Optimization of toolpath trajectories on arbitrary surfaces

by MIGUEL BELBUT GASPAR AND NELSON MARTINS-FERREIRA

Centre for Rapid and Sustainable Product Development,
Polytechnic Institute of Leiria, Marinha Grande, Portugal

Abstract: Given an arbitrary region on the plane, modelled as a graph with a symmetry, we describe a procedure to find the best orientation for slicing, via a zigzag tool-path trajectory based curve, in order to minimize its discontinuities. We also develop some directions on how to generalize the procedure up to the level of optimizing tool-path trajectories on arbitrary surfaces rather than planar regions only.

Keywords: toolpath generation zig-zag strategy optimization additive manufacturing.

1 Introduction

The generation of optimized tool-path trajectories is a demanding and computationally difficult problem that has been considered since the early ages of CAD/CAM systems of rapid prototyping and rapid manufacturing of new products in the area of industry and enterprises in general. Hence it has already been addressed for the last thirty years or so. Nevertheless, the more recent application of 3-D printing, namely to medical application has provided new and challenging problems in this area. The old methods that were already well established are no longer applicable as the technology has moved from the point of view of subtractive manufacturing to additive manufacturing, suggesting new perspectives and strategies of machine toolpath generation. One of the big constraints of this systems is the necessity of a good finishing detail level. In particular it is a heavy handicap the impossibility of always having a continuous toolpath for a whole region at a given layer. We will present some aspect of implementation and optimization for this kind of toolpath generation systems. We will give special attention to the so-called zig-zag strategy, giving some details on how it is modelled and implemented in a computer system, and also on how to optimize the number of discontinuities in the path. These results are already implemented for planar regions but we will also give further indications on how they can be extended to the level of trajectories to cover regions in arbitrary triangulated surfaces. In this text we will also explain how to develop a procedure to automatically generate toolpath trajectories in abstract triangular spaces, with possibly applications to other areas of interest rather that the rapid-prototyping and rapid-manufacturing of products for the industry.

This text is only an extended abstract of a ongoing work. It presents ideas and algorithms that can be implemented in any computational system such as Matlab. It considers the case of tool-path optimization trajectories in arbitrary planar regions defined by a graph with a symmetry, as it is defined in [1]. The trajectories by themselves are generated with an algorithm described in [2]. At the end we give some directions on how to generalize the results for arbitrary surfaces.

2 Establishing the framework

As it is explained in [1], an arbitrary region in the plane may be described as a graph with a symmetry, that is a system (A, B, d, c, φ) in which

$$A \underset{c}{\overset{d}{\rightrightarrows}} B$$

is a directed graph and

$$\varphi\colon A \to A$$

is a bijection such that $d\varphi = c$. As usual, the elements in A are considered as directed edges with an element $a \in A$ pictured as

$$d(a) \xrightarrow{\ a\ } c(a)\ ,$$

and its image by φ considered as the successor directed edge defining the contour of the region

$$d(a) \xrightarrow{\ a\ } c(a) \xrightarrow{\ \varphi(a)\ } c\varphi(a)\ ,$$

with the assumption that its interior is always on the left.

For practical purposes we give a specific example which will be used throughout the text to illustrate the several steps involved in the outlined procedure.

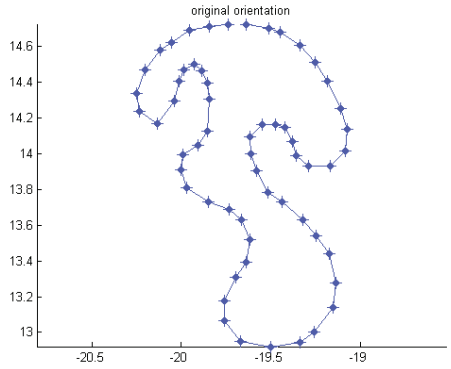


Fig. 1 - Original curve orientation.

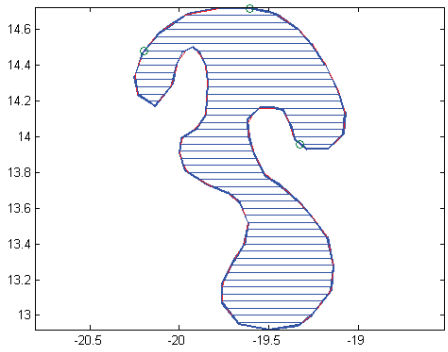


Fig. 2 - Zig-zag toolpath trajectories.

The example is depicted in Figure 1 above and it is defined by the following data, presented as a sequence of directed edges, each one determined by its endpoints, encoded in a vector named P. In other words, we have a set $A = \{1, 2, 3, \ldots, 58, 59\}$, and a set $B \subset \mathbf{R}^2$. The vector P is displayed as:

```
P =
   −19.3566,  13.9876
   −19.2874,  13.9291
   −19.1665,  13.9291
   −19.0801,  14.0168
   −19.0719,  14.1372
   −19.1089,  14.2534
   −19.1829,  14.4065
   −19.2515,  14.5121
   −19.3360,  14.6071
   −19.4487,  14.6820
   −19.5121,  14.7039
   −19.6388,  14.7259
   −19.7368,  14.7259
   −19.8404,  14.7113
   −19.9556,  14.6893
   −20.0536,  14.6235
   −20.1169,  14.5797
   −20.2033,  14.4700
   −20.2494,  14.3385
   −20.2321,  14.2361
   −20.1342,  14.1703
   −20.0363,  14.2946
   −20.0132,  14.4042
   −19.9853,  14.4698
   −19.9246,  14.5015
   −19.8824,  14.4645
   −19.8507,  14.3959
   −19.8402,  14.3062
   −19.8507,  14.1267
   ...
   ...
   −19.9035,  14.0475
   −19.9880,  13.9947
   −19.9985,  13.9102
   −19.9672,  13.8121
   −19.8462,  13.7317
   −19.7310,  13.6879
   −19.6619,  13.6294
   −19.6158,  13.5197
   −19.6388,  13.3955
   −19.6964,  13.3077
   −19.7598,  13.1762
   −19.7598,  13.0665
   −19.6676,  12.9496
   −19.5006,  12.9203
   −19.3335,  12.9423
   −19.2586,  13.0007
   −19.1492,  13.1396
   −19.1377,  13.2785
   −19.1722,  13.4393
   −19.2471,  13.5417
   −19.3220,  13.6294
   −19.4372,  13.7317
   −19.5179,  13.7829
   −19.5812,  13.9072
   −19.6100,  14.0022
   −19.6158,  14.0972
   −19.5467,  14.1630
   −19.4718,  14.1630
   −19.4199,  14.1484
   −19.3796,  14.0680
```

The maps d, c are, respectively, the first and second components, while the map φ is defined by $\varphi(i) = i+1$ if $i \leq 59$ and $\varphi(59) = 1$. The maps d and c may be also seen in terms of the vector P as $d(i) = P(i, 1)$, $c(i) = P(i, 2)$.

In the next section we assume a vector such as P is defined and will deduce a collection of possible orientations which minimize the tool-path trajectories, as exemplified in Figure 2.

3 Finding the best orientation for slicing

Given P as before, we want to minimize the number of inflection points for a scan angle θ. The algorithm is described in pseudo-code, using Matlab syntax.

```
Input: P – n–by–2, polygon's vertices as
rows (x,y)
Output: Aopt – m–by–2, m ranges (amin,amax)
of optimal angles.

% convert to complex form
Pc=P*[1;i];

% Obtain the orientation of each edge:
A=angle(Pc([2:end,1])−Pc)*180/pi

% Get inner angles
IA=mod(mod(180−A,360)+mod(A([2:end,1]),360)
    ,360);
```

We are interested in the concavities, i.e, `IA>180`. At a convex vertex, as the scan slope passes the emerging edge's slope, that vertex ceases to be an inflection point. At the same time, the next vertex becomes a new inflection point, if it is convex, too. So as this scanning slope is reached, no new net inflection points are added. Something different occurs when we consider concave vertices: as we reach the emerging edge's slope, a new inflection point appears (in fact, two).

When two successive vertices are concave, no net inflection point is added, either.

```
% Concave vertices:
CC=IA>180

% Changes, that is, sequences convex−>concave
% This is the non−trivial difficult part
DC=diff(CC([end,1:end]))

% Sort the angles (mod 180):
A3=sortrows([mod(A,180), DC])
```

By cummulative sum of the net change in number of inflection points, we find the range of theta with the minimum amount of infl. points:

```
A4=[A3(:,1), cumsum(A3(:,2))]

% Extract the needed information:
ixm=find(A4(:,2)==min(A4(:,2)));

tmp=[ixm-ixm([end,1:end-1]),...
     ixm([2:end,1])-ixm]~=1
ixm=[ixm(tmp(:,1)),ixm(tmp(:,2))]
AOpt=[A4(ixm(:,1),1),A4(ixm(:,2),1)]
```

The result of each of the preceding steps applied to the input data P as defined in the previous section is presented at the end.

The output result is presented as ranges of angles in the form

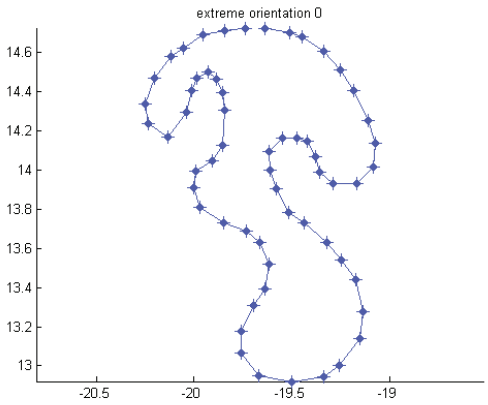
```
ang_min1, ang_max1;
ang_min2, ang_max2;
...
ang_mink, ang_maxk
```

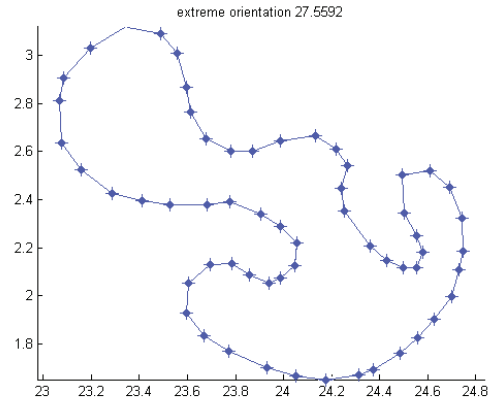
which for the working example has the following result.

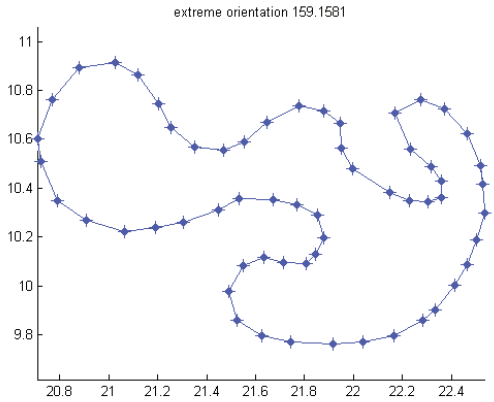
```
AOpt =
         0   27.5592
  159.1581  170.1824
```

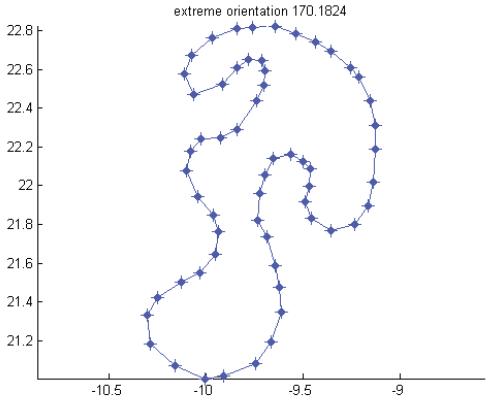
This means that for rotations with an angle between 0 and 27.5592 degrees, there are a minimum number of discontinuities in the tool-path trajectory. For rotation angles ranging from 27.5592 to 159.1581 degrees, the number of discontinuities is increasing while it again attends a minimum when the rotation angle is between 159.1581 and 170.1824. Again form 170.1824 to 360 there is a larger number of discontinuities.

The suggested optimal angles are displayed in the following pictures.









4 Conclusion

To conclude we remark that the procedure may be generalized to the context where the region is not necessarily planar but is embedded into an arbitrary surface in the space. This is important for the purpose of generating tool-path trajectories for rapid manufacturing and rapid prototyping since it will allow a greater number of applications. To to that we will only need to transform the structure used in [2] by imposing that the linking lines between two intersected points in the same g-component, to use the nomenclature of [2], are no longer straight lines but are geodesic path instead. This part is not yet implemented. It will use algorithms for working with

geodesic paths and then combine the structure (E, r, g, s) used in [2],requiring also the geometric information concerning the convexity of each edge, used in here, obtaining thus the same results for surfaces.

References

[1] Gaspar, Miguel Belbut, and Nelson Martins-Ferreira. *A procedure for computing the symmetric difference of regions defined by polygonal curves*, Journal of Symbolic Computation **61** (2014) 53–65.

[2] Miguel Belbut Gaspar, Nelson Martins-Ferreira, *Sobre estratégias de varrimento contínuo para o preenchimento de regiões arbitrárias no plano*, Boletim da Sociedade Portuguesa de Matemática (2010), no. Especial, 78–86.

Appendix

Below is presented the output of each relevant step in the execution of the procedure described in the text. Each vector is listed from top to bottom and from left to right and from one page to the next one.

```
A =              -90.0000          106.8612          123.0239          -33.8943
                 -51.7612           93.4688          131.6335           51.7612
 -123.6901        -9.9282           43.5840          146.3987           78.1326
 -147.9946         7.4788                 0          160.9096           66.9420
  -97.1250        37.9872          -15.7485          170.1824           27.5592
  -72.2576        51.7612          -63.3678          180.0000          -41.1859
  -33.6127        85.2582          -74.0102         -171.9742          -65.2249
  -20.8419       102.1292          -40.2314         -169.2226          -83.2902
  -40.2314       126.1941                 0         -146.1057          -93.3665
  -67.2041       130.4869           45.4323         -145.3096
 -100.5043       138.3852           86.1050         -128.2388
 -123.2922       147.6048          107.7053         -109.3018
 -115.7139       117.0168          115.7693          -80.4154
```

```
CC =             1                 1                 1                 1                 0
                 1                 1                 1                 1                 0
      0          1                 0                 1                 1                 0
      1          1                 0                 1                 1                 0
      1          1                 0                 1                 1                 0
      1          1                 0                 1                 1                 0
      1          1                 0                 1                 1
      0          1                 0                 1                 1
      0          1                 0                 1                 1
      0          1                 0                 1                 1
      0          1                 1                 1                 0
```

```
DC =             1                 0                 0                 0                 0
                 0                 0                 0                 0                 0
      0          0                -1                 0                 0                 0
      1          0                 0                 0                 0                 0
      0          0                 0                 0                 0                 0
      0          0                 0                 0                 0                 0
      0          0                 0                 0                 0
     -1          0                 0                 0                 0
      0          0                 0                 0                 0
      0          0                 0                 0                 0
      0          0                 1                 0                -1
```

```
A3 =
         0         0
         0         0
         0         0
    7.4788         0
    8.0258         0
   10.7774         0
   27.5592         0
   32.0054    1.0000
   33.8943         0
   34.6904         0
   37.9872         0
   43.5840         0
   45.4323         0
   51.7612         0
   51.7612         0
   51.7612         0
   56.3099         0
   56.7078    1.0000
   64.2861         0
   66.9420         0
```

```
   70.6982         0
   78.1326   -1.0000
   79.4957         0
   82.8750         0
   85.2582         0
   86.1050         0
   86.6335         0
   90.0000         0
   93.4688         0
   96.7098         0
   99.5846         0
  102.1292         0
  105.9898    1.0000
  106.8612         0
  107.7053         0
  107.7424         0
  112.7959         0
  114.7751         0
  115.7693         0
  116.6322         0
  117.0168         0
```

```
  123.0239         0
  126.1941         0
  128.2388         0
  130.4869         0
  131.6335         0
  138.3852         0
  138.8141         0
  139.7686         0
  139.7686         0
  146.1057         0
  146.3873         0
  146.3987         0
  147.6048   -1.0000
  159.1581   -1.0000
  160.9096         0
  164.2515         0
  170.0718         0
  170.1824         0
```

```
A4 =     0         0
         0         0
         0         0
    7.4788         0
    8.0258         0
   10.7774         0
   27.5592         0
   32.0054    1.0000
   33.8943    1.0000
   34.6904    1.0000
   37.9872    1.0000
   43.5840    1.0000
   45.4323    1.0000
   51.7612    1.0000
   51.7612    1.0000
   51.7612    1.0000
   56.3099    1.0000
   56.7078    2.0000
   64.2861    2.0000
   66.9420    2.0000
```

```
   70.6982    2.0000
   78.1326    1.0000
   79.4957    1.0000
   82.8750    1.0000
   85.2582    1.0000
   86.1050    1.0000
   86.6335    1.0000
   90.0000    1.0000
   93.4688    1.0000
   96.7098    1.0000
   99.5846    1.0000
  102.1292    1.0000
  105.9898    2.0000
  106.8612    2.0000
  107.7053    2.0000
  107.7424    2.0000
  112.7959    2.0000
  114.7751    2.0000
  115.7693    2.0000
  116.6322    2.0000
```

```
  117.0168    2.0000
  123.0239    2.0000
  126.1941    2.0000
  128.2388    2.0000
  130.4869    2.0000
  131.6335    2.0000
  138.3852    2.0000
  138.8141    2.0000
  139.7686    2.0000
  139.7686    2.0000
  146.1057    2.0000
  146.3873    2.0000
  146.3987    2.0000
  147.6048    1.0000
  159.1581         0
  160.9096         0
  164.2515         0
  170.0718         0
  170.1824         0
```

```
tmp =
     1     0
     0     0
     0     0
     0     0
     0     0
```

```
     0     0
     0     1
     1     0
     0     0
     0     0
     0     0
     0     1
```

```
ixm =
     1     7
    55    59
```