Journal of King Saud University – Computer and Information Sciences 34 (2022) 1240–1245

Contents lists available at ScienceDirect

Journal of King Saud University – Computer and Information Sciences

journal homepage: www.sciencedirect.com

A novel algorithm for a continuous and fast 3D projection of points on triangulated surfaces for CAM/CAD/CAE applications

L. Orazi, B. Reggiani *

DISMI Department of Sciences and Methods for Engineering, University of Modena and Reggio Emilia, Via Amendola 2, 42122 Reggio Emilia, Italy

ARTICLE INFO

Article history:
Received 25 February 2020
Revised 13 May 2020
Accepted 7 June 2020
Available online 11 June 2020

Keywords:
Projection method
Triangle mesh
Vertex normals
Parameterization
Bilinear projection
Continuous projection

ABSTRACT

In the present work, a novel algorithm for the continuous projection of point triangles belonging to a triangle mesh is presented. The algorithm uses the normals defined at the vertices of the triangle to perform the projection. The direction of projection is not chosen *a priori* but depends on the point to be projected: the proposed algorithm, named Fast Continuous Projection method (FCP) lets the projection direction vary continuously on the mesh. Moreover, the direction is coherent with the original surface that is approximated with the triangle mesh. An optimized version of the algorithm is also presented: this uses pre-evaluated matrices to reduce the calculation time. This algorithm can be effectively used when a large set of points has to be projected on a coarse mesh as, for example, to generate the scanning vectors for laser engraving/milling.

1. Introduction

Between the existing methods of 3D surface representation and modelling, triangle meshes and NURBS (Non Uniform Rational Basis-Splines) remain the most widely used (Pottmann et al., 2015) in case, respectively, of unstructured and structured parametric representations. However, while NURBS can accurately and continuously represent multiform 3D objects but at a cost of higher mathematical complexity (Chang and Chen, 2011), triangle meshes are discontinuous but allow fast calculations, in particular in cases like projections, intersections between object, path calculations, etc... (Guo et al., 2019). There are situations where continuity and fast calculations are both required. An example is offered in CAE (Computer Aided Engineering) applications when objects are discretized by hexahedral or tetrahedral meshes resulting in quadrangular bilinear facets or triangular ones. The continuous projection on the first type of meshes has been investigated and partially solved by (Ramsey et al., 2004) and this algorithm was implemented with some modifications as example in laser hardening of a part (Tani et al., 2007), where solution of a 3D heat-flow equation on a discrete mesh through Finite Element Method (FEM) or Finite Volume Method (FVM) solution scheme is also required. A detailed description of normal congruence and of several operators for discrete mesh geometry can be found in (Sun et al., 2016) where authors present examples of a line-congruent mapping between triangles nearly parallel one each other. Another method to map 3D features from a texture space to a mesh space is presented in (Porumbescu et al., 2005). In this work, triangles in texture space are extruded generating prisms that are subsequently divided in rectangular tetrahedra used to map points from texture space to mesh space. The method can be useful employed to distribute a 3D object over a mesh, but it requires a very complicated topological management of the generated prisms. In the case of the universally used triangle meshes, except for the basic geometric algorithms in (O'Rourke, 1998), to the best of the authors' knowledge, nothing has been proposed in literature about a continuous projection method that can be needed, as example, in CAM for the generation of scanning vectors for laser engraving of a free-form part (Schmidt et al., 2018). During the treatment of the part, the laser head is placed at a given position and then, a specific area of the part is treated from that position generating 2D laser scanning vectors at different layers before passing to the next head placement. Thus, given the discontinuous nature of the laser head placements, a coarse triangle mesh

* Corresponding author at: DISMI - Department of Sciences and Methods for Engineering, University of Modena and Reggio Emilia, Viale Amendola 2, 42122 Reggio Emilia, Italy.
E-mail address: barbara.reggiani@unimore.it (B. Reggiani).

Peer review under responsibility of King Saud University.

Production and hosting by Elsevier

https://doi.org/10.1016/j.jksuci.2020.06.005
1319-1578/

is used by the CAM to represent the different triangular patches to be scanned in each placement (Cuccolini et al., 2013). Moreover, the part to be engraved/milled is already treated by the CAM as a fine triangle mesh characterized by a complex topography. In this condition, a large set of points have to be continuously projected in space onto placement triangles in order to delineate the laser path, thus stating the requirement to have a fast algorithm to project points on triangles belonging to a triangle mesh. Furthermore, continuity of the variation of the projection direction is required to have laser scanning vectors consistent with the real part curvature, that is only approximated by the mesh. To properly process the part, the laser ray must hits the work surface within a maximum angle from the local normal since abrupt changes of the angle result in an aesthetical degradation of the marking quality. A representative scheme of the proposed method is reported in Fig. 1, adapted from (Cuccolini et al., 2013), that represents the section of a workpiece to be laser engraved. In the Figure, the continuous black line is the trace of the workpiece, the blue line is the fine mesh containing the points $\boldsymbol{P}$ to be projected, the dashed line is the trace of a triangle T of the coarse mesh approximating the workpiece, $\boldsymbol{n}_i$ are the vertex normals and $\boldsymbol{n}_T$ is the normal of the triangle$T$. The green vector connecting $\boldsymbol{P}$ to $\boldsymbol{P}_{\boldsymbol{proj}}$, generally not aligned to $\boldsymbol{n}_T$, is the requested projection. In laser processing applications, the number of points to be projected N_p is much higher than that of placement triangles N_t, the former coming from a fine mesh of the textured surface with an expected number of points between 10^5 and 10^7 and the latter by a coarse discretization of the workpiece with no more than $10^2\tilde{A}\cdot 10^3$ triangles, each corresponding to a scanning head placement. This and similar issues motivate for a fast-projection algorithm that ensures a continuous variation of the projection direction with respect to the position of the point to be projected without discontinuities between triangles. The proposed original algorithm, named FCP (Fast-Continuous-Projection) method, has been thought to fulfil these requirements by determining the existence of the projection and calculating the projected point with a procedure that guarantees the projection continuity between adjacent triangles, like the vertex normals interpolation in the Phong shading method (Phong, 1975). However, the efficiency of the proposed method is moreover increased when projecting a large set of points by the implementation of an original projection matrix for every triangle.

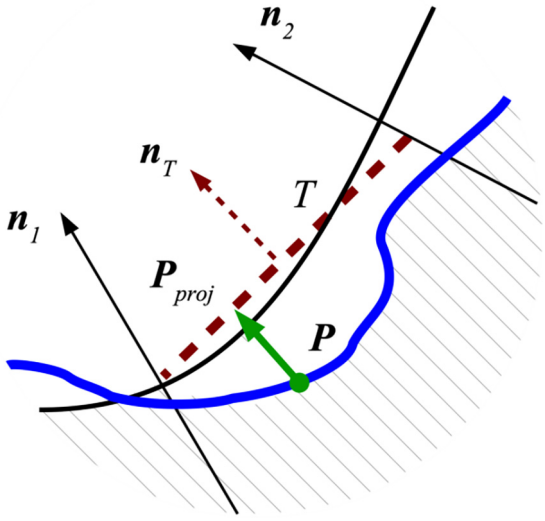

Fig. 1. Section of the workpiece to be laser engraved (black line: trace of the workpiece; blue line: fine mesh with the points **P** to be projected; dashed line: trace of a triangle T of the coarse mesh approximating the workpiece; $\mathbf{n}_i$: vertex normals; $\mathbf{n}_T$ normal of the triangle T; green vector: requested projection).

2. The FCP method

As stated in the introduction section, the novel FCP method uses the normal defined at the vertices of the mesh (Fig. 2 left). Usually, these are calculated and provided by the software used to manage triangle meshes, but if they are not available or somehow inconsistent with the real geometry, they can be calculated as proposed in (Hege and Polthier, 2011). The inputs are a mesh triangle T having vertices$\boldsymbol{V}_{ig}, i = 0,1,2$, and the Point $\boldsymbol{P}_g = [x_g y_g z_g]$ to be projected on triangle T (Fig. 2), where the subscript "g" stands for quantities expressed in the "Global frame of reference". The projection direction $\boldsymbol{d}$ is unknown, conversely to the usual standard projection or ray tracing.

2.1. Pre-processing

The problem is firstly converted into a two dimensional one, similarly as in (Jones, 1995) and (Baldwin and Weber, 2016), by considering T as lying on the xy plane. The simplest way to do this is to calculate the translation and rotation matrices to transform T so that $\boldsymbol{V}_0$ lies on the origin, $\boldsymbol{V}_1$ lies on the x-axis and $\boldsymbol{V}_2$lies on the xy plane. The resulting transformation matrix, the transformed vertices $\boldsymbol{V}_i = [V_{i0} V_{i1} V_{i2}]$ and the transformed vertex normals $\boldsymbol{n}_i = [n_{i0} n_{i1} n_{i2}]$ $(i = 0,1,2)$ can all be evaluated once for each triangle in a pre-processing step, step moreover mandatory in case of laser vectors generation: point $\boldsymbol{P}_g$ is transformed in $\boldsymbol{P} = [xyz]$, the projected point $\boldsymbol{P}_{proj}$ is calculated in the local frame and then it will be transformed back to the global frame of reference. The computational cost of this phase is $O(N_p + N_v)$, where N_v is the number of mesh vertices. Ideally, before applying the method, $\boldsymbol{P}$ is checked to be in the axis aligned bounding box representing each control volumes (Fig. 2 right) inside which each point is actually projected on T.

These semi-spaces are bounded by the three bilinear surfaces obtained connecting $\boldsymbol{n}_i$ and $\boldsymbol{n}_{i+1(\mathrm{mod}3)}$ truncated by the planes at$z = \pm z_{pmax}$. The value of z_{pmax} is appropriately selected and, for a practical computation of laser engraving vectors, it is limited by the depth of focus characterizing the optical system and it represents the maximum allowed distance between the workpiece and the mesh to be engraved. It then simply remains to check that $\boldsymbol{P}$ is inside this bounding box before proceeding further with calculations, with a $O(N_p)$ computational cost in the worst case.

The regularity requested by the mesh is that the three vertex normals of a given triangle are all oriented in the positive direction of the z axis, as shown in Fig. 3.

2.2. The FCP algorithm

The proposed method itself starts by constructing the affine triangle $T(z)$ lying on the plane π which contains the point $\boldsymbol{P}$ and is parallel to T (Fig. 3). The distance z comes from the roto-translation in the preprocessing phase. Each vertex $\boldsymbol{U}_i$ of $T(z)$ is obtained by intersecting the plane π with the line having the normal $\boldsymbol{n}_i$ as direction and passing through the original triangle vertex $\boldsymbol{V}_i$; this is synthetized by the following equation:

$$\boldsymbol{U}_i(z) = \boldsymbol{V}_i + \boldsymbol{n}_i r_i i = 0,1,2 \tag{1}$$

where r_i are the distances between the correspondent vertices $\boldsymbol{V}_i$ and $\boldsymbol{U}_i$.

It can be mentioned that the checking of the counter clockwise direction of vertices $\boldsymbol{U}_i$ guarantees the consistency of the following method and the uniqueness of the projection direction on the

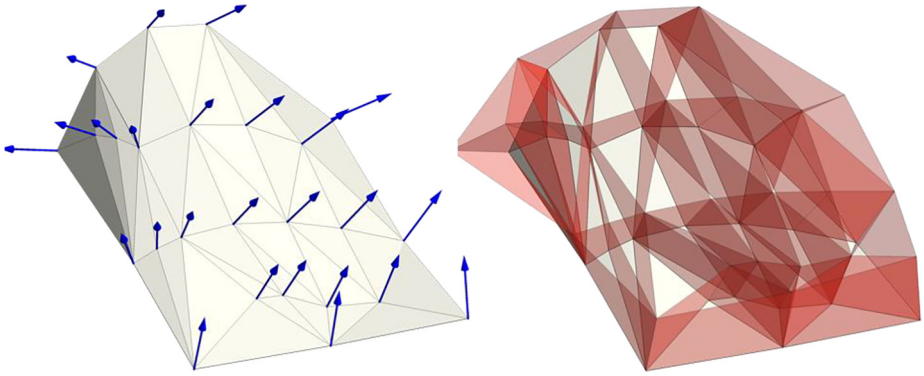

Fig. 2. On the left, the triangle mesh with its vertex normals shown in blue and, on the right, the correspondent control volumes.

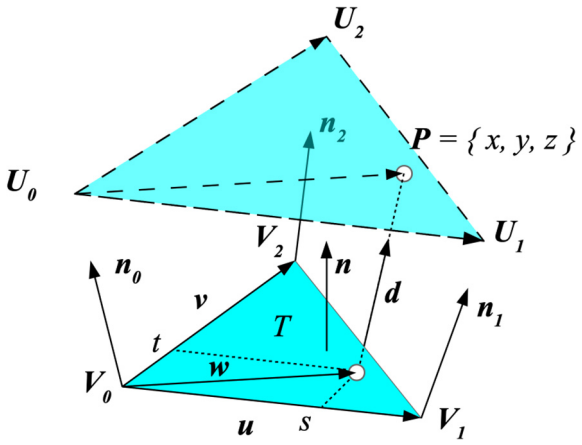

Fig. 3. Main quantities used by the proposed method.

given triangle in case of concave meshes. Instead, in case of mesh with convex zones, some vertex normals $\boldsymbol{n}_i$ could be oriented in the negative direction in z. It is supposed that the method can be applied only outside these conditions.

Subsequently, $\boldsymbol{u}(z)$, $\boldsymbol{v}(z)$ and $\boldsymbol{w}(z)$, parallel to the xy plane, are defined (Fig. 3):

$$\begin{aligned}\boldsymbol{u}(z) &= \boldsymbol{U}_1(z) - \boldsymbol{U}_0(z) = [u_{z0}\, u_{z1}\, 0]\\ \boldsymbol{v}(z) &= \boldsymbol{U}_2(z) - \boldsymbol{U}_0(z) = [v_{z0}\, v_{z1}\, 0]\\ \boldsymbol{w}(z) &= \boldsymbol{P} - \boldsymbol{U}_0(z) = [w_{z0}\, w_{z1}\, 0]\end{aligned} \tag{2}$$

Then, the parametric position (s,t) of $\boldsymbol{P}$ in $T(z)$ can be evaluated from:

$$w(z) = u(z)s + v(z)t\# \tag{3}$$

After a few calculations, the barycentric coordinates of the parametric position are obtained by:

$$\begin{aligned}s &= \frac{v_1 w_0 - v_0 w_1}{u_0 v_1 - u_1 v_0}\\ t &= \frac{u_0 w_1 - u_1 w_0}{u_0 v_{z1} - u_1 v_0}\#\end{aligned} \tag{4}$$

The point $\boldsymbol{P}$ is contained in the triangle $T(z)$ if:

$$\begin{cases} s \geq 0 \\ t \geq 0 \quad \# \\ s + t \leq 1 \end{cases} \tag{5}$$

It is then assumed that the parametric position (s,t) of $\boldsymbol{P}$ in $T(z)$ coincides with the parametric position of the searched projection $\boldsymbol{P}_{proj}$ on T. By defining (Fig. 3):

$$\begin{aligned}\boldsymbol{u} &= \boldsymbol{V}_1 - \boldsymbol{V}_0\\ \boldsymbol{v} &= \boldsymbol{V}_2 - \boldsymbol{V}_0 \quad \#\\ \boldsymbol{w} &= \boldsymbol{P}_{proj} - \boldsymbol{V}_0\end{aligned} \tag{6}$$

it is possible to calculate the required projection $\boldsymbol{P}_{proj}$ by:

$$\boldsymbol{P}_{proj} = \boldsymbol{V}_0 + us + vt\# \tag{7}$$

The direction of projection $\boldsymbol{d}$ is thus dependant on $\boldsymbol{P}$ in a bilinear way, i.e. this whole procedure is equivalent to projecting $\boldsymbol{P}$ on T with direction $\boldsymbol{d}$ given by:

$$d = \boldsymbol{n}_0 + (\boldsymbol{n}_1 - \boldsymbol{n}_0)s + (\boldsymbol{n}_2 - \boldsymbol{n}_0)t\# \tag{8}$$

The last expression shows that this method permits a continuous variation of the projection direction on the triangle parametric domain, as it was requested. Moreover, since in a triangle mesh the normals pertain to vertices and are thus shared by adjacent triangles, this continuity of the projection direction variation is implicitly ensured on the whole mesh given that self-intersections between parametric volumes are excluded.

2.3. Matrix form: Optimized FCP algorithm

A further optimization consists in rewriting the equations in matrix form and pre-calculating all the matrices that depend on the original triangle only and not on the point to project. This can greatly speed up the calculation when the number of points to be projected on each triangle is high. By substituting Eqn. (1) into Eqn. (2), $\boldsymbol{u}(z)$ becomes:

$$u(z) = \boldsymbol{V}_1 + \boldsymbol{n}_1 \tfrac{z}{n_{12}} - \boldsymbol{V}_0 - \boldsymbol{n}_0 \tfrac{z}{n_{02}} = \boldsymbol{V}_1 - \boldsymbol{V}_0 + \left(\tfrac{\boldsymbol{n}_1}{n_{12}} - \tfrac{\boldsymbol{n}_0}{n_{02}}\right) z\# \tag{9}$$

and similarly, for $\boldsymbol{v}(z)$ and $\boldsymbol{w}(z)$. Subsequently, let define:

$$\begin{aligned}\beta_{00} &= n_{00}/n_{02}\\ \beta_{01} &= n_{01}/n_{02}\\ \beta_{10} &= n_{10}/n_{12}\\ \beta_{11} &= n_{11}/n_{12}\\ \beta_{20} &= n_{20}/n_{22}\\ \beta_{21} &= n_{21}/n_{22}\end{aligned}\ \# \tag{10}$$

By definition, vectors$\boldsymbol{u}(z)$, $\boldsymbol{v}(z)$ and $\boldsymbol{w}(z)$ are parallel to the xy plane while the original triangle has been transformed in such a way that $\boldsymbol{V}_0$ lies on the origin, $\boldsymbol{V}_1$ lies on the x-axis and $\boldsymbol{V}_2$lies on the xy plane. Eqn. (9) can be rewritten as follows:

$$\begin{bmatrix} u_{z0} \\ u_{z1} \end{bmatrix} = \begin{bmatrix} \beta_{10} - \beta_{00} & V_{10} - V_{00} \\ \beta_{11} - \beta_{01} & V_{11} - V_{01} \end{bmatrix} \cdot \begin{bmatrix} z \\ 1 \end{bmatrix} = \begin{bmatrix} \beta_{10} - \beta_{00} & V_{10} \\ \beta_{11} - \beta_{01} & 0 \end{bmatrix} \cdot \begin{bmatrix} z \\ 1 \end{bmatrix} \# \tag{11}$$

Similarly, for $\boldsymbol{v}(z)$ and $\boldsymbol{w}(z)$:

$$\begin{bmatrix} v_{z0} \\ v_{z1} \end{bmatrix} = \begin{bmatrix} \beta_{20} - \beta_{00} & V_{20} - V_{00} \\ \beta_{21} - \beta_{01} & V_{21} - V_{01} \end{bmatrix} \cdot \begin{bmatrix} z \\ 1 \end{bmatrix} = \begin{bmatrix} \beta_{20} - \beta_{00} & V_{20} \\ \beta_{21} - \beta_{01} & V_{21} \end{bmatrix} \cdot \begin{bmatrix} z \\ 1 \end{bmatrix} \# \tag{12}$$

$$\begin{bmatrix} w_{z0} \\ w_{z1} \end{bmatrix} = \begin{bmatrix} 1 & 0 & -\beta_{00} & -V_{00} \\ 0 & 1 & -\beta_{01} & -V_{01} \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & -\beta_{00} & 0 \\ 0 & 1 & -\beta_{01} & 0 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \# \tag{13}$$

or, in a more concise form:

$$u(z) = \boldsymbol{U}\, z\# \tag{14}$$

$$v(z) = \boldsymbol{V}\, z\# \tag{15}$$

$$w(z) = \boldsymbol{W}\, P\# \tag{16}$$

The matrices required to evaluate the projection $\boldsymbol{P}_{proj}$ on the triangle T are $\boldsymbol{UU}$, $\boldsymbol{UV}$, $\boldsymbol{VV}$,$\boldsymbol{UW}$ and $\boldsymbol{VW}$ obtained by the following relations:

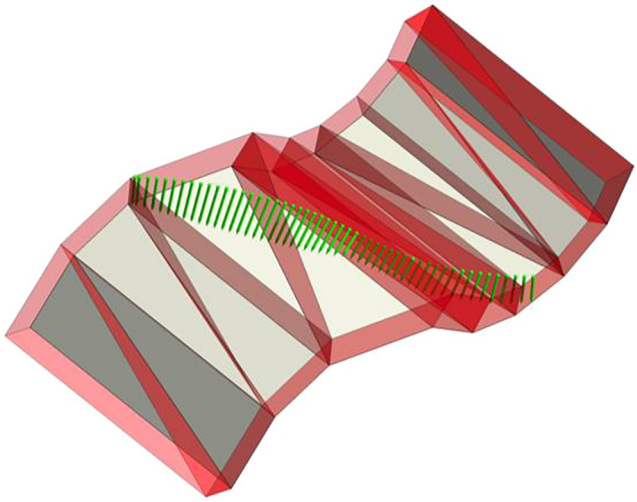

Fig. 4. The SFCP method applied to a surface with only one curvature direction. Light grey: triangles; semi-transparent red: control volumes; green: projection vectors.

$$\begin{aligned} \boldsymbol{UU} &= \boldsymbol{U}^T \boldsymbol{U} \\ \boldsymbol{VV} &= \boldsymbol{V}^T \boldsymbol{V} \\ \boldsymbol{UV} &= \boldsymbol{U}^T \boldsymbol{V} \quad \# \\ \boldsymbol{UW} &= \boldsymbol{U}^T \boldsymbol{W} \\ \boldsymbol{VW} &= \boldsymbol{V}^T \boldsymbol{W} \end{aligned} \tag{17}$$

These matrices can be pre-evaluated once for each triangle T prior to project points on it. Once a point $\boldsymbol{P}$ has to be projected, the trivial vectors $\boldsymbol{p} = [xyz1]^T$ and $\boldsymbol{z} = [z1]^T$ have to be created and the following factors must be calculated:

$$\begin{aligned} uu(z) &= \boldsymbol{z}^T \cdot \boldsymbol{UU} \cdot \boldsymbol{z} \\ vv(z) &= \boldsymbol{z}^T \cdot \boldsymbol{VV} \cdot \boldsymbol{z} \\ uv(z) &= \boldsymbol{z}^T \cdot \boldsymbol{UV} \cdot \boldsymbol{z} \quad \# \\ uw(z) &= \boldsymbol{z}^T \cdot \boldsymbol{UW} \cdot \boldsymbol{P} \\ vw(z) &= \boldsymbol{z}^T \cdot \boldsymbol{VW} \cdot \boldsymbol{P} \end{aligned} \tag{18}$$

These factors permit to evaluate the parametric position (s, t) of the projected point $\boldsymbol{P}_{proj}$ with the following equations:

$$s = \frac{uv(z)\, vw(z) - vv(z)\, uw(z)}{uv(z)^2 - uu(z)\, vv(z)} \# \tag{19}$$

$$t = \frac{uv(z)\, uw(z) - uu(z)\, vw(z)}{uv(z)^2 - uu(z)\, vv(z)} \# \tag{20}$$

The condition (5) must be verified for the projection $\boldsymbol{P}_{proj}$ to be contained in the triangle T while Eqn. (7) gives the resulting projection.

3. Results

Fig. 4 illustrates the case of a discretized single curved surface with all the entities used by the proposed method defined in Section 2. This example mesh highlights how this method projects points along a direction that varies continuously and consistently with the curvature of the original surface. This happens because the vertex normals are evaluated when calculating the mesh that approximates the original surface, and thus they carry information about it that is not found in the triangles left alone. Indeed, there is a substantial gain of consistency with respect to projecting along the direction of triangle normal, mainly for two reasons. The first one inherits that the proposed method "follows" the original surface curvature, as illustrated in Fig. 5., in which two orthogonal views of the same mesh are reported, while projecting along the triangle normal would leads to discontinuities of projection direction over all the mesh. Furthermore, the FCP method subdivides the space surrounding the surface in not overlapping and congruent volumes with no zones where points cannot be projected as

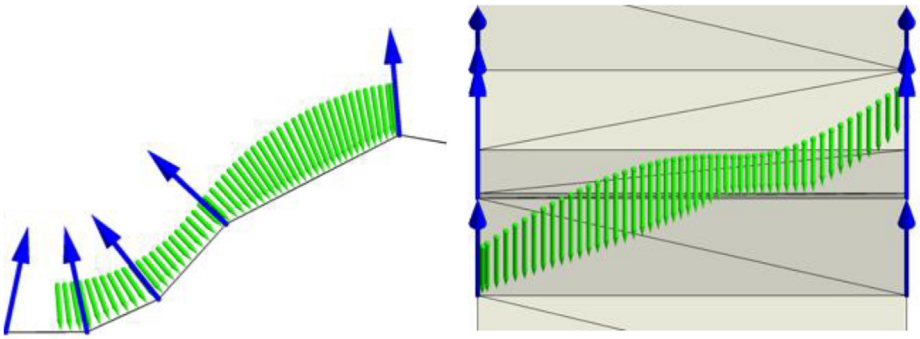

Fig. 5. Detail of the continuous variation of the direction of projection on the same mesh of Fig. 4: left) lateral view; right) frontal view. Light grey: triangles; green: projection vectors; blue: vertex normals.

depicted in Fig. 2 right, where the control volumes described in Section 2.1 are illustrated in semi-transparent red.

The Fig. 6 and Fig. 7 refer to the triangle mesh approximating a free-form surface. In Fig. 7 the view orientation is selected in order to show more clearly the continuous variation of the projection direction and how it follows the vertex normals.

Further examples on general meshes are reported in Fig. 8.

In Table 1 is reported the average computational time of the two versions of the proposed algorithm. Measurements were performed on a Core i7 laptop and do not consider the pre-processing and post-processing time. In case of the optimized algorithm, the measurements include also the matrices building time. This is done in order to compare the sheer performance of the two versions. On average, the optimized algorithm is approximately 10% faster than the standard version. This gain comes at a cost of some extra memory occupation, due to the space required for the matrices. Nevertheless, the speed advantage of the optimized version compensates for the extra memory it requires.

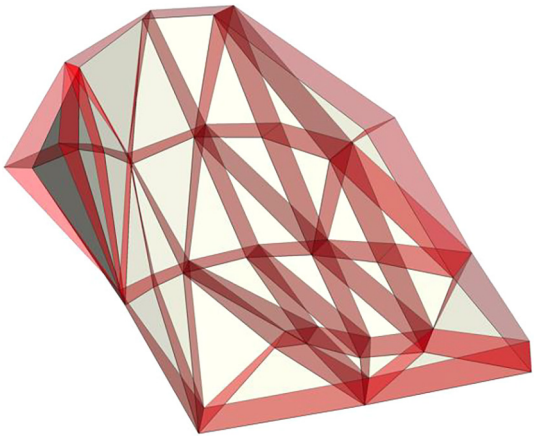

Fig. 6. Projection results in case of a free-form surface. Light grey: triangles; semi-transparent red: lateral faces of the control volumes;

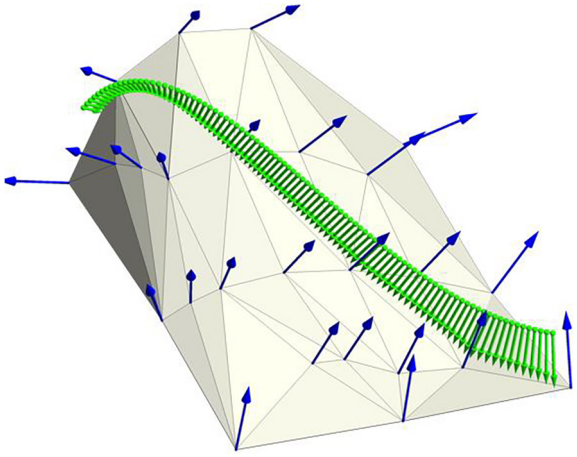

Fig. 7. Detail of the projection shown in Fig. 6. Mesh triangles (light grey) with their vertex normals (blue): note how the projection vectors in green continuously vary their direction accordingly to these normals.

Table 1
. Average computational time of the two versions of the proposed algorithm.

Number of projections	Standard algorithm	Optimized Algorithm	Time Ratio Standard/ Optimized
	[10^6 projections/s]	[10^6 projections/s]	[-]
> 10^6	34.7	30.6	1.133

4. Discussion and conclusions

The proposed algorithm intends to offer an explicit solution for the continuous projection of points on triangle mesh. The problem was approached by the authors with particular regards to laser processing techniques, the reported examples being representative of very frequent situations in those fields. The main outcomes of the work can be summarize as in the following:

- The continuous variation of projection direction given by the proposed algorithm can be highly appreciated in several applications where a more precise representation of the "true" normal is required, such as in laser engraving or milling.
- The novel algorithm allows for a not overlapping and congruent subdivision of the projection volume, feature that is fundamental in the cases of laser processing that are not robust to the path-overlapping outcome.
- In laser processing, the rays must hit the work surface represented by the discretized mesh within a maximum angle to the local normal in order to be effective.
- In case of freeform surfaces, projecting perpendicularly to triangles of a coarse mesh may lead to incorrect results while the mesh refinement will result in a too large scanning head placement.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

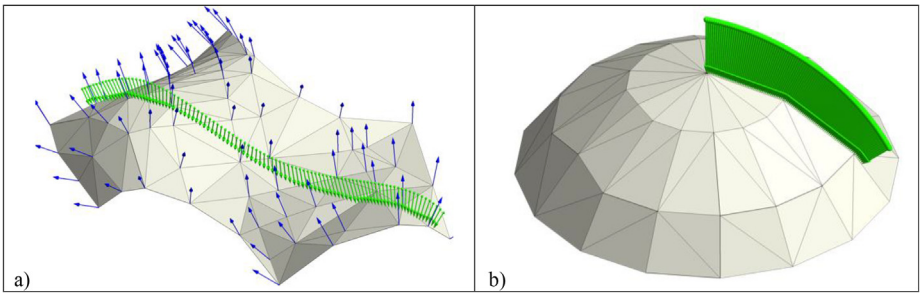

Fig. 8. Further illustrations of the algorithm results: a) freeform surface; b) spherical cap. Light grey: triangles; blue: vertex normals; green: projected vectors.

Acknowledgements

Authors would like to thanks Dr Michele Cotogno for his precious and unevaluable support in the development of this work.

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Appendix A. Supplementary data

Supplementary data to this article can be found online at https://doi.org/10.1016/j.jksuci.2020.06.005.

References

Baldwin, D., Weber, M., 2016. Fast Ray-Triangle Intersections by Coordinate Transformation. Journal of Computer Graphics Techniques (JCGT). 5, 39–49.

Chang, K.-H., Chen, C., 2011. 3D Shape Engineering and Design Parameterization. Comput.-Aided Des. Applic. 8, 681–692. https://doi.org/10.3722/cadaps.2011.681-692.

Cuccolini, G., Orazi, L., Fortunato, A., 2013. 5 Axes computer aided laser milling. Opt. Lasers Eng. 51, 749–760. https://doi.org/10.1016/j.optlaseng.2013.01.015.

Guo, J., Ding, F., Jia, X., Yan, D.-M., 2019. Automatic and high-quality surface mesh generation for CAD models. Comput. Aided Des. 109, 49–59. https://doi.org/10.1016/j.cad.2018.12.005.

Hege, H.-C., Polthier, K., 2011. Visualization and mathematics III. Springer, Berlin; London.

Jones, M.W., 1995. 3D Distance from a Point to a Triangle, Technical Report CSR-5-95 - Department of. Computer Science, University of Wales Swansea, p. 6.

O'Rourke, J., 1998. Computational geometry in C. Cambridge University Press.

Phong, B.T., 1975. Illumination for computer generated pictures. Commun. ACM. 18, 311–317. https://doi.org/10.1145/360825.360839.

Porumbescu, S.D., Budge, B., Feng, L., Joy, K.I., 2005. Shell Maps. ACM Trans. Graph. 24, 626–633. https://doi.org/10.1145/1073204.1073239.

Pottmann, H., Leopoldseder, S., Hofer, M., Steiner, T., Wang, W., 2005. Industrial geometry: recent advances and applications in CAD. Comput. Aided Des. 37 (7), 751–766. https://doi.org/10.1016/j.cad.2004.08.013.

Ramsey, S.D., Potter, K., Hansen, C., 2004. Ray Bilinear Patch Intersections. Journal of Graphics Tools. 9, 41–47. https://doi.org/10.1080/10867651.2004.10504896.

Schmidt, M., Zäh, M., Li, L., Duflou, J., Overmeyer, L., Vollertsen, F., 2018. Advances in macro-scale laser processing. CIRP Ann. 67, 719–742. https://doi.org/10.1016/j.cirp.2018.05.006.

Sun, X., Jiang, C., Wallner, J., Pottmann, H., 2016. Vertex Normals and Face Curvatures of Triangle Meshes. In: Bobenko, A.I. (Ed.), Advances in Discrete Differential Geometry. Springer, Berlin, Heidelberg, Berlin, Heidelberg, pp. 267–286. https://doi.org/10.1007/978-3-662-50447-5_8.

G. Tani, L. Orazi, A. Fortunato, G. Campana, G. Cuccolini, Laser hardening process simulation for mechanical parts, in: S.J. Davis, M.C. Heaven, J.T. Schriempf (Eds.), San Jose, CA, 2007: p. 645404. https://doi.org/10.1117/12.700345