2018 European Control Conference (ECC)
June 12-15, 2018. Limassol, Cyprus

Vector Field Analysis for Optimization of the Tool Path of the Five-Axis Milling Machine*

L. V. Dang, S.S. Makhanov

Abstract— **The paper presents a method to increase the production rate of a five axis milling machine through improving how to change the coordinates of the to-be-milled points from the "workpiece" to the "machine" coordinate system, and then deciding the cutting path by maximizing the material removal rate. The method to optimize the tool path is based on the analysis of a vector field of optimal directions which can be pre-computed for a particular machine configuration. The method includes grid generation, space filling curves and vector field decomposition using rotationally invariant moments. The proposed tool path performs the machining 1.2-6 times faster than conventional methods.**

I. Introduction

Machining large industrial parts requires tens, hundreds of thousands or even millions of cutter location points and hundreds hours of machining. Reduction of the machining time through optimization of the tool path is one of the most important topics in the control of five-axis milling machines.

The machine is guided by a controller which is fed with the CNC program which is a sequence of commands or a *G* code $G = \{G_1, G_2, ..., G_N\}$ carrying three spatial coordinates of the tool-tip (cutter location points or CL-points) and a pair of rotation angles to establish the orientation of the tool. We shall call the first rotary axis the *B*-axis and the second rotary axis the *A*-axis. The following coordinate systems are used to model the kinematics of the machine: the workpiece coordinate system O_1, a coordinate system of the first rotary part O_2, a coordinate system of the second rotary part O_3 and a coordinate system of the spindle O_4 (Fig. 1).

Mathematically the problem can be formulated as follows:

$$\underset{G}{\text{minimize}}(T)$$
$$\text{subject to}$$
$$\varepsilon \le \varepsilon_0, h \le h_0,$$

where T is the total machining time, ε is the Hausdorff distance between the tool tip trajectories and the part surface, h is the scallop height and ε_0, h_0 prescribed tolerance. Usually, $\varepsilon_0 = h_0$.

The idea to match the tool path and a vector field (VF) of "optimal" directions to reduce the machining time was proposed by Chiou and Lee [1]. The recent modifications are the tensor based approach [2] and segmentation into regions of similar pathlines [3]. A comprehensive survey of this type of optimization is given in [4]. The majority of VF methods employ multiple initial paths which are propagated inside the region. It is not hard to show that the approach is not always accurate from the viewpoint of the global optimization. It may also be sensitive to local variations of the optimization criteria. Moreover, finding the initial tool path is a computationally expensive, NP hard problem. Finally, only a few methods have been applied to a real industrial format such as Stereolithography (STL), Initial Graphics Exchange Specification (IGES) or Standard for the Exchange of Product model data (STEP) [5].

We propose a new toolpath generation method based on curvilinear grids [6]-[8] and Adaptive Space Filling Curves (ASFC) [9]-[10] applicable to one of the most popular industrial STL format. The resulting VF aligned path is a compromise between the flexibility of methods based on propagating of the initial track and the simplicity of the zigzag topology. The optimization can be performed using a variety of criteria such as the machining time, tool path length, kinematic error, etc. This version of the algorithm constructs a curvilinear path which partly or even entirely aligns with the direction of the maximum material removal rate. The proposed VF adaptive curvilinear toolpath has been tested on a variety of STL surfaces. The reference methods are the standard iso-parametric path (ISO), MasterCam and advanced toolpath generation algorithms of UG-NX9 (Unigraphics-Next Generation of Siemens), namely, Helical/Spiral (HS) and Follow Periphery (FP). The experiments have been performed on a milling machine HAAS VF2TR. Unfortunately, it is not always possible to accurately fit the curvilinear grid to the complex and possibly noisy VF. Therefore, our second modification is based on segmentation of the desired VF and applying the curvilinear tool path in each subdomain independently. The proposed method generates a tool path which performs the machining 1.2-6 times faster than conventional methods. Finally, the model does not include many types of errors such as tool bending, thermal deformations, clamping conditions, tool wear and others. However, as far as the accuracy, scallop height and kinematics errors are considered, the virtual five-axis machine has been successfully tested and compared with the actual machine[10].

*The research is supported by the Center of Excellence in Biomedical Engineering, Thammasat University, Thailand

L.V. Dang and S.S. Makhanov are with Sirindhorn International Institute of Technology, Thammsat University, Rangsit, Pathum Thani 12000, Thailand (e-mails : haidang3a2009@gmail.com, makhanov@siit.tu.ac.th, S.S. Makhanov : 66894881796)

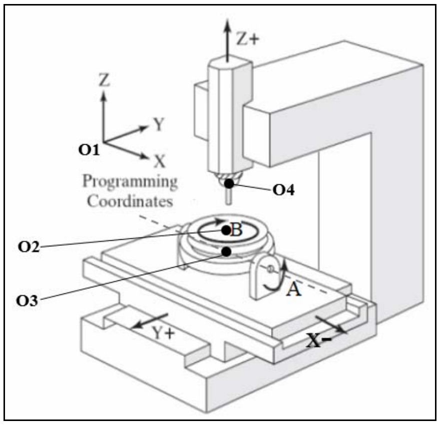


Figure 1 A scheme of 5axis milling machine HAAS VF2TR, National Institute of Metrology, Thailand

II. VECTOR FIELD OF OPTIMAL TOOL DIRECTIONS

Let $S(u,v)=(x(u,v),y(u,v),z(u,v))$ be the required part surface, where *u* and *v* are the parametric variables. Let W_1 be an arbitrary CC (cutter contact) point on $S(u,v)$ (Fig. 2a). Consider a set of points on the surface defined by $\Omega_{W_1}=\{W:\mathrm{dist}_S(W_1,W)=l_1\}$, where dist_S is the geodesic distance and l_1 is a small prescribed step. The corresponding set of points in the machine coordinates is denoted by Ω_{M_1}. The distance between the corresponding points is given by $l_{1,M}\equiv l_{1,M}(W)$. The machining strip corresponding to the feed direction $\overrightarrow{W_1,W}$ is denoted by $w_1\equiv w_1(W)$. The machining strip depends on the effective cutting profile of the tool and the curvature of the part surface [8]. Fig. 2 (b) exemplifies Ω_{W_1} and Ω_{M_1} obtained by the inverse kinematic transformation of HAAS VF2TR. Note that Ω_{W_1} is a geodesic circle, whereas Ω_{M_1} is an irregular closed curve. Clearly, equal increments on the surface (in the workpiece coordinate system) do not lead to equal increments in the machine coordinates. Therefore, the machining time depends critically on the translations and rotations in the machine coordinates rather than in the workpiece coordinates.

Furthermore, introduce an instantaneous material removal rate in the direction $\overrightarrow{W_1,W}$ given by $R_M(W)=\dfrac{F\,l_1(W)w_1(W)}{l_{1,M}(W)}$, where F is the feed rate. The machining strip w_1 corresponding to the prescribed feed direction is evaluated by locating the intersections of the effective cutting profile and the design surface[11]. Note that the material removal rate is measured in $\mathrm{mm}^2/\mathrm{sec}$ and does not include the thickness of the blank workpiece. This is a common assumption of this type of machining models.

We will call the direction $\overrightarrow{W_1,W_2}$ optimal, if $W_2=\underset{W\in\Omega_{W_1}}{\arg\max}\,R_M(W)$. In other words, if W_2 maximizes the material removal rate. Evaluating vectors $\overrightarrow{W_1,W_2}$ for each surface point by sampling and transferring them into the parametric domain (u,v) generates the VF of optimal directions $V(u,v)$.

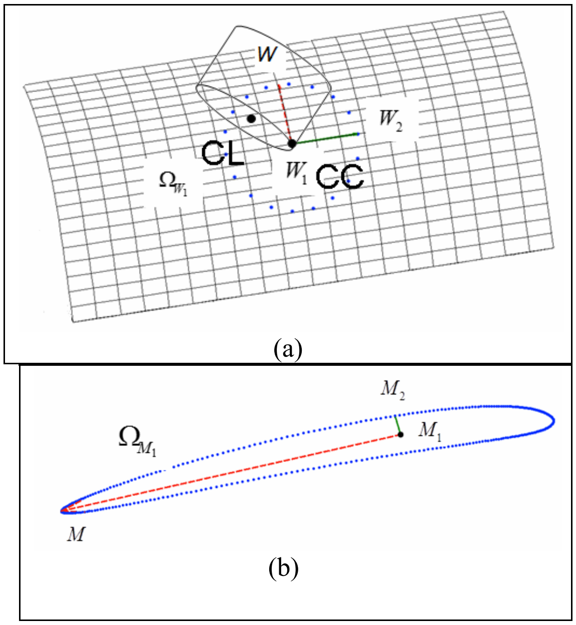


Figure 2 Possible tool feed directions: (a) in the workpiece coordinates (b) in the machine coordinates.

In order to reduce the machining time we maximize the utility function $R_M(W)$ and generate the VF of the optimal directions based on that particular but important criterion. Note that a variety of other objective functions related to the machine kinematics can be used to produce the required VFs. An example of such a VF based on $R_M(W)$ is given in Fig. 3.

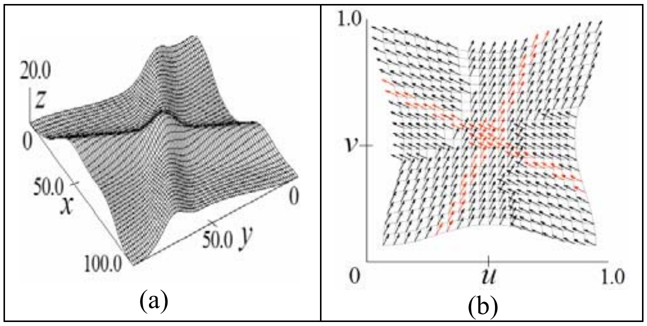


Figure 3.(a) A part surface $S(u,v)$, (b) the corresponding vector field $V(u,v)$.

III. GRID GENERATION

We arrange the CC-points $\{(u_{i,j},v_{i,j}),0\le i\le N_\xi$ $0\le j\le N_\eta\}$ as a curvilinear grid in the parametric domain (*u,v*). Mathematically, it means that the CC points are a discrete analogy of a mapping from the computational region $\Delta = \{0 \le \xi \le N_\xi,\ 0 \le \eta \le N_\eta\}$ onto a parametric region defined in the parametric coordinates (u,v). In other words, there exists a pair of functions $\{u(\xi,\eta), v(\xi,\eta)\}$ such that the rectangular grid $\{i, j\}$ being fed to $\{u(\xi,\eta), v(\xi,\eta)\}$ becomes $\{u_{i,j}, v_{i,j}\}$ (Fig. 4). $V(u,v)$ can be additionally partitioned into two VFs $(\alpha(u,v), \beta(u,v))$ (the dual VF) corresponding to the ξ and η directions as follows:

$$\alpha(u,v) = \begin{cases} V(u,v) \in \Omega_\xi, \\ 0, \text{otherwise}, \end{cases} \quad \beta(u,v) = \begin{cases} V(u,v) \in \Omega_\eta, \\ 0, \text{otherwise}, \end{cases}$$

where Ω_ξ, Ω_η are prescribed subsets of $V(u,v)$ selected according to a certain criteria (such as the direction, see Fig. 5) .

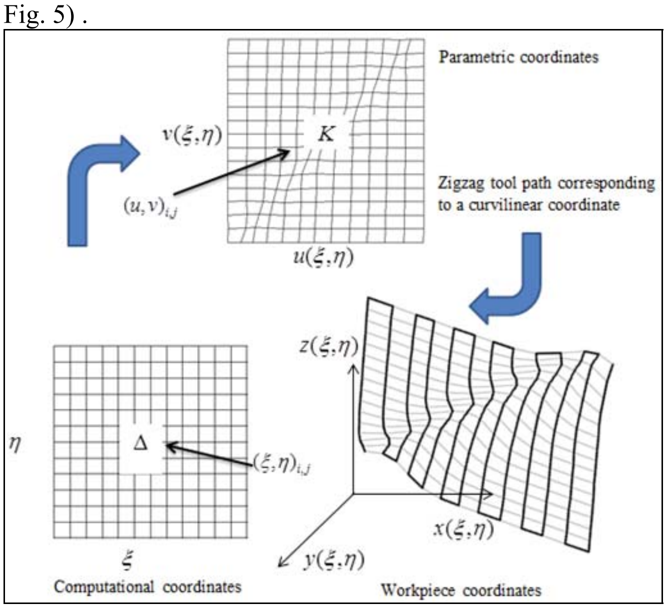
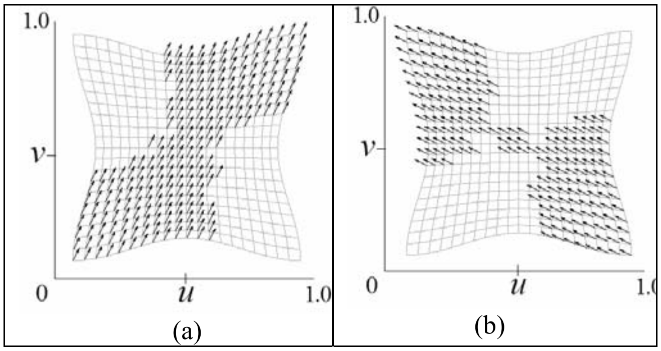


Figure 4. Coordinate transformations and the curvilinear grid, Δ - computational domain, K -parametric domain.



(a) (b)

Figure 5. The dual vector field : (a)- $\alpha(u,v)$, (b)- $\beta(u,v)$

Furthermore, the curvilinear grid $\{u(\xi,\eta), v(\xi,\eta)\}$ is aligned with the dual VF $(\alpha(u,v), \beta(u,v))$ using a modification of classical grid generation methods [6]-[8]. The smoothness of the grid is represented by a functional given by

$$F_S = \int\int u_\xi^2 + u_\eta^2 + v_\xi^2 + v_\eta^2 \mathrm{d}\xi\mathrm{d}\eta, \qquad (1)$$

where subscripts denote the partial derivatives.

Note that F_s is a classical variational functional providing smoothness of the mapping $\{u(\xi,\eta), v(\xi,\eta)\}$ [1]. The corresponding Euler equations for (1) are Laplacians able to offset the boundary due to their smoothing property. For example, if the boundary of the parametric region is a rectangle, functional (1) generates a rectangular grid corresponding to the conventional zigzag tool path. Furthermore, we show that a combination of the smoothness functional F_S and the VF alignment generates the required curvilinear tool path. For simplicity, consider alignment of the grid lines η=const to a VF $\alpha(\xi,\eta) \equiv (\alpha_1(\xi,\eta), \alpha_2(\xi,\eta))$.

The alignment is provided by a functional given by

$$F_A = \int\int (s_\xi \alpha')^2 \mathrm{d}\xi\mathrm{d}\eta, \qquad (2)$$

where $\alpha'(\xi,\eta) \equiv (\alpha_1', \alpha_2') = (-\alpha_2, \alpha_1)$ is the VF perpendicular to $\alpha(\xi,\eta)$ and $s_\xi = (u_\xi, v_\xi)$ is the tangent to the grid line η=const . If s_ξ is parallel or antiparallel to α, then $F_A = 0$. Following [5], the functionals F_S and F_A are combined linearly: $\Phi = F_S + \lambda F_A$, where λ is a weighting coefficient evaluated experimentally.

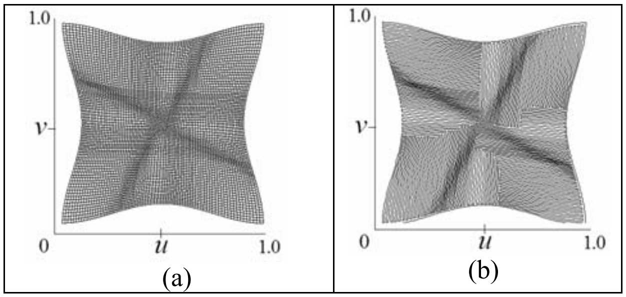
The corresponding Euler equations are

$$\Phi_u - \Phi_{\xi,u_\xi} - \Phi_{\eta,u_\eta} = 0,$$
$$\Phi_v - \Phi_{\xi,v_\xi} - \Phi_{\eta,v_\eta} = 0. \qquad (3)$$

The numerical solution of system (3) is based on the discrete Laplacian, the central differences for the first derivatives and numerical iterations. The corresponding finite-difference equations are solved by the Newton method.

IV. NUMERICAL EXAMPLES AND CUTTING EXPERIMENTS

The proposed VF based method has been combined with the ASFC [4] and tested against the iso-parametric zigzag tool path (ISO) for multiple convex-concave parts. We also test the method against "Follow Periphery" (UG-FP) and "Helical or Spiral"(UG-HS) options of Unigraphics NX9. The test surfaces were initialized in a MasterCam environment using the parametric representation. Next, they were exported into the STEP or IGES formats and imported into the UG. This section presents an introductory example for a surface in Fig. 3(a). The curvilinear grid adapted to the dual VF in Fig. 5 is shown in Fig. 6(a) and the corresponding ASFC tool path in Fig. 6 (b).



(a) (b)

Figure 6 (a)- Adaptive grid, (b) –ASFC tool path

The surfaces produced by the conventional ISO zigzag and by the proposed method are displayed in Fig. 7 (a) and (b) respectively. The ISO path creates inaccuracies near the edge, whereas the proposed method generates a smooth surface.

Table 1 shows the machining time. The proposed VF method outperforms the above algorithms for every prescribed scallop height, h_0. For instance, for h_0=0.25mm the VF tool path is 6.7 times faster than UG-HS, 3.2 times faster than UG-FP and 1.2 times faster than ISO.

TABLE I. MACHININIG TIME

Method		**Scallop height, mm**		
		0.25	0.1	0.05
UG	HS	2:59:36	7:10:23	14:54:49
	FP	1:27:47	2:18:48	3:16:02
ISO		0:33:49	0:53:10	1:15:04
VF method		**0:26:43**	**0:41:56**	**0:56:26**

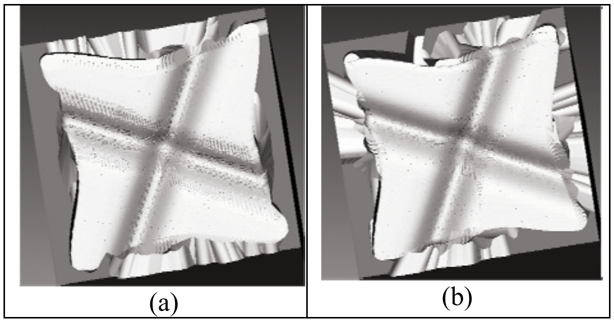

(a) (b)

Figure 7. Machining results, (a) ISO zigzag, (b) VF method

V. DECOMPOSITION OF THE VECTOR FIELD

Several algorithms have been proposed to decompose the part surface for tool path generation [11]-[13]. However, the algorithms are not applicable to the complex STL surfaces since the corresponding VF is usually irregular and noisy. In this section we present a new clustering approach based on invariant moments to circumvent the above drawbacks. The basic steps of the algorithm are as follows:

1) Flatten the STL surface using the conformal mapping [14]. 2) Construct the VF on 3D mesh and map it onto the 2D mesh. 3) Cluster the 2D VF using the moment invariants[15]-[16] 4) Construct the tool path for each cluster using a transfinite interpolation [17].

We convert the VF into an angle between $V(u,v) \equiv (v_x(u,v), v_y(u,v))$ and the u axis as follows:

$$\theta(x,y)=\begin{cases}\arctan\left(\dfrac{v_y}{v_x}\right), \text{if } v_x \neq 0,\\[2ex] -\dfrac{\pi}{2}, \text{otherwise.}\end{cases}$$

defined in a local coordinate system with the origin at the CC point $(u_{i,j}, v_{i,j})$.

A complex moment of the order, $p+q$ is defined by:

$$c_{pq} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} (u+iv)^p (u-iv)^q f(u,v)\,du\,dv$$

In the polar form

$$c_{pq} = \int_0^{\infty}\int_0^{2\pi} r^{p+q+1} e^{i(p-q)\varphi} f(r,\varphi)\,dr\,d\varphi \qquad (5)$$

where r, φ are the polar coordinates.

Clearly, $\theta(u,v)=\theta(-u,-v)$ and $\theta(-u,v)=\theta(u,-v)$. In other words, $f(x,y)=e^{i\theta(x,y)}$ is "flip invariant" and represent the orientation field (rather than a uni-directional vector field). Further, it is not hard to demonstrate that any moment constructed as

$$\prod_{j=1}^{n} c_{p_j q_j} \text{ with } \sum_{j=1}^{n}(p_j - q_j + 1) = 0 \qquad (6)$$

is rotationally invariant. The normalized moments (5)-(6) are also translation and scale invariant [15]-[16]. The most important information is stored in the low-order moments. We consider the following set[16]:

$$B = \{c_{01}, c_{00}c_{02}, c_{11}c_{02}, c_{10}c_{02}^2, c_{20}c_{02}^3\}.$$

$V(u,v)$ is obtained from an STL file by flattening and linear interpolation on the corresponding 2D triangles. Define a rectangular grid (u_i, v_j) which covers the parametric domain and interpolate $S(u,v)$ on that grid. Define a moving square window centered at (i,j) with the size $k \times k$. For each position (i,j) we evaluate a sequence of moments $B_{i,j}^k$ corresponding to $V_{i,j} \equiv V(u_i, v_j)$. The increasing size of the window $k = 3,5,\ldots,k_{\max}$ creates a pyramid structure. Note that since the window is not circular, the rotation invariance is hold only approximately, however, we consider square windows for computational convenience, whereas our numerical experiments show that the numerical error is negligible and clustering is exactly the same as in the case of circular windows.

Our clustering algorithm is based on a set of standard configurations S_p for which the invariant moments are calculated. The clustering algorithm is given below:

Input. 1) a 2D VF 2) prototype moments S_p corresponding to the standard patterns.

1. Calculate the pyramid moments $B_{i,j}^k$.

2. For each prototype S_p and (i,j,k) find $B_{i,j}^{k,p}$ such that $\| S_p - B_{i,j}^{k,p} \| < \delta$, where δ is the threshold.

3. Sort $B_{i,j}^{k,p}$ by the size k.

4. If a point (i,j) belongs to two or more different clusters attach this point to the largest k. If a point belongs to several clusters with the same size, attach the point to the cluster with the smallest $\| S_p - B_{i,j}^{k,p} \|$.

4. Generate super-clusters. If clusters C_1 and C_2 have a common boundary and the vectors at the boundary are nearly perpendicular, replace all angles in C_2 by $\theta+\pi/2$. Form a cluster $C_{12}=C_1\cup C_2$ (Fig. 8). Proceed until there is no possibility to create the super-clusters.

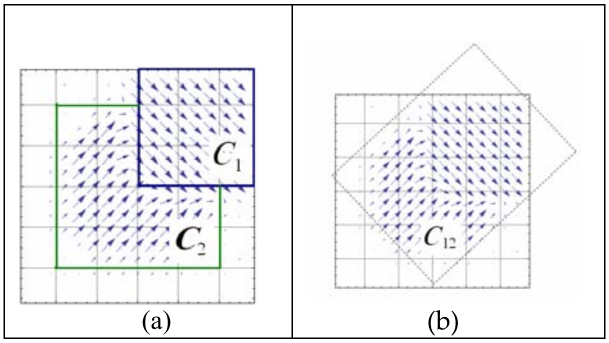
5. Process points which do not belong to any cluster. Include the remaining points into the closest clusters using the K-means or similar techniques.

6. Detect the boundary of each cluster using the chain code algorithm.

7. Smooth the boundary using the cubic splines.

8. Match each cluster with the prototype region and generate the corresponding curvilinear path.

Output. A set of clusters $C'_{n,p}$ each corresponding to a certain template S_p, Curvilinear tool paths G_p.



(a) (b)

Figure 8. Clusters C_1 and C_2 with nearly perpendicular directions at a joint boundary (b) super-cluster C_{12}

VI. Tool path generation. Numerical Example.

Three grid prototypes: the H-grid, the O-grid and the C-grid are depicted in Fig. 9 (a) along with the corresponding prototype VFs in Fig. 9 (b).).

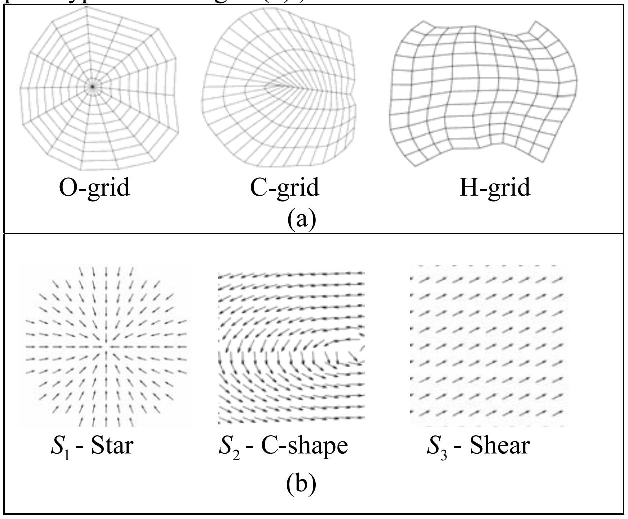


Figure 9. (a) Prototype grids, (b) prototype vector fields

Consider a test STL surface in Fig. 10(a) and a flattened surface in Fig. 10(b). The corresponding VF $V(u,v)$ based on the maximization of material removal rate $R_M(W)$ is shown in Fig. 10 (c) along with the corresponding clusters. The bounding boxes of the clusters are shown in Fig. 10 (d).

In each sub-region the required grid is generated using the transfinite interpolation [17]. Our version of the transfinite method assumes that the region is a curvilinear rectangle and constructs the grid topologically equivalent to a Cartesian grid in the computational region (see Fig. 4). However, points outside the bounding box (rectangular for the H and the C-grid, and circular for the O-grid) are discarded.

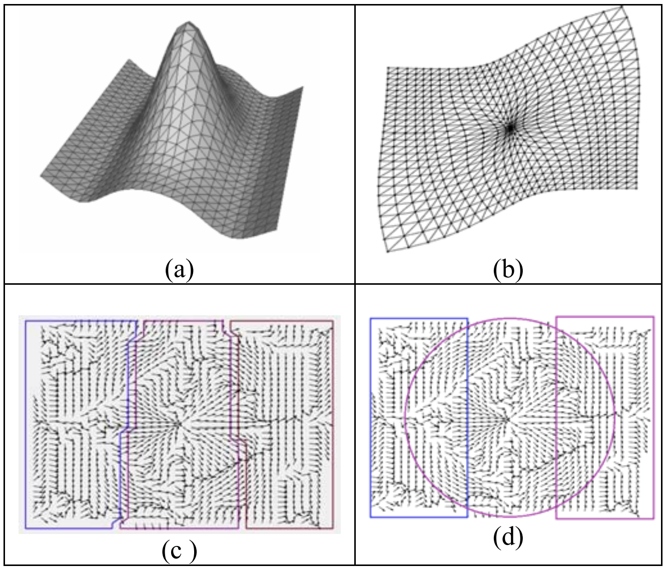

(a) (b) (c) (d)

Figure 10. (a) A test STL surface, (b) flattened surface, (c) vector field and clusters, (d) bounding box grids

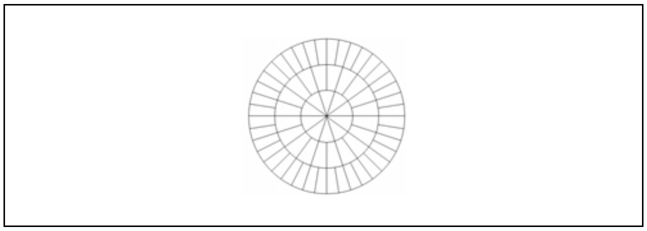


Figure 11. Radial-zigzag tool path

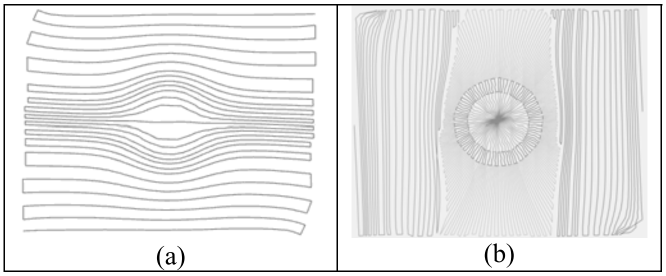

(a) (b)

Figure 12. (a) ISO path (b) proposed method

Note that the maximum allowed scallop height h_0 between the tool tracks defines the machining strip and the number of the required tracks for the ball-nose, toroidal and flat cutter. Further, it is often the case that the optimal directions generate redundant "star" patterns. We replace such patterns by a combined radial/circular zigzag pattern displayed in Fig. 11. Figs 12 (a) and (b) show the ISO parametric tool path and the tool path generated using the proposed decomposition of a test surface. Fig. 13 shows the machining results.

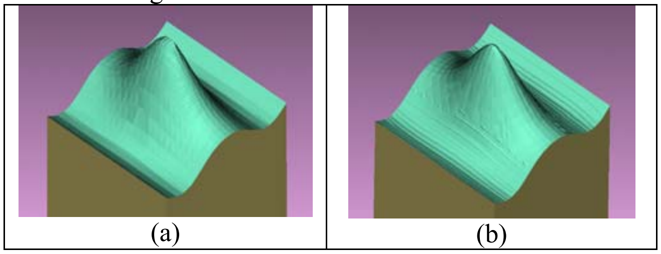


Figure 13. Virtual milling surface (a) ISO path (b) the proposed VF method

The machining time is evaluated by

$$T = \sum_{i=0}^{N-1} t_{i+1/2} \text{ , where}$$

$$t_{i+1/2} = \max(\frac{X_{i+1}-X_i}{v_L}, \frac{Y_{i+1}-Y_i}{v_L}, \frac{Z_{i+1}-Z_i}{v_L}, \frac{A_{i+1}-A_i}{v_R}, \frac{B_{i+1}-B_i}{v_R}),$$

where (X_i, Y_i, Z_i) is the machine coordinates of the cutter location point i, A_i, B_i the rotation angles, v_L and v_R are the maximum speed of the linear and rotary axis respectively.

Table II compares the machining time obtained using the ISO parametric method, the contour parallel method (slicing) and the proposed VF clustering approach. The VF method performs better because it follows the direction, where $R_M(W)$ is maximal. Thus, on every step the tool removes more material than the conventional methods. In turn it makes it possible to increase the step-over between the tracks and, therefore, reduce the machining time.

TABLE II. MACHININIG TIME (MIN:SEC)

Method	**Scallop, mm**		
	0.1	0.05	0.01
Contour method	6:15	7:28	17:13
ISO	4:55	6:01	17:35
VF method	**3:10**	**3:50**	**6:54**

VII. CONCLUSIONS

A new method for generation of VF-aligned tool paths for five-axis machining has been presented and analyzed. The new idea is the numerical generation of a curvilinear tool path adapted to the VF of optimal directions. This is combined with clustering of the vector field and generation of local optimal sub grids providing faster machining. The experiments show a substantial decrease in the machining time with regard to preceding methods.

REFERENCES

[1] C.-J Chiou, Y-S. Lee, "A machining potential field approach to tool path generation for multi-axis sculptured surface machining," *Comp. Aided Des.*, vol. 34, no. 5, pp. 357-371, 2002.

[2] X. Liu, Y. Li, S.Ma, C.-H. Lee "A tool path generation method for freeform surface machining by introducing the tensor property of machining strip width," *Comp. Aided Des.*, vol. 66, pp. 1-13, 2015.

[3] G.H. Kumazawa, H.Y. Feng, M.J.B. Fard, "Preferred feed direction field: A new tool path generation method for efficient sculptured surface machining," *Comp. Aided Des.*, vol. 67, pp. 1-12, 2015.

[4] S. Moodleah, E.J. Bohez, S.S. Makhanov, "Five-axis machining of STL surfaces by adaptive curvilinear toolpaths," *Int. J.Prod. Res.,* vol. 54, no. 24, pp. 7296-7329, 2016.

[5] Y.W. Sun, D.M. Guo, Z.Y Jia, H.X. Wang, "Iso-parametric tool path generation from triangular meshes for free-form surface machining," *Int.J Adv. Manuf. Tech.*", vol. 28, pp. 721-726, 2006.

[6] A.E. Giannakopoulos, A.J. Engel, "Directional control in grid generation," *J. of Comp. Phys.*, vol. 74, no. 2, pp. 422-439, 1988.

[7] V.D. Liseikin, "Grid generation methods," Springer, 2010.

[8] J.E. Castillo, J.S. Otto," Numerical techniques for the transformation to an orthogonal coordinate system aligned with a vector field," *Comp. & Math. with Appl.*, vol. 40, no. 4-5 , pp. 523-535, 2000.

[9] W. Anotaipaiboon, S.S. Makhanov, "Tool path generation for five-axis NC machining using adaptive space-filling curves," *Int. J. Prod. Res.*, vol. 43, no. 8, pp.1643-1665, 2005.

[10] S.S. Makhanov, W. Anotaipaiboon, "Advanced numerical methods to optimize cutting operations of five-axis milling machines," Springer 2007.

[11] N.V. Tuong, P. Pokorný, "A practical approach for partitioning free-form surfaces," *Int.J. Comp. Integr. Manuf.*, vol. 23, no.11, pp. 992-1001, 2010.

[12] A. Roman, S. Bedi, F. Ismail. "Three-half and half-axis patch-by-patch NC machining of sculptured surfaces," *Int. J. Adv. Manuf. Tech*., vol. 29, no. 5–6, pp. 524–531, 2006.

[13] S. Ding, M.A. Mannan, A.N. Poo, D.C.H Yang,, Z. Han "The implementation of adaptive isoplanar tool path generation for the machining of free-form surfaces," *Int.J.Adv.Manuf. Techn*., vol. 6, no. 7–8, pp. 852–860, 2005.

[14] A. Sheffer , B. Lévy, M. Mogilnitsky, A. Bogomyakov "ABF++ : fast and robust angle based flattening," *ACM Trans. on Graphics*, vol. 24, no.2 , pp.311-330, 2004.

[15] J. Flusser "On the independence of rotation moment invariants," *Pattern Recogn.* vol. 33, pp.1405–1410, 2000.

[16] J. Flusser "On the inverse problem of rotation moment invariants," *Pattern Recogn*. vol. 35, pp. 3015–3017, 2002

[17] W.J. Gordon, L.C.Thiel "Transfinite mappings and their application to grid generation," *Appl. Math. and Comp*. vol. 10–11, pp. 171-233, 1982.