Available online at www.sciencedirect.com

ScienceDirect

Procedia CIRP 00 (2014) 000–000

www.elsevier.com/locate/procedia

9th CIRP Conference on Intelligent Computation in Manufacturing Engineering - CIRP ICME '14

CNC Paths Optimization in Laser Texturing of Free Form Surfaces

L. Orazi[a*], F. Montanari[a], G. Campana[b], L. Tomesani[b], G. Cuccolini[c]

[a]*DISMI – University of Modena and Reggio Emilia, Reggio Emilia*
[b]*DIN – School of Engineering and Architecture, University of Bologna, Bologna*
[c] *C.B. Ferrari S.p.A., Modena*

* Corresponding author. Tel.: +39 320 4332627; fax: +39 0522 522609 *E-mail address*: leonardo.orazi@unimore.it

Abstract

Laser Texturing and Laser Engraving of free form surfaces are interesting alternative to EDM and micro milling in the fabrication of moulds and dies. The material removal rate of this process is typically very low and the maximum working area is limited by the maximum travel range of the galvanometric scanning head and also by the maximum deflection of the scanner or the surface curvature. As a consequence, in order to manufacture large or free form surfaces, a set of placements of the scanning head respect to the work piece must be calculated and controlled by the CNC.
The number of these placements can be quite high due the small amount of material removed in each of them and the non-operative time spent to move the scanning head can be relevant compared to the overall time. In this work a method based on the solution of the Traveling Salesman Problem is proposed with the aim to optimise the number of displacements of the scanning head and generally in order to reduce the number of movement of the numerically controlled axes.
The method that takes into account both the architecture and the dynamic characteristics of a 5 axis CNC system was implemented in the CALM (Computer Aided Laser Manufacturing) software used to program the laser path for part texturing and applied in industrial cases.

Keywords: Laser; Texture; Optimization; Traveling Salesman Problem

1. Problem Description.

The application of laser technology in manufacturing processes has increased very quickly in the last years. Many thermal processes like welding, cutting but also milling use this kind of heat source. Laser manufacturing have a lot of advantages than traditional machining: higher speed and precision are guaranteed due to the physical characteristics of laser radiations and the contact-less work method. Nevertheless, the differences between laser technology and conventional machining cause some peculiar and critical issues due to the nature of the interaction between the tool and the raw surface. In conventional milling, the material is removed due to the shearing forces exchanged between tools and work-piece. Laser technology interacts with the material by means of a thermal process that heats up the material until the evaporating temperature or above the ablation threshold.

An important issue of Laser Texturing (LT) is well evident when the laser operates by overlapping paths. In Fig. 1 is showed a typical overlapping working area and the error generated during a laser machining in comparison with a traditional machining.

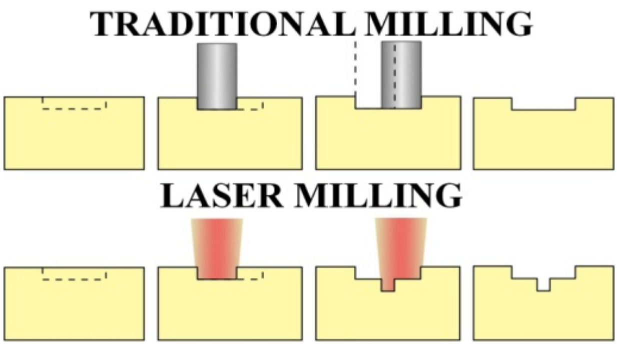

Fig.1 – Error due to overlapping paths.

To solve the mentioned problem, a partition of the working area is necessary but it generates some further issues for managing the displacement movements of the laser beam or the scanning head. The number of placements can be often very high, in the order of thousands and they have to be repeated for every processed layer. The solution of the overlapping paths problem needs a strategic approach in order to forecast and control the overall machining time.

The authors of this article propose to implement the Traveling Salesman Problem (TPS) with the aim of minimizing the airtime of a LT operation.

2. Previous approaches

The TPS problem is an already known problem in the field of the operative research. A description and the formulation of this problem will be discusses in the next sections. Many issues regarding minimizing path or finding the cheapest circuit, if the cost is not a length and is expressed by a formulation, can be traced by the TSP and an algorithm to solve it can be used.

At first, the attention for optimizing the traditional milling processes was focused on the working time optimization, in this way the productiveness increased. Successively the interest to reduce the machining time was focused on the airtime, which is not a productive time and reduces efficiency and yield return of machining process. This time is relevant when small areas and lots of points have to be processed, as discussed in [1] by Castelino et al.

A first implementation for reducing the path during the laser cutting of nested stock was discussed by Han and Na [2]; they developed a simple algorithm to find the optimal solution. Others applications of algorithms with the aim to reduce the displacement time are proposed by Oysu and Bingul [3] by means of a hybrid genetic algorithm to reduce the airtime, or by Kolahan and Liang [4] that considered not only the displacement time but also the time spent on tool change and the tool wear and maintenance.

The authors tested the adopted solution on a free form object using a five axis CNC machine and described in the following pages. In scientific literature several articles dealt with the problem of machining a free form and the issue generated using a 5 axis CNC. For example, in the work by Lazoglu et al. [5] not only the path length but also the force generated by the tool during the machining was optimized. A second article by Stanislav Makhanov [6] described the optimization of the tool path of a 5 axis milling machine and in these case the movements of the tools was partially solved.

3. A Computer Aided Laser Manufacturing system

Authors already presented in [8] Computer Aided LASER Manufacturing system that was named CALM. It is a CAM software designed specifically for laser manufacturing (milling and texturing) to guarantee a correct processing usage of a 3 or 5 axes CNC machine. This software was developed in C++ language and it can be used on the different operating systems available on the market.

A further software tool was then implemented in CALM for solving the TPS problem.

3.1. Placement Triangles

CALM processes the input surfaces and generates two triangulations, applying a classical algorithm proposed by Piegl in [7]. The first triangulation is generated minimizing the chordal error in order to approximate accurately the real raw surface and avoiding errors during machining; thus a high density of triangles characterizes this first triangulation.

A second type of triangulation, which represents the placement of the scanning head, is created on the raw part for partitioning the working area and solving the problem described in the previous section. Those triangles have a larger area than the other, because they do not need to approximate as well as possible the surface of the raw part. The parameters that determine the size of the placement triangles area are dependent on the focusing ability of the machining head and on the geometry of the free form surface. They are:

- Focusing distance.
- Spot laser.
- Curvature of the raw surface.
- Presence of high depth to width ratio.

Another aspect that must be considered for generating the placement triangulations is the Material Removal Rate (MRR): it is usually possible to remove about 1 μm of material for a single placement per each layer. A repetition of the placement triangulation per each layer determines a recursive error that is evident in the border of each triangle of the laser-machined surface. For this reason the raw part is divided into layers and then a different placement triangulation can be calculated for each layer because otherwise it is not possible to guarantee the highest quality of the machined surface.

3.2. Output files

CALM produces the following output files for all the devices of the laser milling system such as:

• PROJECT. JOB: contains the list of the HPG files in the order in which they must be executed.

• HPG files for the scanning head, these files contain the geometrical coordinates on which the source is switched on or off and the movements of the two galvanometric mirror of the head.

• PP.ISO files describe the movement of the axes of the CNC in the G-code. The minimization of the number of displacements regards the PP.ISO file, as the path of the tool is contained in it.

• PARAM.PRO contain the processing parameters for the laser source.

4. Traveling Salesman Problem

The geometry of the raw part is divided in layers by CALM and each of them is triangulated for solving the overlapping areas problem. To minimize the displacement time between the triangles the authors have analyzed the algorithms proposed to solve the Traveling Salesman Problem (TSP). The TSP consists in visiting a set of customers, only once, the target is minimizing the path length (airtime). The customers for CALM are the gravity center of the placement triangles.

4.1. Mathematical description

Given a graph (or a complete graph*) $G = (V, E)$ where $V = \{1,2,..,n\}$ is the set of vertex (customers) and E is the set of edges with an associated cost c_{ij} with $i, j \in V$, the problem consist in finding the shortest (or the cheapest) Hamiltonian circuit, which is a close circuit that visits one and only one time each vertex of the graph. The mathematical model can be described by setting a decision variable as follows:

$$x_{ij} = \begin{cases} 1, \text{if the edge from } i \text{ to } j \text{ is chosen} \\ 0, \text{otherwise} \end{cases}$$

In this article is represented the simplest mathematical model for TSP:

$$minimize \ \sum_{j=1}^{n} \sum_{i=1}^{n} c_{ij} \cdot x_{ij} \quad (1)$$

$$\sum_{i=1}^{n} x_{ij} = 1 \quad \forall j \in V \quad (2)$$

$$\sum_{i=1}^{n} x_{ji} = 1 \quad \forall j \in V \quad (3)$$

$$\sum_{i \in S} \sum_{j \in S} x_{ij} \leq |S| - 1 \quad \forall S \dot{\subseteq} V : |S| \leq |V| - 1 \quad (4)$$

$$x_{ij} \in \{0,1\} \quad \forall i, j \in V \quad (5)$$

The objective function (eq. 1) minimizes the total cost of the circuit. Eq. 2 describes a constraint condition or that is necessary to go one time in each vertex $j \in V$. Eq. 3 imposes a further constraint or that is necessary to exit exactly one time from the vertex. In this way each vertex is visited one and only one time. Eq. 4 clarifies another constraint: it is necessary in order to eliminate the possible Sub-Tours, otherwise the best solution will be a N-recursive circuit.

In the scientific literature of operative research exists other mathematical models to express the constraints, like MTZ (Miller-Truckler-Zemlin) formulation, one-commodity formulation or the multi-commodity formulation. All those are different formulations to simplify the constraints in eq. 4. The mathematical procedure always finds an optimal solution, but in real cases the optimal solution can be hard to find.

4.2. Computational complexity

The TSP, unfortunately, is a difficult problem to solve also for small instances. The number of constraints increases much faster than the size of instances and there are no algorithms that can find the optimal solution in a practicable time.

The class of this kind of problem in the operative research is named as NP-hard problem, Non-deterministic Polynomial-time hard. The most diffused methods of solutions are:

- Heuristic Algorithms, most of them are freely available on the web.
- Genetic Algorithm, that are difficult to be applied in our case due to the parameters setup.
- Ant Colony Optimization, described by Dorigo in [8] [9].

5. The Concorde program

Concorde is not an algorithm but a program developed in C ANSI code for solving TSP problems. It won several prizes in solving big instances [13] and it can be considered the state of art of TSP's solvers. Concorde usually finds a good solution in few seconds for an instance with thousand of vertexes and presents a small gap than the optimal mathematical solution. The software is free for research scope and is available as source code or as user-friendly program. In this work the source code was adopted. Concorde contains more than 700 functions, it can solve different kinds of TSP and uses different algorithms for finding as fast as possible the better solution.

Concorde is based on is the chained Lin-Kernighan heuristic algorithm.

5.1. Chained Lin-Kernighan Heuristic (LKH)

CALM triangulates each layer of the free-form surfaces. The chained Lin-Kernighan Heuristic (LKH) algorithm can solve efficiently a 2D-TSP. LKH startegy for a 2D instance can be summarized in the following points:

- Create a random Hamiltonian tour.
- Calculate the gain of swapping two or three vertex's position in the tour. LKH decides if swapping two or three vertex and uses the best logic improvement, thus each gain is evaluated.
- Swap the most positive profitable set of vertex and return to the previous point.

If the gains are all equal to zero or negative, accept a worst solution in anticipation of finding the best solution among the acceptable current ones.

The loop continues until a parameter is reached, like number of iterations, percentage of gap, value, etc.

To avoid the possibility to dedicate too much times to the same set of vertexes, a tabu-list that penalize the too recurring changes is created. A full description of LKH is proposed in [13].

This algorithm solves well the 2D–TSP when the *Triangle Inequality* condition is guaranteed:

$$\forall (i, j) \in V, \nexists k \in V : c_{ik} + c_{kj} < c_{ij} \quad (6)$$

This request is respected by *CALM* and it gives to *Concorde* as input file a matrix in TSPLIB format. TSPLIB is a language developed specifically for the TSP problems. In this file is contained the description of the problem to be solved: the TSP type, the number of the edges, the type of the edge costs (Euclidean Distances, Geographical Coordinates, Explicit). CALM calculates the costs between the barycenter of the triangles, so the edge costs are explicit.

* In a complete graph exist an edge for each couple $(i; j) \in V, \ i \neq j$

5.2. Integration CALM-Concorde.

During the elaboration of a single layer, *CALM* selects the triangles that have to be processed, generates the HPG file and before writing the PP.ISO file, it calculates the costs to connect each couple of triangles. *Concorde* is then used to optimize the costs. The triangles indexes and the related distances are described in the TSPLIB file that *CALM* exchanges with the *Concorde* program.

Then *Concorde* returns to *CALM* the file with the sequence of the triangles that minimize the airtime.

Finally *CALM* writes the optimized PP.ISO file and passes at the next layer.

6. Matrix Cost calculation

The authors of this article focused their attention on calculating the matrix of operational cost by evaluating the cost due to the real airtime of the CNC machine. *Concorde* is a very efficient resolver but in this article will be not discussed in deep details.

Three different approaches have been considered and the part program has been generated: (a) the "Not Optimized Solution" , (b) the "3D Euclidean distances" and (c) the "5X time distances"

6.1. Not Optimized Solution

In this case *CALM* generate the PP.ISO file without optimizing the sequence. The path is random as generated by the triangulation alghorithm and generally the cycle is time-expensive.

6.2. 3D Euclidean Distance approach

In the TSPLIB file passed to *Concorde*, *CALM* calculates the distances in *mm* by considering the length of the segment which connects the center of gravity of the two triangles. The used formula is:

$$d = \sqrt{(x_a - x_b)^2 + (y_a - y_b)^2 + (z_a - z_b)^2} \quad (7)$$

Concorde calculates the solution that does not represent the minimum airtime but the minimum path for the considered layer. Using this method on a 5 axis system can produce non-optimized solutions. The curvature of the surface is not considered and this can produce unnecessary rotations and large displacements of the scanning head due to the focus distance that is normally relevant.

6.3. 5X Time approach

The approach introduces the cost of the real axes movements in the optimization code. This operation can be done by investigating the machine architecture. In Fig. 2 the machine architecture is shown: translation axes are X, Y and Z; their values are referred to the home position of the CNC system represented by the coordinate frame O_G. For a given placement the point *P* represents the position of the laser spot on the work-piece, while *Q* is the position of the scanning head that is moved and controlled by the CNC axis. The rotations *A* and *B* are referred, respectively, to X and Y.

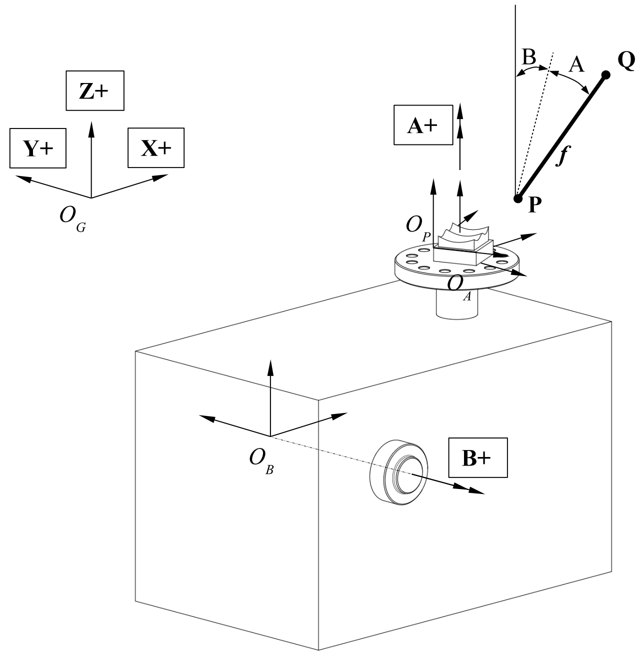

Fig.2 – Machine architecture.

CALM evaluates the positions of P in the work-piece reference system O_P and the optimization performed by using the 3D Euclidean distance approach is evaluated in this reference system. The proposed advanced 5X method requires the evaluation of the movements of *Q* referred to the reference O_G; this is obtained at first by evaluating the point Q_P in the reference O_P:

$$\mathbf{Q_P} = \mathbf{T_{P\rightarrow Q}} \cdot \mathbf{P} \quad (8)$$

The transformation matrix $T_{P\rightarrow Q}$ can be written as below:

$$\begin{bmatrix} \cos(B) & \sin(A)\sin(B) & \cos(A)\sin(B) & f\cos(A)\sin(B) \\ 0 & \cos(A) & -\sin(A) & -f\sin(A) \\ -\sin(B) & \sin(A)\cos(B) & \cos(A)\cos(B) & f\cos(A)\cos(B) \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (9)$$

where f is the focal length. The matrix is obtained by applying the two rotation $T_{P,A}$, $T_{P,B}$ and the translation $T_{P,f}$ perpendicularly to the surface.

The transformation $Q_A = T_{P\rightarrow A}\, Q_P$ from the system O_P to the system O_A is obtained by means of the following matrix:

$$\mathbf{T_{P\rightarrow A}} = \begin{bmatrix} \cos(\theta_P) & -\sin(\theta_P) & 0 & \delta_{P,x} \\ \sin(\theta_P) & \cos(\theta_P) & 0 & \delta_{P,y} \\ 0 & 0 & 1 & \delta_{P,z} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (10)$$

Where $\delta_{P,x}$, $\delta_{P,y}$ and $\delta_{P,z}$ are the position of the work-piece reference system respect the rotary table and ϑ_P is the rotation. The transformation of the position of Q in the system O_B is done by $Q_B = T_{A\rightarrow B}\, Q_A$ where:

$$\mathbf{T_{A\to B}} = \begin{bmatrix} \cos(B) & 0 & \sin(B) & \delta_{A,z}\sin(B)+\delta_{A,x}\cos(B) \\ 0 & 1 & 0 & \delta_{A,y} \\ -\sin(B) & 0 & \cos(B) & \delta_{A,z}\cos(B)-\delta_{A,x}\sin(B) \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (11)$$

with $\delta_{A,x}$, $\delta_{A,y}$ and $\delta_{A,z}$ that represents the position of the reference O_A respect to O_B. There is a final translation from system O_B to system O_G represented by a matrix $T_{B\to G}$.

Then the overall transformation is:

$$Q_G = T_{P\to G} \cdot P = T_{B\to G} \cdot T_{A\to B} \cdot T_{P\to A} \cdot T_{P\to Q} \cdot P \quad (12)$$

the matrix $T_{P\to G}$ contains 12 non trivial coefficients:

$$\mathbf{T_{P\to G}} = \begin{bmatrix} gr_{11} & gr_{12} & gr_{13} & gt_{14} \\ gr_{21} & gr_{22} & gr_{23} & gt_{24} \\ gr_{31} & gr_{32} & gr_{33} & gt_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (13)$$

The distances between successive positions of Q points are invariant respect $\delta_{A,x}$ and respect $T_{P\to G}$ moreover, for simplicity, it is possible to consider the work-piece reference $O_P = O_A$. With these assumptions the coefficients of $T_{P\to G}$ become:

$$\begin{aligned}
gr_{11} &= \cos^2(B) - \sin^2(B) \\
gr_{12} &= 2\sin(A)\sin(B)\cos(B) \\
gr_{13} &= 2\cos(A)\sin(B)\cos(B) \\
gt_{14} &= \cos(B)\delta_{A,x} + \sin(B)\delta_{A,z} + 2f\cos(A)\sin(B)\cos(B) \\
gr_{21} &= 0 \\
gr_{22} &= \cos(A) \\
gr_{23} &= -\sin(A) \\
gt_{24} &= -f\sin(A) \\
gr_{31} &= -2\sin(B)\cos(B) \\
gr_{32} &= \sin(A)\cos^2(B) - \sin(A)\sin^2(B) \\
gr_{33} &= \cos(A)\cos^2(B) - \cos(A)\sin^2(B) \\
gt_{34} &= -\sin(B)\delta_{A,x} + \cos(B)\delta_{A,z} + f\cos(A)\cos^2(B) - f\cos(A)\sin^2(B)
\end{aligned}$$

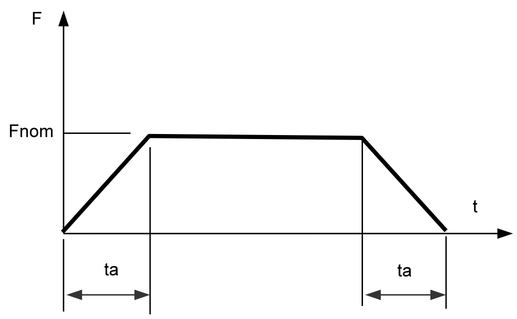

Fig.3 – Motion law

It is now possible to assume that the law of motion for every axis can be approximated as in Fig. 3 where F_{nom} is the nominal feed and t_a is the acceleration and deceleration time of the given axes; under these conditions the time requested for a displacement equals to s is given by Eqn. 14.

$$t = \frac{s}{F_{nom}} + t_a \qquad if\ s > \frac{F_{nom}\,t_a}{2}$$
$$t = \sqrt{2\frac{s}{F_{nom}}\,t\_a} \qquad otherwise \quad (14)$$

The adopted nominal feed, acceleration and deceleration time were experimentally measured and are reported below in Tab. 2.

Table 2 – Experimentally measured motion laws

Axes	F_{nom}	t_a
X	218 mm/s	0.239 s
Y	218 mm/s	0.246 s
Z	218 mm/s	0.245 s
A	2.12 rad/sec	0.207 s
B	0.87 rad/sec	0.156 s

For each movement it is possible to calculate the required time by each axis to reach the final position.

The final cost will be given by the maximum value amongst them: $d = \max\,(t_X, t_Y, t_Z, t_A, t_B\,)$.

7. Numerical and experimental results

The experiments were conducted by processing a feature on a free form surface with a circular planar projection. The feature is shown in Fig. 4. It is composed of 0.98 billion triangles processed in 3043 placements which sequence have to be optimized.

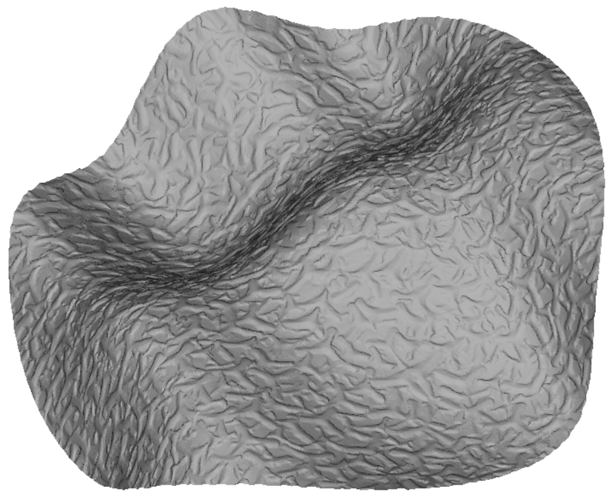

Fig.4 – The feature to process.

In Fig. 5 the sequence of placements was generated by listing one by one the placement triangulation.

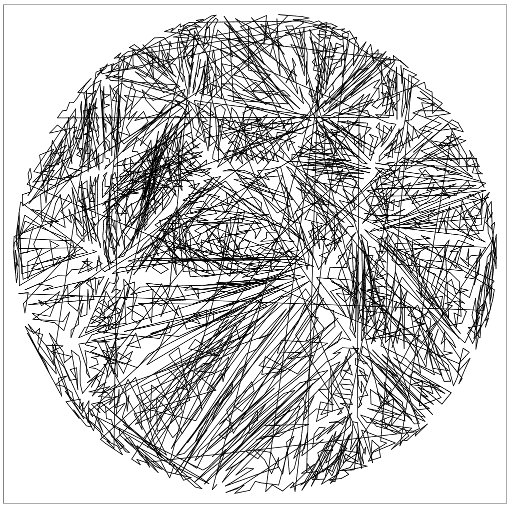

Fig.5 – Non optimized sequence

Fig. 6 shows the result of the optimization based on the 3D euclidean distance. The sequence seen from the top view is intuitively less time-comsuming respect the previous one.

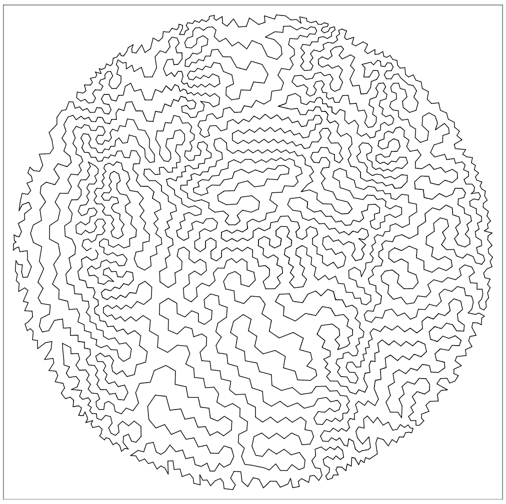

Fig.6 – 3D Euclidean Distance approach

In Fig. 7 the optimization based on the calculation of the 5 axes movements and the machine dinamics. Apparently the results are casual but the total time of processing was substanzially reduced as indicate in Table 3.

Table 3 – Experimentally laser processing time

Method	Experimental Measured Time [s]	Gain [%]
As is	2775	-
3D euclidean	1939	30.1
5X time	1715	38.2

The calculation time of the TSP optimizion is of the order of 2 minutes that appear negligible respect the overall computation time requested to generate the Part Program.

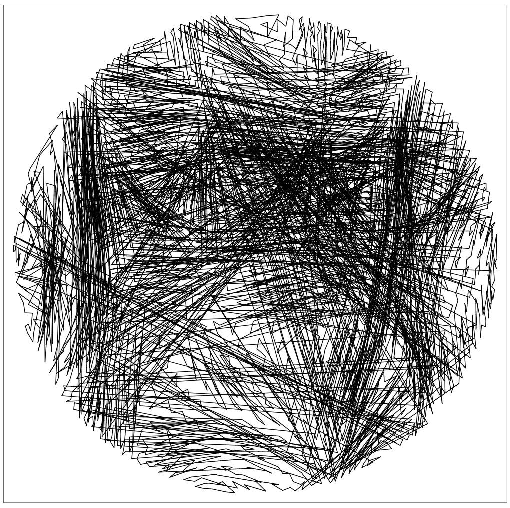

Fig.7 – The feature to process.

References

[1] Castelino K, D'Souza R, Wright PK (2003) Toolpath Optimization for Minimizing Airtime during Machining. Journal of Manufacturing Systems 22(3):173–180.

[2] Guk-chan Han, Suck-joo Na : A study on torch path planning in laser cutting processes part 2: Cutting path optimization using simulated annealing . Journal of Manufacturing Systems, Volume 18, Issue 2, Supplement 1, 1999, Pages 62-70.

[3] Oysu, C., Bingul, Z. (2009) : Application of Heuristic and Hybrid-GASA Algorithms to Tool–path Optimization Problem for Minimizing Airtime during Machining. Engineering Applications of Artificial Intelligence 22, 389–396.

[4] Kolahan F, Liang M. Optimization of hole-making operations: a tabu-search approach. International Journal of Machine Tools and Manufacture International Journal of Machine Tools and Manufacture. 2000;40:1735-53.

[5] Lazoglu I, Manav C, Murtezaoglu Y: Tool path optimization for free form surface machining, CIRP Annals 2009, 58:101-104.

[6] S. Makhanov : Optimization and correction of the tool path of the five-axis milling machine: part 1. spatial optimization. Mathematics and Computers in Simulation, 75 (2007), pp. 210–230 Applied Scientific Computing: Advanced Grid Generation, Approximation and Simulation.

[7] Piegl L.A, Tiller W, Geometry-based triangulation of trimmed NURBS surfaces. Computer Aided Design. 30(1):11-18 (1998).

[8] Cuccolini G., Orazi, L. and Fortunato A. 5 Axes computer aided laser milling. Optics and Lasers in Engineering 51 (6):749-760 (2013).

[9] Tianjun Liao, Thomas Stützle, Marco Antonio Montes de Oca, and Marco Dorigo. A unified ant colony optimization algorithm for continuous optimization European Journal of Operational Research 234(3):597-609 (2014).

[10] Dorigo, M., Maniezzo, V., & Colorni, A. (1991). The ant system: An autocatalytic optimizing process. Tech. Rep. 91-016 Revised, Dipartimento di Elettronica, Politecnico di Milano, Italy.

[11] M. Dorigo, V. Maniezzo, A. Colorni : Ant system: Optimization by a colony of cooperating agents, IEEE Transactions on Systems, Man, and Cybernetics – Part B, 26 (1) (1996), pp. 29–41.

[12] M. Dorigo, T. Stützle: Ant colony optimization, MIT Press, Cambridge, London, England (2004).

[13] Applegate D.L., Bixby R.E., Chvátal V., Cook W., Espinoza D.G., Goycoolea M., Helsgaun, K., Certification of an optimal TSP tour through 85,900 cities, (2009) Operations Research Letters, 37 (1), pp. 11-15.

[14] Applegate, D., Cook, W., Rohe, A.: Chained Lin-Kernighan for large traveling salesman problems, (2003) INFORMS Journal on Computing, 15 (1), pp. 82