



Title:

A Predictor-Corrector Type Incremental Algorithm for Weighted Straight Skeletons

Authors:

Barış İrhan, barisirhan@gmail.com, Arkitech Advanced Construction Technologies

Keywords:

Weighted Straight Skeleton, Predictor-Corrector, Incremental, Roof Modeling, Computational Solid Mechanics

DOI: 10.14733/cadconfP.2021.56-60

Introduction:

In classical straight skeleton problem, each edge of the polygon moves inwards with a constant unit velocity in such a way that it remains essentially parallel to itself throughout the course of the shrinking process. The regions swept by the polygon edges construct the faces of the straight skeleton structure of the underlying polygon. During the shrinking process, polygon topology might change due to edge collapse and edge split events. The edge collapse event takes place at an instance when at least one edge of the polygon collapses down to a point, whereas the edge split event comes into play as at least one of the vertices hits an edge or another vertex of the polygon.

The existing algorithms for the construction of straight skeletons are based on wave propagation and they make use of the additional tools like kinetic triangulation [1] or motorcycle graph [2]. The weighted version of the straight skeleton algorithm was pioneered in [3].

Contribution:

To the author knowledge, a predictor-corrector type incremental algorithm has never been used before for the construction of straight skeletons. In this work an algorithm of this class is presented. In addition to being original, the algorithm enjoys some very powerful features. First of all, it is simple, intuitive and easy to implement. It is general that it can be applied to any kind of closed planar polygons of arbitrary complexity equally well. There is no room for exceptional cases. Algorithm works with raw input provided by the underlying polygon. Construction of a kinetic triangulation or a motorcycle graph is not necessary.

The Algorithm:

Consider a closed planar polygon whose vertices change position in time with a prescribed constant velocity. Recall that velocity of the vertices has a particular form in case of straight skeleton problem such that an edge belonging to the polygon remains essentially parallel to itself during incremental motion. Then for a given time increment

$$\Delta t_n = t_{n+1} - t_n \quad (2.1)$$

the position of a vertex at time t_{n+1} can exactly be computed by direct integration as

$$\mathbf{P}_{n+1} = \mathbf{P}_n + \Delta t_n \mathbf{v}_P \quad (2.2)$$

While the polygon is shrinking, there are two geometrical constraints to be respected. First constraint states that polygon edges cannot swap during incremental motion. If edge swap happens, it indicates that there are edges which collapse within the given time increment. The second constraint is the so-called edge impenetrability which states that during incremental motion polygon cannot penetrate into itself. If such a penetration occurs, it indicates that there are edges which split within the given time increment. Position of the vertices computed by direct time integration is then used as a predictor if edge collapse or edge split events take place during incremental motion. In such a case, predicted positions must be corrected. Correction phase requires the determination of the exact time instances of the edge collapse and edge split events.

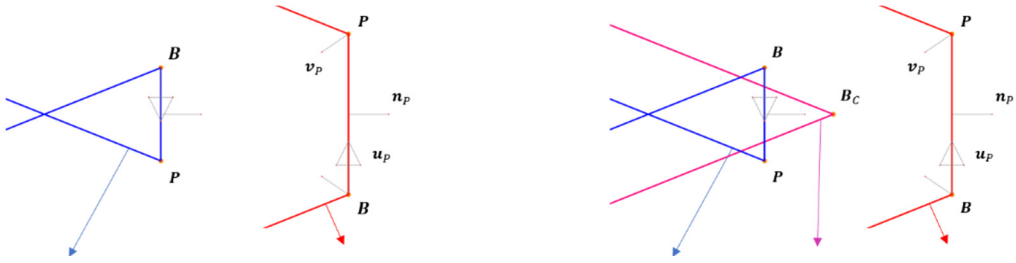


Fig. 1: Edge swap (left). The instance of edge collapse (right)

Edge Collapse Event:

Edge collapse event within a time increment is detected by edge swap. An edge is swapped if it changes direction (see Figure 1). Since vertices move along a linear path with a constant velocity, the instance of an edge collapse event can exactly be computed by linear interpolation.

Projected lengths of a given vertex P corresponding to time steps t_n and t_{n+1} are given, respectively, by

$$l_n = (\mathbf{P}_n - \mathbf{B}_n) \cdot \mathbf{u}_P \quad (2.3)$$

and

$$l_{n+1} = (\mathbf{P}_{n+1} - \mathbf{B}_{n+1}) \cdot \mathbf{u}_P \quad (2.4)$$

Then one can parameterize projected length within a time increment as

$$l = \frac{1}{2}(1 - \xi)l_n + \frac{1}{2}(1 + \xi)l_{n+1} \quad (2.5)$$

with $\xi \in [-1, 1]$.

At the instance of edge collapse event the following equation must hold:

$$l = 0 \quad (2.6)$$

Inserting Eq. (2.5) into Eq. (2.6), local coordinate corresponding to edge collapse event is then computed as

$$\xi_C = -\frac{l_M}{l_D} \quad (2.7)$$

where l_M and l_D are defined as

$$\begin{aligned} l_M &= \frac{1}{2}(l_{n+1} + l_n) \\ l_D &= \frac{1}{2}(l_{n+1} - l_n) \end{aligned} \quad (2.8)$$

One can also parameterize the time increment, similar to Eq. (2.5), as

$$\Delta t = \frac{1}{2}(1 - \xi)0 + \frac{1}{2}(1 + \xi)\Delta t_n \quad (2.9)$$

By inserting ξ_C computed in Eq. (2.7), time increment corresponding to edge collapse event, see Figure 1, is obtained as

$$\Delta t_C = \frac{1}{2}(1 + \xi_C)\Delta t_n \quad (2.10)$$

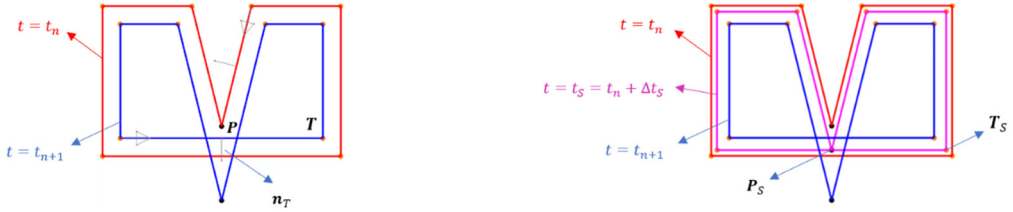


Fig. 2: Edge penetration (left). The instance of edge split event (right)

Edge Split Event:

Edge split event within a time increment is detected by edge penetration (see Figure 2). Since the vertices move along a linear path with constant velocities, exact instance of edge split event can again be computed by using linear interpolation. For this purpose, we define parameterization of vertices P and T as

$$\begin{aligned} P &= \frac{1}{2}(1 - \xi)P_n + \frac{1}{2}(1 + \xi)P_{n+1} \\ T &= \frac{1}{2}(1 - \xi)T_n + \frac{1}{2}(1 + \xi)T_{n+1} \end{aligned} \quad (2.11)$$

At the instance of edge split event, vertex P must be on the edge belonging to vertex T (see Figure 2). Therefore, the following relation must hold:

$$(\mathbf{P}_S - \mathbf{T}_S) \cdot \mathbf{n}_T = 0 \quad (2.12)$$

Inserting Eq. (2.11) into Eq. (2.12), local coordinate corresponding to edge split event is computed as

$$\xi_S = -\frac{(\mathbf{P}_M - \mathbf{T}_M) \cdot \mathbf{n}_T}{(\mathbf{P}_D - \mathbf{T}_D) \cdot \mathbf{n}_T} \quad (2.13)$$

where \mathbf{P}_M and \mathbf{P}_D are defined as

$$\begin{aligned} \mathbf{P}_M &= \frac{1}{2}(\mathbf{P}_{n+1} + \mathbf{P}_n) \\ \mathbf{P}_D &= \frac{1}{2}(\mathbf{P}_{n+1} - \mathbf{P}_n) \end{aligned} \quad (2.14)$$

Definitions for \mathbf{T}_M and \mathbf{T}_D follows from Eq. (2.14). Using the equation for the parameterization of the time increment, Eq. (2.9), time increment corresponding to edge split event is obtained as

$$\Delta t_S = \frac{1}{2}(1 + \xi_S)\Delta t_n \quad (2.15)$$

Predictor-Corrector Algorithm:

For a given time increment, predicted position of a typical vertex P is computed by direct integration as

$$\mathbf{P}_{n+1}^{pred} = \mathbf{P}_n + \Delta t_n \mathbf{v}_P \quad (2.16)$$

After predicted positions are computed, admissibility of the polygon is checked with respect to edge collapse and edge split events using Eq. (2.6) and Eq. (2.12). If there is no event taking place, final positions are equal to predicted positions, i.e.,

$$\mathbf{P}_{n+1} = \mathbf{P}_{n+1}^{pred} \quad (2.17)$$

If there is an edge collapse event within the time increment, predicted positions are corrected as

$$\mathbf{P}_{n+1} = \mathbf{P}_{n+1}^{pred} + \Delta \mathbf{P}_C \quad (2.18)$$

with

$$\Delta \mathbf{P}_C = (\Delta t_C - \Delta t_n) \mathbf{v}_P \quad (2.19)$$

Similarly, if there is an edge split event within the time increment, predicted positions are corrected as

$$\mathbf{P}_{n+1} = \mathbf{P}_{n+1}^{pred} + \Delta \mathbf{P}_S \quad (2.20)$$

with

$$\Delta \mathbf{P}_S = (\Delta t_S - \Delta t_n) \mathbf{v}_P \quad (2.21)$$

If both edge collapse and edge split events take place within the same increment, the one which happens earlier is considered and the other one is disregarded.

Roof Modeling:

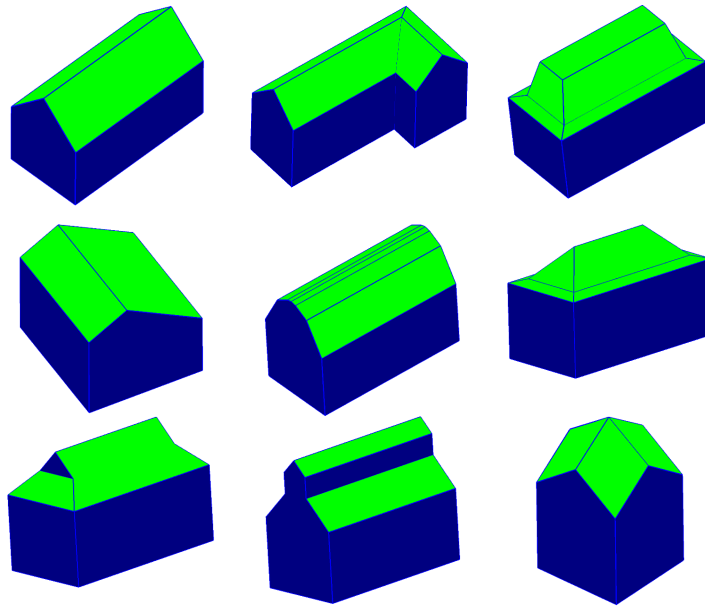


Fig. 3: Different roof types

In the roof construction using weighted straight skeleton algorithm, new position of the polygon is computed based on predictor-corrector algorithm for a prescribed offset in z direction. A roof is complete if there remains no possibility to further apply an offset. Offset polygons are kept and used to construct the roof polygons, namely the straight skeleton structure, in a systematic way by paying special attention to the vertices which are split, collapsed or colinear. In Figure 3, several representative non-exhaustive examples are given.

Conclusions:

A predictor-corrector type incremental algorithm for the construction of weighted straight skeletons of 2D general planar polygons, possibly with holes, has been developed. The algorithm works with the raw input provided by the underlying polygon. Additional tools like kinetic triangulation or motorcycle graphs are not needed. Polygon edges which move at different speeds and/or start to move at different times are supported by default.

References:

- [1] Aichholzer, O; Aurenhammer, F.: Straight skeletons for general polygonal figures in the plane. In: International computing and combinatorics conference. 1996, p. 117–126. https://doi.org/10.1007/3-540-61332-3_144
- [2] Cheng, S.-W., Vigneron, A.: Motorcycle graphs and straight skeletons. In: Proceedings of the thirteenth annual ACM-SIAM symposium on Discrete algorithms. Society for Industrial and Applied Mathematics, 2002, p. 156–165.
- [3] Eppstein, D., Erickson, J.: Raising roofs, crashing cycles, and playing pool: Applications of a data structure for finding pairwise interactions. *Discrete & Computational Geometry*, 22 (4), 1999, p. 569–592. <https://doi.org/10.1007/PL00009479>