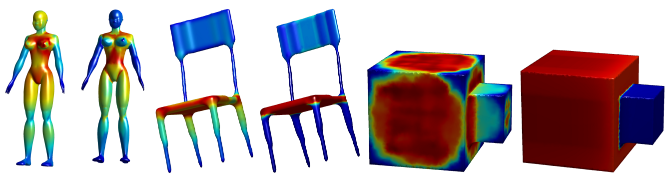
Continuous Visibility Feature

Guilin Liu, Yotam Gingold, Jyh-Ming Lien
Department of Computer Science, George Mason University



(a) (b)

Figure 1: (a) Example of 2D continuous visibility. The vertex r is continuously visible from the vertex p but the vertex q is not. (b) While p is continuously visible from q, q is not continuously visible from p.



(a) SDF (b) CVF (c) SDF (d) CVF (e) SDF (f) CVF

Figure 2: Examples of SDF and CVF on the same models. Red and blue indicate the largest and smallest SDF and CVF values, respectively. From these examples, we can see that CVF provides a better metric to distinguish visually different parts of a model.

Feature extraction often serves as a fundamental engineering task for higher level applications. For example, almost all of the recognition tasks start with defining or learning features. Visibility among points inside a given shape has been used directly or indirectly as shape features, in particular for the task of semantic shape segmentation. The intuition behind most of these visibility-based features is from the observation that two points sampled from the same semantic part tend to be visible from each other. For example, shape diameter function (SDF) [2] is a descriptor to describe the thickness of a mesh defined via local visibility. We believe that the traditional line-of-sight visibility used in all existing visibility-based shape features is insufficient.

In this paper, we will describe a new feature named *continuous visibility feature*. The feature provides stronger visibility measure by considering the continuously visible region for a vertex or facet. Thus this feature is defined in a per-vertex manner. We say that a point q on the mesh is continuously visible by p if there exists a geodesic path connecting p and q that is entirely visible by p. Note that, this path may not be the shortest geodesic path between p and q. More specifically, we define the function $\textsc{cv}(p,q)$ that returns **TRUE** if and only if

$$\exists \pi \text{ such that } \forall r \in \pi, \overline{pr} \cap M = \emptyset \,,$$

where π is a geodesic path connecting p to q on mesh M. An illustration of the continuous visibility is shown in Figure 1 (a). Comparisons with SDF are shown in Fig 2.

This first approach uses the property that the continuous visible region of the vertex v must be continuous because, by definition, every vertex in the continuous visible region is connected to v via a path that is entirely visible by v. In this Breadth First Search (BFS) based method, we iteratively expand the search only at the vertices that are visible to v and stop at the vertices that are invisible to v.

The second approach is to filter out the faces that guaranteed to be outside the continuously visible region of a given vertex v. If a triangle f and v form a tetrahedron of positive volume, then, we say that f is a positive triangle. Then a *continuously positive region* of v is a set T_v of positive triangles continuously connected to v. It is clear that v's continuously visible region must be a subset of T_v. Algorithm 1 sketches the overall idea of this filter-based approach.

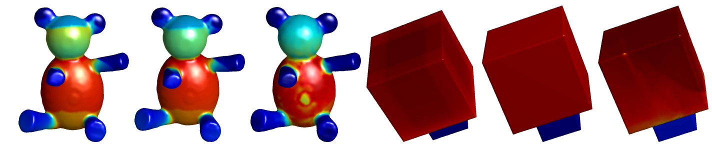
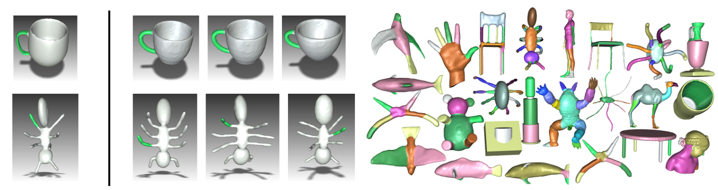


Figure 3: Results obtained from CVF (left) vs. strong CVF (middle) vs. weak CVF (right).



(a) Retrieval (b) Segmentation

Figure 4: Application in part-base retrieval and shape segmentation

Data: a vertex v and its positive continuous region T_v
Result: the continuous visible region of v
while $\exists f \in \partial T_v$ *whose visibility to v is undetermined* **do**
  Let $u \notin T_v$ be a vertex incident to f ;
  Find a path π connecting v and u ;
  Starting from v, let $e = \{w', w\}$ be the first edge such that w' is invisible and w is visible ;
  Starting with e, find a closed visible/invisible boundary loop L containing e
  Mark all vertices connected to u without crossing L be continuously invisible from v
end

Algorithm 1: Filter-based CVF computation

If we use the filter-based approach on the same set of models, the intersection tests and running time reduce by around 80% and 40%.

We also proposed some variants of CVF including *strong continuous visibility* and *weak continuous visibility*. We say that two points p and q have *strong continuous visibility* if both $\textsc{cv}(p,q)$ and $\textsc{cv}(q,p)$. For example, in Fig 1(b), p and q don't have strong continuous visibility. We propose that another measure of *weak continuous visibility* $\textsc{wcv}$. Given two points p and q, $\{\textsc{wcv}(p,q) = \textbf{TRUE}\}$ if p and q are visible to each other but the length of the invisible part of the shortest geodesic path connecting p and q is smaller than a user defined value τ. Let π be the the shortest geodesic path connecting p and q and let $\bar{\pi}_w$ and $\bar{\pi}_v$ be subset of π invisible to v and w, respectively. The function $\textsc{wcv}(p,q)$ returns **TRUE** if $\max(||\bar{\pi}_w||, ||\bar{\pi}_v||) \leq \tau$. Results of two variants are shown in Fig 3.

We show the application of CVF in part-aware shape retrieval, shown in Fig 4(a) and in segmentation, shown in Fig 4(b) and evaluated in Princeton Shape Segmentation Benchmark [1].

[1] X. Chen, A. Golovinskiy, and T. Funkhouser. A benchmark for 3d mesh segmentation. *ACM Transactions on Graphics (TOG)*, 28(3):73, 2009.

[2] L. Shapira, A. Shamir, and D. Cohen-Or. Consistent mesh partitioning and skeletonisation using the shape diameter function. *The Visual Computer*, 24(4):249–259, 2008.

This is an extended abstract. The full paper is available at the Computer Vision Foundation webpage.