Research Project SCAN3D_CNC in the Robotics & AI Laboratory, University Politehnica of Bucharest

Project CNCSIS IDEI 69

Generating CNC toolpaths from grey level image processing and pattern robot motion

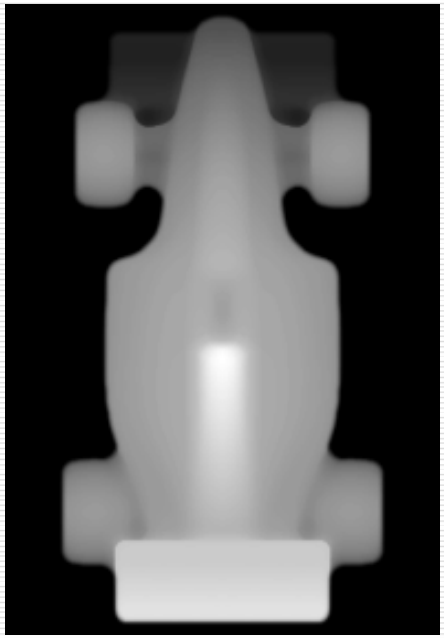
University Politehnica of Bucharest, AC&CS Faculty, Centre of Research CIMR, cimr@cimr.pub.ro

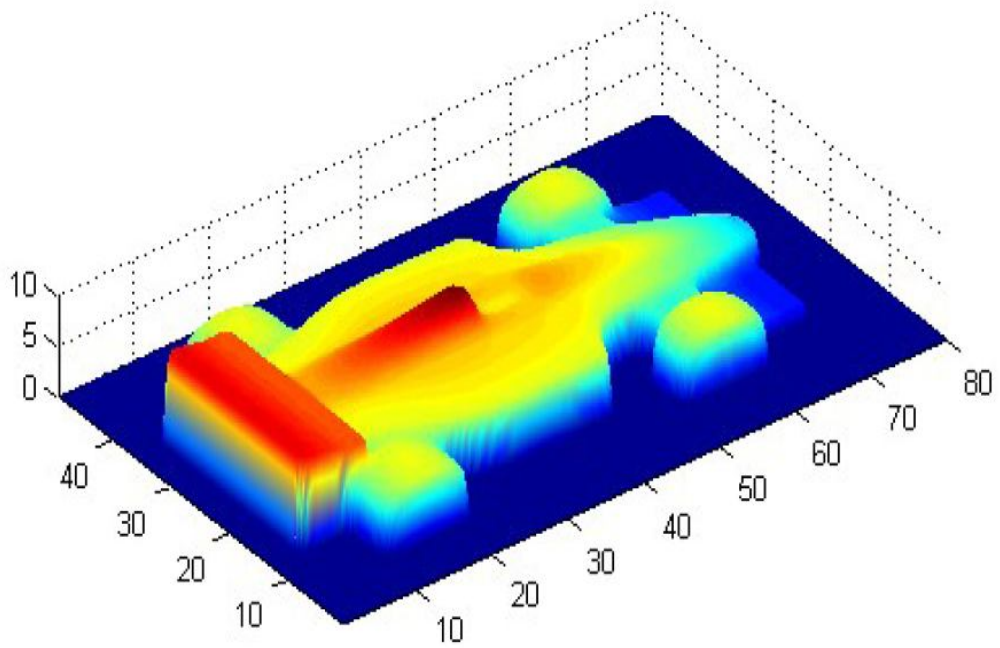
Summary: Generating CNC toolpaths from grey level image & pattern robot motion

- Height Map images
- Hand-held laser range finder and robot motion patterns
  - Laser – robot calibration
  - Rotary table – robot calibration
  - Real time computation of adaptive robot paths for collision-free motion
- Modelling the machining surface and tool shape
- Performing tool compensation
- Generating roughing & finishing toolpaths
- Error analysis
- Future developments

Generating CNC toolpaths from grey level image and pattern robot motion

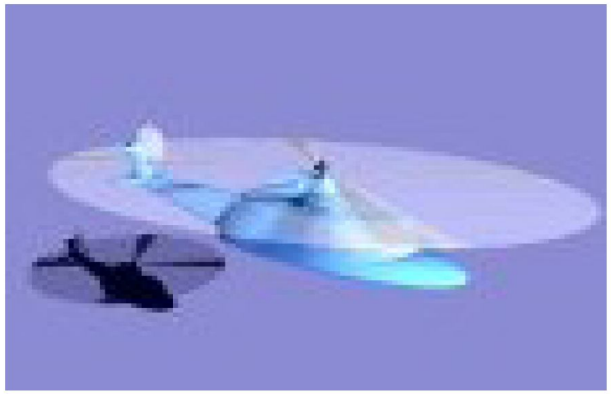
- **Height Map images**



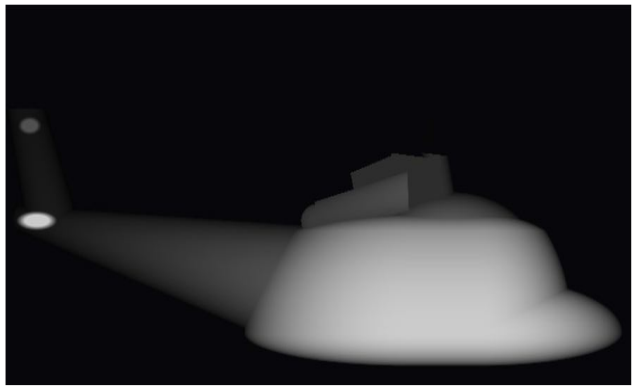


Generating CNC toolpaths from grey level image and pattern robot motion

❑ Obtaining height map images



3D Model in POV-Ray



Height Map Model

1) Remove light sources
2) Remove material textures
3) Use an ortographic camera
4) Apply a pigment with luminance proportional to the distance from the camera plane:
   - ☐ farthest point: pure black (luminance 0)
   - ☐ closest point: pure white (luminance 1)

Generating CNC toolpaths from grey level image and pattern robot motion

❑ Obtaining height map images

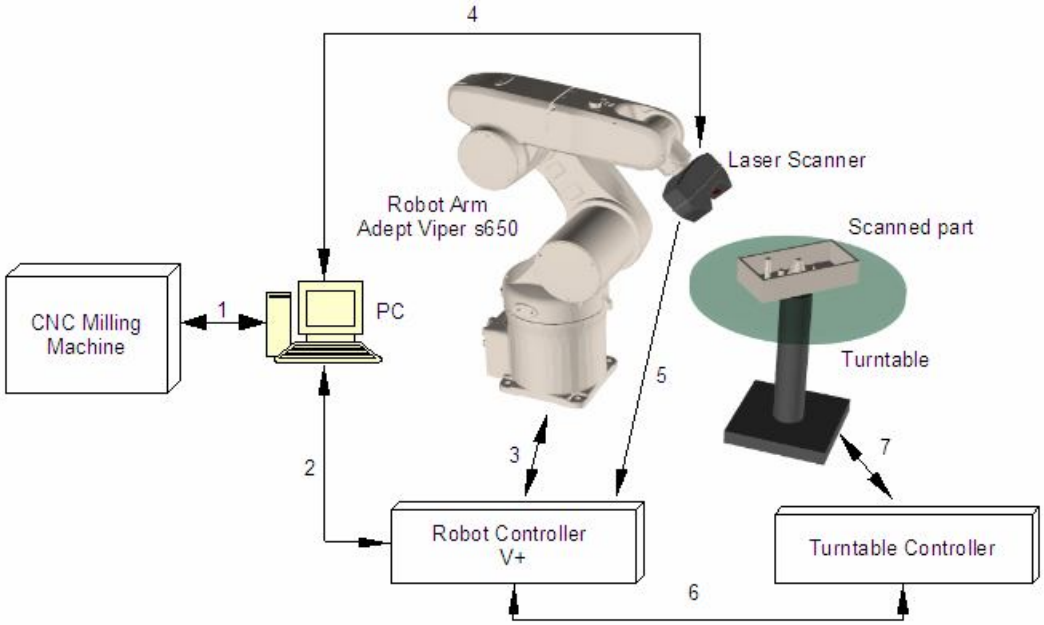
The 3D surface of a model to be machined is scanned with a laser range finder device:

- A vertical stripe of laser light is moved across the model object surface, and captured by a video camera. Along each horizontal scan line of the video frame, the *brightest* spot is taken to be the point at which the laser stripe "hits“ the surface (detection at sub-pixel resolution).
- The relative positions of the laser and the video camera are used to find the 3D coordinates of the brightest spot by triangulation.
- The *x*-coordinate of each point in the output depth image is determined by the position of the laser stripe for a particular video frame.
- The *y*-coordinate corresponds to a raster line in the video frame.
- The depth value is computed from the brightness peak detected along the raster line in the video frame.

Generating CNC toolpaths from grey level image and pattern robot motion

❑ Arm-mounted laser range finder and robot motion patterns

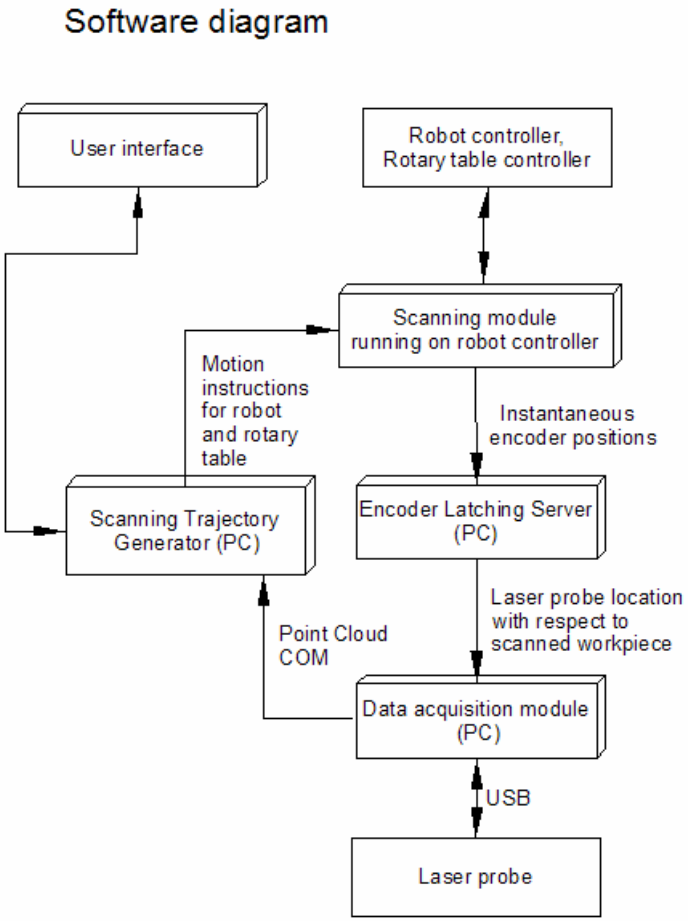
- Dual laser probe measure distances from 70 to 250 millimetres, with an accuracy of 30 µm
- The laser probe is arm-mounted on a 6-d.o.f. robot
- The scanning paths are computed in *real-time* by the robot controller from *predefined* or adaptive motion patterns
- The range finder device generates depth map-type information describing the object's surface, *synchronously* with the motion of the laser scanner probe
- Robot working envelope: spherical, 650 mm radius; resolution of rotary table: 0.03 deg
- Hardware controllers: robot-, rotary table-, CNC machine; PC integrated



Generating CNC toolpaths from grey level image and pattern robot motion

❑ Arm-mounted laser range finder and robot motion patterns

- A linear laser module projects a line of red light on the scanned object
- The line is detected by two cameras located on the laser probe
- The object's contour along the laser line is measured using Cartesian coordinates
- The robot arm and rotary table form a **7-DOF kinematic chain** that can move the laser probe to a precise location relative to the work piece, according to the *scanning trajectory*; from this location, a *measurement* is made
- The measured points *lie* in the laser plane
- Since the position of the laser probe (and therefore the laser plane) is known *relative to the work piece* at each measurement point, the measured points can be transformed into a *unique* reference frame, which is attached to the work piece
- In this way, a point cloud is obtained, which is the *3D representation of the scanned object*



Software diagram of the laser scanning system

Generating CNC toolpaths from grey level image and pattern robot motion

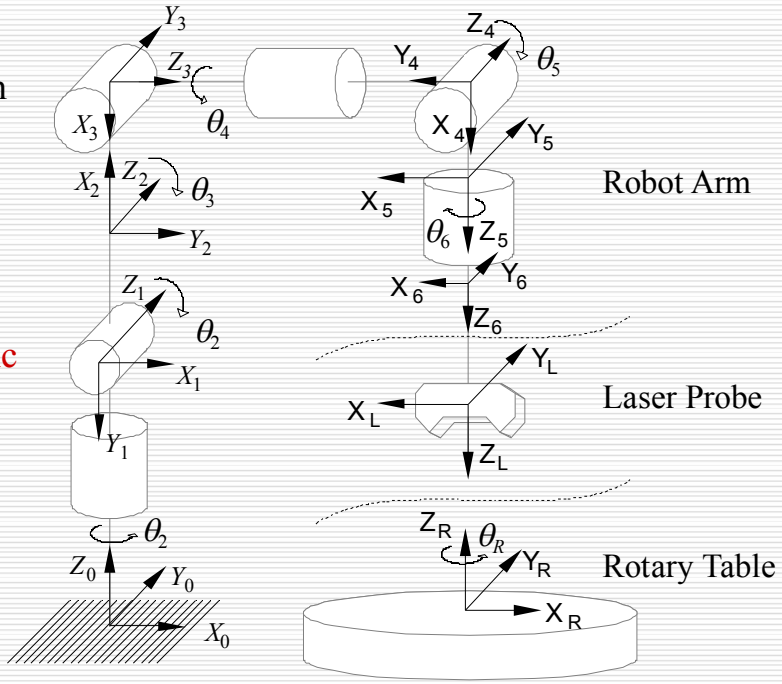
❑ Arm-mounted laser range finder and robot motion patterns

➢ Communication

- **USB interface** connects laser probe to the PC
- **Data acquisition module** (DAM) takes measurements from laser probe
- Measurements are aligned in the *workpiece's frame*; the instantaneous pose of laser probe relative to rotary table is computed by the **Encoder latching server** (ELS) module
- ELS receives the *encoder readings* from robot- and rotary table controllers (TCP/IP connection) and uses the kinematic model of the system
- The location of the laser probe is sent to the DAM in **X-Y-Z-yaw-pitch-roll** format

➢ Synchronization

- The robot- and rotary table controllers latch their current position using an external trigger signal sent by the laser probe every time a measurement is made
- Scanning rate: 50-150 frames / sec (frame = line of scanned points; robot / table position latching: less than 1 msec



Denavit-Hartenberg reference frame assignment

Generating CNC toolpaths from grey level image and pattern robot motion

❑ Arm-mounted laser range finder and robot motion patterns

➢ **Communication (con't)**

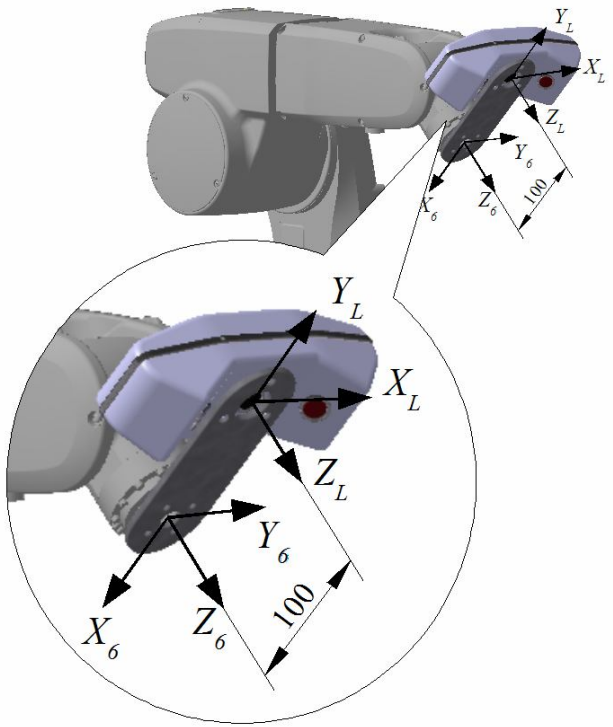
- The Scanning module (SM), running on the RC, detects the trigger signal; gets the current position of the rotary table; appends it to the current positions of the 6 joints (sent through TCP/IP link)
- The **Scanning Trajectory Generator** (STG) computes predefined or *adaptive scanning paths*, based on part shape and approx. size T

➢ **Robot – laser probe calibration**

- The *transformation* from the 6th link of the robot to the laser probe reference frame is $T_L^6 = \mathcal{R}_Z(90°) \cdot \mathcal{T}(0, -100, 0)$
- An exact expression of the transformation *compensates mechanical errors* is computed in a custom-design calibration procedure

➢ **Robot – rotary table calibration**

- The transformation from the rotary table to the robot base is $T_0^R(\theta_R) = \mathcal{T}(500, 0, 200) \cdot \mathcal{R}_Z(\theta_R)$. It is calibrated with **HPS cal program**
- Measurements from the laser probe, which are in the $X_L Y_L Z_L$ frame, are aligned to the object frame (rotary table frame) $X_R Y_R Z_R$ by pre-multiplying all laser probe measurements with the **alignment trans** $T_{align} = T_L^R = T_0^R(\theta_R) \cdot T_6^0(\theta_{1\ldots6}) \cdot T_6^L$



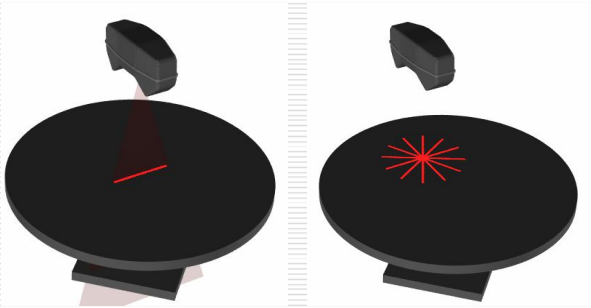
Robot – Laser Probe transformation

Generating CNC toolpaths from grey level image and pattern robot motion

❑ Arm-mounted laser range finder and robot motion patterns

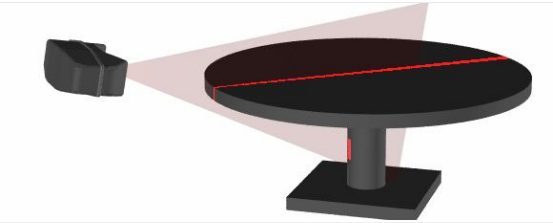
➢ **Robot – rotary table calibration (con't)**

- Alignment test (AL): verifies the alignment between the vector normal to the table surface and the Z_L axis of the laser probe:



Laser probe locations for alignment test

- Offset test (OT): AL The is repeated for various angles around the table, and the measurements should remain the same for a correct table centre
- Eccentricity test (ET): verifies the compensation for internal mechanical errors (*eccentricity* and *internal tilt*) of the table



Laser probe location for eccentricity test

➢ **Real-time computing adaptive robot scanning paths**

- **STG input**: *scanning toolpaths* – a series of locations in the object's reference frame in which the robot and table should be synchronously moved to allow the laser probe to take measurements
- **STG output**: the sequence of robot joint values and rotary table angle providing the desired laser probe location relative to the object (*7-DOF mechanism IK*)
- **Secondary objectives**:
  - *limit* rotary *table speed* and *minimize accelerations*
  - *avoid singularities* and *collisions* between the manipulator and the rotary table

Generating CNC toolpaths from grey level image and pattern robot motion

❑ Arm-mounted laser range finder and robot motion patterns

➢ Kinematic model

- T_L ; $T_L^{(k)}$: location of laser probe (LP) relative to object; trajectory of LP at discrete time k
- $T_M^{(k)}$: LP location relative to manipulator (M) base
- $T_M^{(k)} = T_{DK}\left(\theta_1^{(k)} \ldots \theta_6^{(k)}\right) \cdot T_{TL}$, T_{TL}: *LP* loc to TMF
- $T_R^{(k)}\left(\theta_R^k\right) = \mathcal{T}(500,0,200) \cdot \mathcal{R}_Z\left(\theta_R^k\right)$: loc of rotary table (RT) relative to M base
- Inverse Kinematics: $\theta_{1\ldots6}^{(k)} = IK_6\left(T_{DK}^{(k)}\right)$

➢ Problem decomposition

- Solution of IK problem computed in 2 steps:
  1. *Choose a suitable RT angle* (*ang_rt*)
  2. *Solve IK* for 6-d.of. manipulator AND *ang_rt*
- From (1), LP loc to M: $T_M^{(k)}\left(\theta_R^k\right) = T_R^{(k)}\left(\theta_R^k\right) \cdot T_L^{(k)}$
- Solve IK: $\theta_{1\ldots6}^{(k)} = IK_6\left(T_M^{(k)}\left(\theta_R^k\right) \cdot \left(T_{TL}\right)^{-1}\right)$ for a M configuration flags choice: R/L, A/B, F/NF

- The 7-d.o.f. planning problem reduces to planning $\theta_R^{(k)}$ for all time steps $k = \overline{1,n}$, such that there exists at least 1 solution for IK
- Strategy: modify RT angle in fixed increments (1 deg) until the IK problem has 1 solution

➢ Configuration space and map

- Configuration space: 2D set with:
  - X axis: discrete time $k = \overline{1,n}$
  - Y axis: RT angle θ_R ranging from -180^o to 180^o
- Configuration map: 2D discrete image in config. Space

➢ Binary configuration map M_B

- Pixel value at location (i, j): time step j, RT angle $\theta_R(i)$

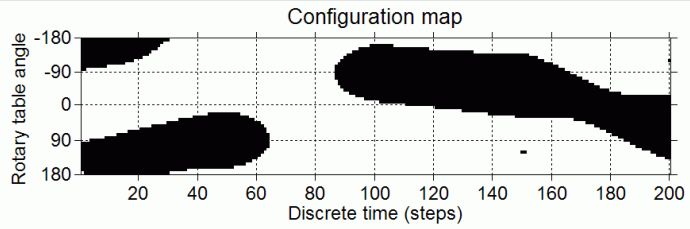
$$M_B(i,j) = \begin{cases} 0\,(\text{black}),\ IK_6 \text{ has no solution for } T_L^{(j)} \text{ and } \theta_R^{map}(i) \\ 1\,(\text{white}),\ IK_6 \text{ has at least one solution} \end{cases}$$

- Allowed (white) regions on map: C_{free} : open set
- Forbidden (black) regions on map: C_{obs} : closed set

Generating CNC toolpaths from grey level image and pattern robot motion

❑ Arm-mounted laser range finder and robot motion patterns

➢ Binary configuration map (con't)



Example of binary configuration map.

- Motion trajectories: the configuration map only depends on the ser of desired trajectories $T_L^{(j)}$
- Once defined the configuration map, find a path through "**obstacles**" from the starting RT angle (j = 1) to any final position (j = n)

➢ Grey level configuration map

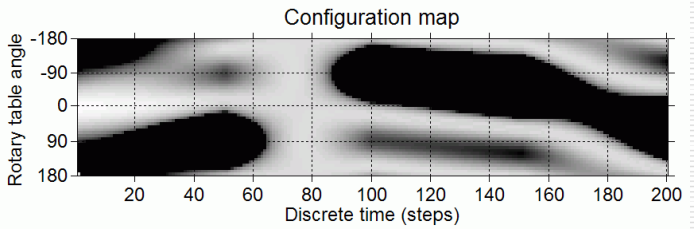
- Objective: provide LP paths that keep a sufficient distance to "obstacles", i.e.:
  - not close to manipulator *joint limits*;
  - not close to manipulator *singular points*

- Values of configuration map: in (0, 1]; any of these states are allowed; shows how "desirable" are states
  - C_{obs} states remain 0;
  - C_{free}: the higher a pixel value, the "best" the state
  - joint limits for link j: $\theta_j^{\min} \le \theta_j \le \theta_j^{\max}$
  - function $f(\ )$ equal to zero at the joint limits, reaches its maximum value at the middle of the interval:

$$f_j(\theta_j) = \left(\sin\left(\frac{\theta_j - \theta_j^{\min}}{\theta_j^{\max} - \theta_j^{\min}} \right) \right)^{\gamma_j}, \quad f_R(\theta_{1\cdots 6}) = \prod_{j=1}^{6} f_j(\theta_j)$$

- Extended configuration map M_G: *grey level image*:

$$M_G(i,j) = \begin{cases} 0, & IK_6 \text{ has no solution for } T_L^{(j)} \text{ and } \theta_R(i) \\ f_R(\theta_{1\ldots 6}), & IK_6 \text{ has 1 solution, } \theta_{1\ldots 6} \end{cases}$$



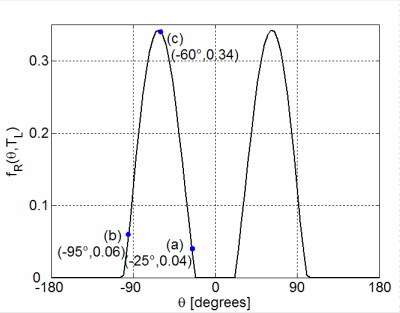
Example of grey level configuration map

Generating CNC toolpaths from grey level image and pattern robot motion

❑ Arm-mounted laser range finder and robot motion patterns

➢ Path planning principles

- **Search criteria** for robot (LP) paths on M_G map:
  - *primary*: the value function $f_R(\theta_{1\ldots6})$, *not touch the configuration space obstacles*
  - *secondary*: *cost of path* (length, smoothness)



Robot arm and laser looking at a work piece from a given pose T_L

(a) Robot is near a "too close" condition (table rotated at -25°)
(b) Robot is near a "too far" condition (table rotated at -95°)
(c) Robot is not close to its limits (table rotated at -60°)
(d) Graph of $f_R(\theta,T_L)$ for $\theta=[-180,180]$ and T_L , with configurations from (a), (b) and (c) represented as points

- **Performance criteria**:
  - *minimize delays* in the scanning process (interlaced waits of robot manipulator and rotary table)
  - *limit high speed and accelerations* of the rotary table

Generating CNC toolpaths from grey level image and pattern robot motion

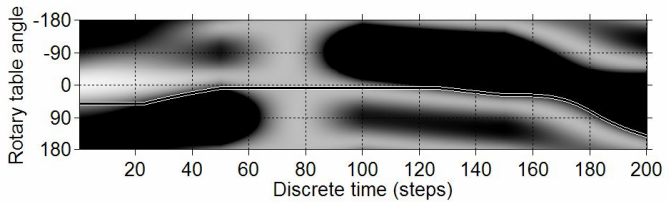
❑ Arm-mounted laser range finder and robot motion patterns

➢ **Path planning algorithms**

▪ **(1) Rotate when out of range**

− Whenever the trajectory hits an obstacle on the M_B, the algorithm *finds the minimum amount of rotation that gets out of the obstacle*

− The computed path will *touch the obstacles* (not desirable), and every time the table turns, the *LP must wait* until the computed rot angle is reached



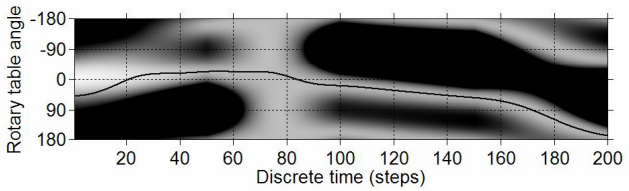
Solution of Algorithm 1. The planned path touches the edges of the obstacles

▪ **(2) Local maxima search**

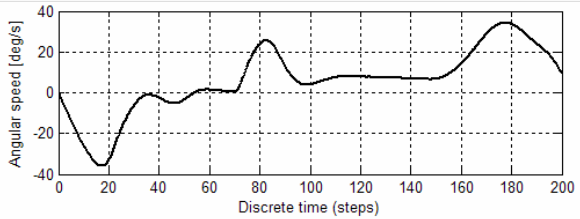
− A guide path *touching the local maxima of the value function* f_R is computed

− Improves strategy (1): the path does not touch the edges of obstacles

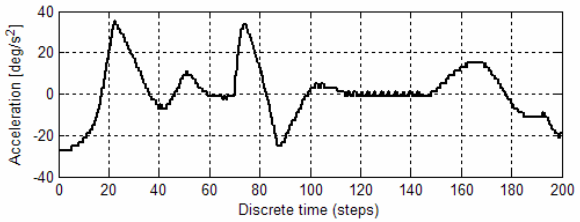
(a)



(b)



(c)



Smoothed local maxima path: (a) Position [deg]; (b) Angular speed [deg/s]; (c) Angular acceleration [deg/s^2]

Generating CNC toolpaths from grey level image and pattern robot motion

❑ Arm-mounted laser range finder and robot motion patterns

➢ **Path planning algorithms (con't)**

▪ **(3) Dijkstra-like algorithm**

– Finds a minimal cost path through a graph with positive weights (*costs*) on its edges, from a given node to all other notes reachable from it

– A 3rd dimension is added to the configuration space M_G: the angular velocity ω

– A node in the graph will be expressed as a vector of discrete coordinates (i, j, k), which maps to its continuous counterpart $(\theta_i, t_j, \omega_k)$

– From node $(\theta_i, t_j, \omega_k)$, one may *advance* using the acceleration and reach the node corresponding to $\left(\theta_i + \omega_k \Delta t + a\frac{(\Delta t)^2}{2},\ t_j + \Delta t,\ \omega_k + a\Delta t\right)$

– The **cost of a node** is:

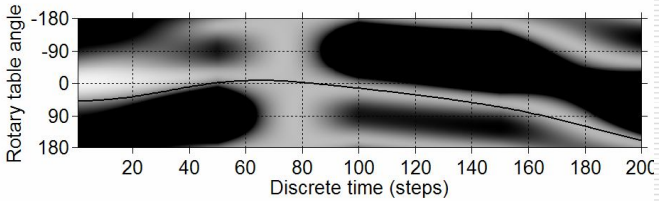
$$C_{node} = k_\omega (\omega_k)^2 + 1 - f_R\left(\theta_i, T_L^{(j)}\right)$$

– The **cost of an edge** is:

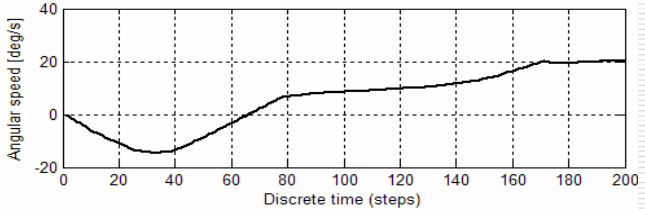
$$C_{edge} = k_a a^2 + k_\omega (\omega_k + a\Delta t)^2$$

– The planned path does not touch obstacle edges, the motion is smooth and the acceleration rates are much lower than those obtained with the Local Maxima (2)

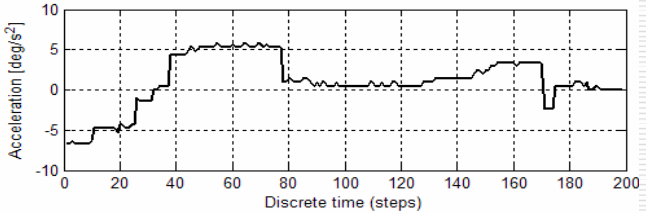
(a)



(b)



(c)



Path computed by Dijkstra algorithm for a grayscale M_G. The path does not touch obstacles. (a) Position; (b) Angular speed; (c) Angular acceleration

Generating CNC toolpaths from grey level image and pattern robot motion

❑ Arm-mounted laser range finder and robot motion patterns

➢ **Path planning algorithms (con't)**

▪ **(4) "Ray-Shooting" heuristic algorithm**

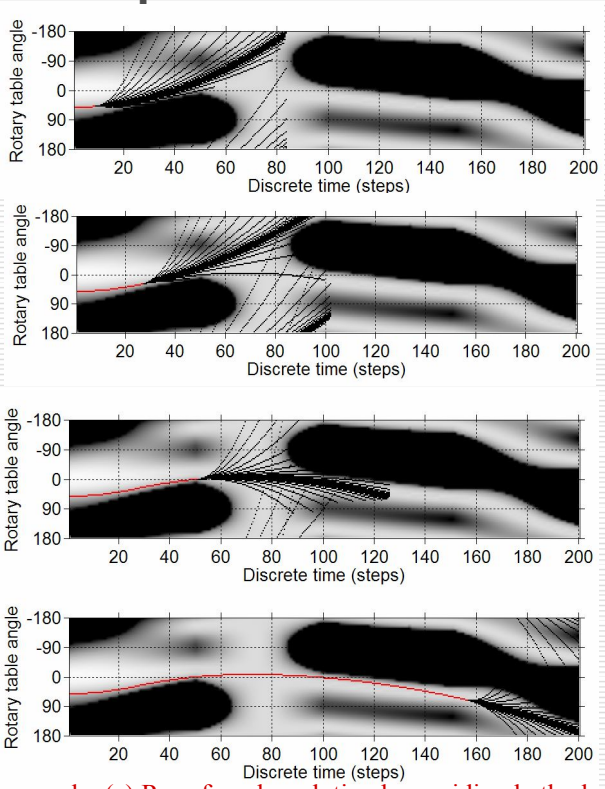
– Provides a better solution than the local maxima heuristic, with real time planning perspective

– Computes a smooth motion, comparable to the one obtained with Dijkstra algorithm

– At every time step k, the algorithm *looks ahead* p future time steps, that is, from $k+1$ through $k+p$. Over this range, performs a motion with constant angular acceleration, for planned path smoothness

– A finite set of acceleration values a_j, $j=\overline{1,n_a}$, and for every acceleration a_j, a possible path is evaluated, starting from the current state and spanning on the following p time steps.

– From the set of paths, the best one is chosen, for an acceleration $a_{j.max}$, and the motion from time k through time $k+1$ is done with this acceleration



Ray Shooting example: (a) Rays found a solution by avoiding both obstacles on the left side; (b) One ray starts seeing an alternative path: avoiding the 2nd obstacle on the right side; (c) The 2nd alternative has lower cost than the 1st one, thus it is chosen; (d) Search almost finished. Rays wrap around on *Y* axis.

Generating CNC toolpaths from grey level image and pattern robot motion

2.5D Surface Modelling

- Pixel *grey level* at (i, j) *encodes surface height* at (x,y)
- *Pixel-to-millimeter* ratio:
  - $x = R\ i$
  - $y = R\ j$
- Minimum Z of the surface: black pixel
- Maximum Z of the surface: white pixel
- Grey level value:
  - 8 bit integer: low precision, low storage space
  - 16 bit integer: good precision
  - Floating point: best precision, high storage space

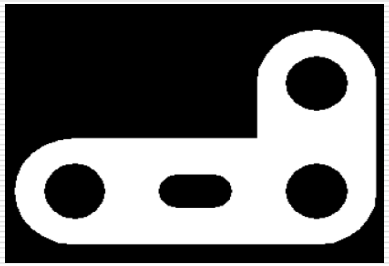
Generating CNC toolpaths from grey level image and pattern robot motion

2.5D Surface Modelling

Simplest Case: 2 Dimensions

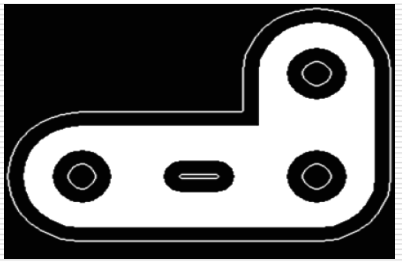
- Offsetting is equivalent to image dilation
- For efficiency, only *contour pixels* need to be considered
- Tool Path is generated by extracting the contour
- By image erosion, one obtains the machined shape

❖ Part model:



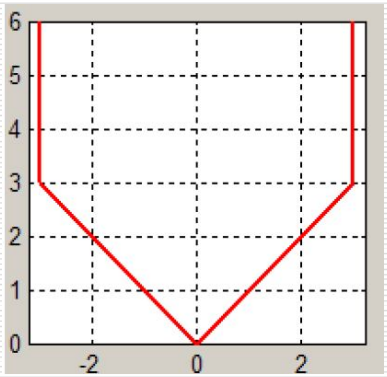
❖ Tool model:

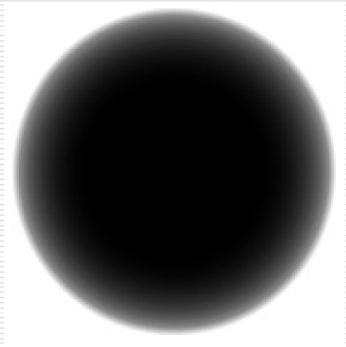
❖ Tool compensation:



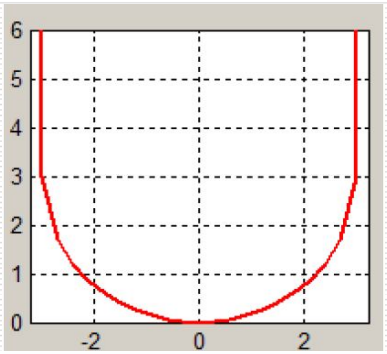
Generating CNC toolpaths from grey level image and pattern robot motion

❑ Tool Shape Modelling

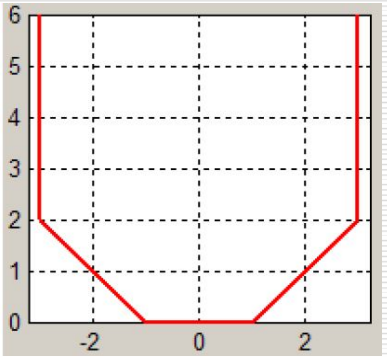




Conic Mill



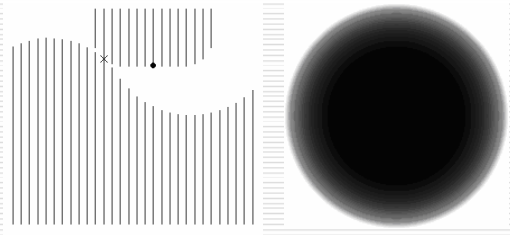
Round End Mill



Bull End Mill

Generating CNC toolpaths from grey level image and pattern robot motion

❑ Tool Compensation



The principle of discrete cutter compensation based on gradient computing in the 2D grey scale cutter shape image:

- **Objective**: Generating gouge-free tool paths
- **Idea**: For each (x,y) position, find the maximum depth at which the end mill can go down without cutting extra material
- **Algorithm**: Greyscale image dilation
- **Image**: Surface model
- **Structural element**: Tool model

Idea: at every location in the XY plane (i.e. any image pixel) the depth is computed which should be reached by the milling cutter, in order to be tangent to the surface.

The shape of the milling cutter was modeled as a grey scale image, using the same scale factors as for the surface to be milled.

Generating CNC toolpaths from grey level image and pattern robot motion

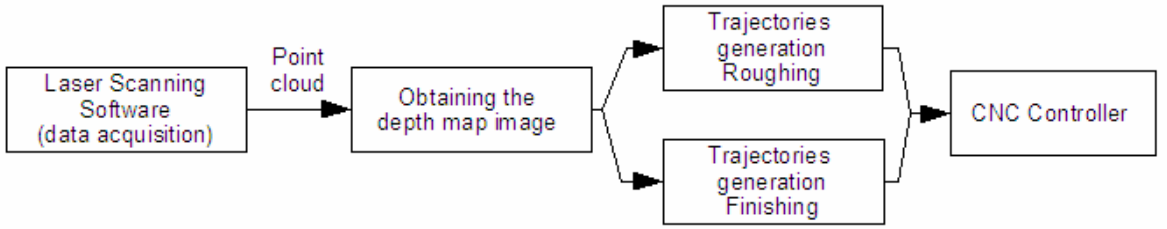
Tool Compensation Results

Advantages

- Gouge-free tool paths for many tool shapes
- Immediate generation of basic roughing and finishing tool paths
- Simple implementation, no need for complex 3D geometry computations

Disadvantages

- High computation time
- High storage space
- Compromise between precision and speed!

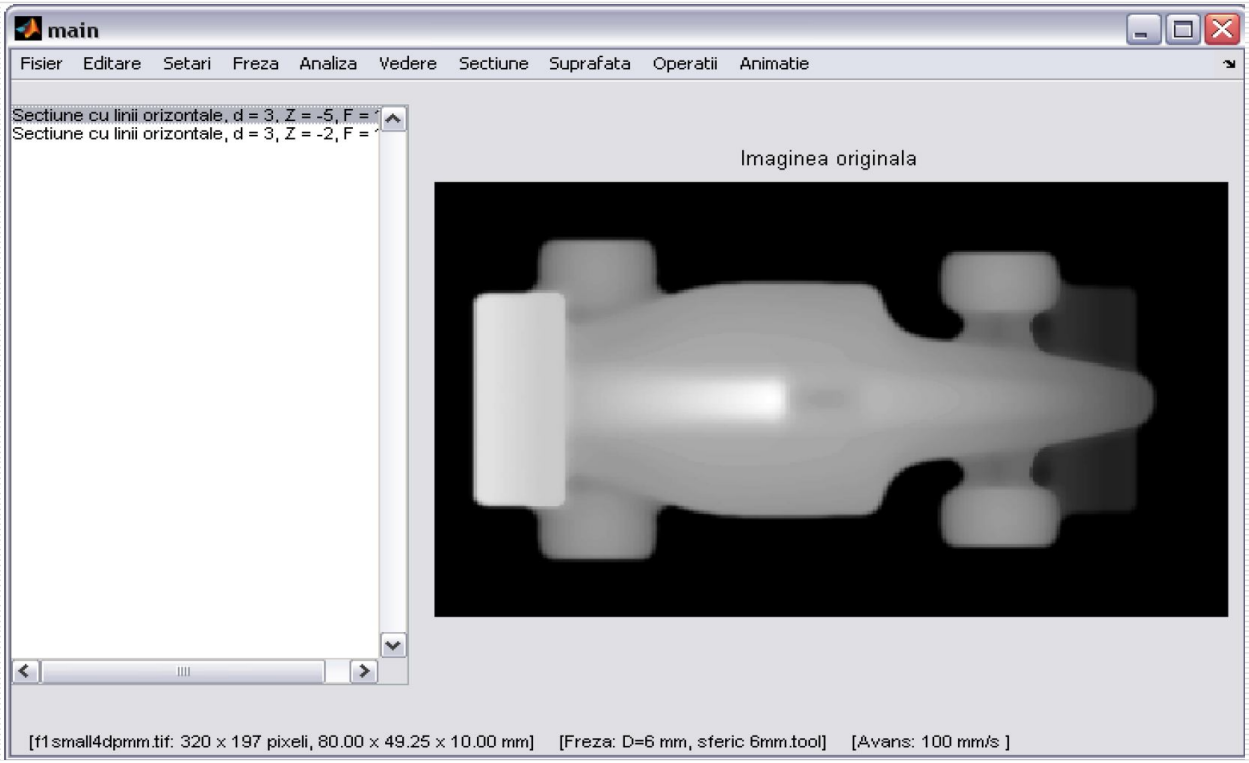


Increasing Speed

- It is not always necessary to compute the whole surface
- The algorithm can be parallelized

Generating CNC toolpaths from grey level image and pattern robot motion

The Software – GUI Design

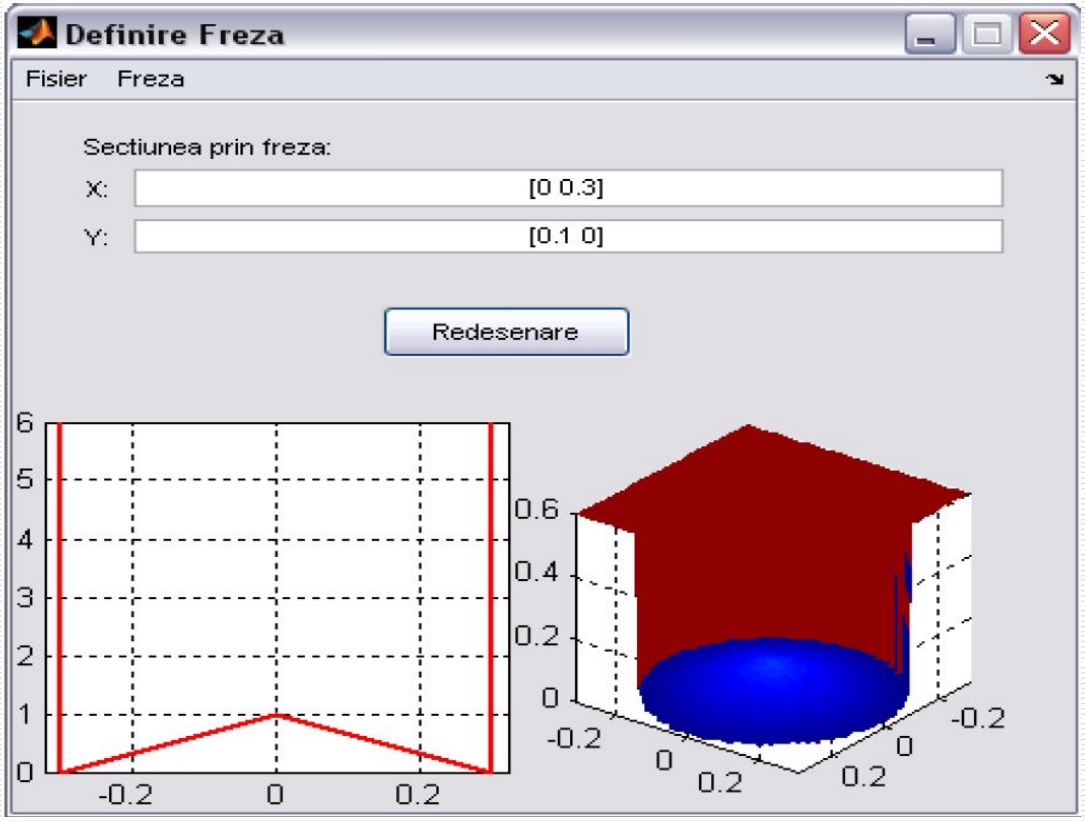


Software Features

- Grayscale model support
- Tool shape editor
- Roughing toolpath generation
- Finishing toolpath generation
- ISO CNC (G-Code) output

Generating CNC toolpaths from grey level image and pattern robot motion

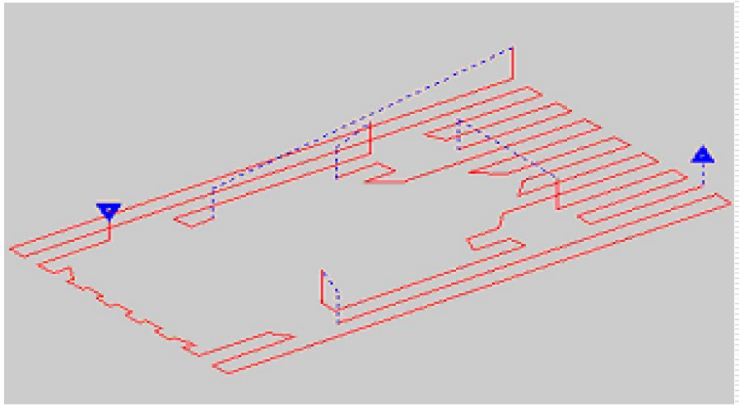
❑ The Software – Tool Shape Editor

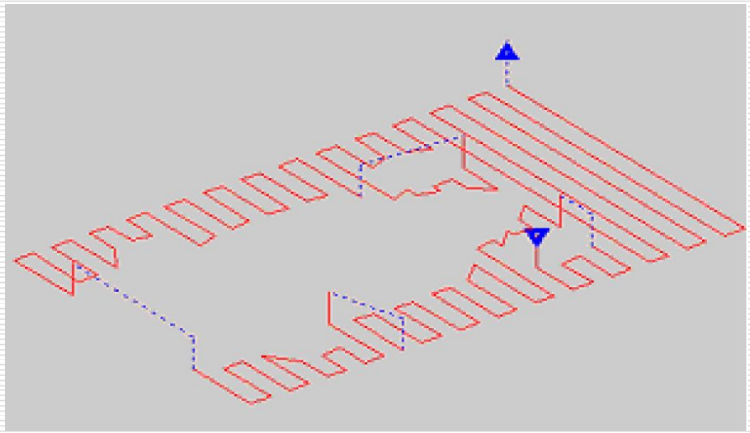


Generating CNC toolpaths from grey level image and pattern robot motion

Generating Roughing Toolpaths

- Each roughing stage is performed *at constant Z level*
- At a given Z level, selecting the region where the cutter should clean up is an *image thresholding* operation
- For flat endmill cutters 2D offset compensation was used

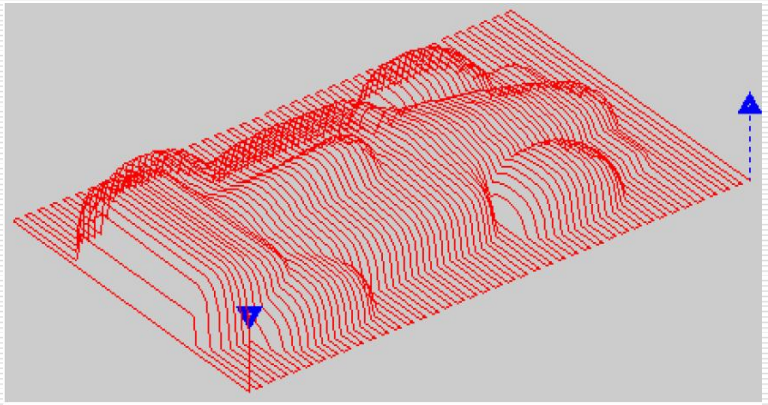


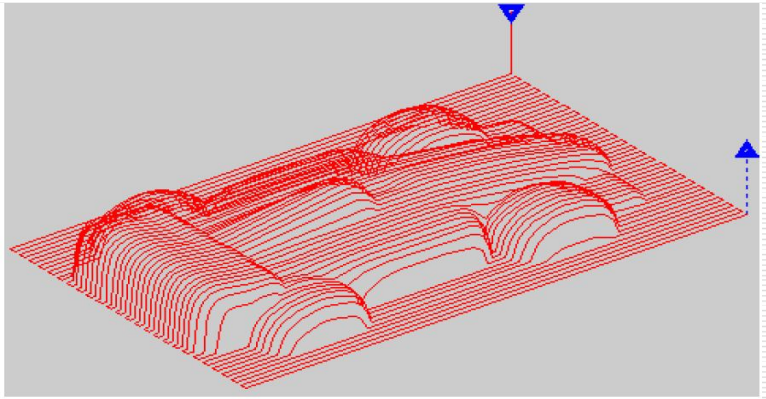


Generating CNC toolpaths from grey level image and pattern robot motion

Generating Finishing Toolpaths – 1st method

- In XY plane, the tool moves parallel with one axis or direction
- The tool moves on the "safe surface"
- There is no need to compute the whole "safe surface"

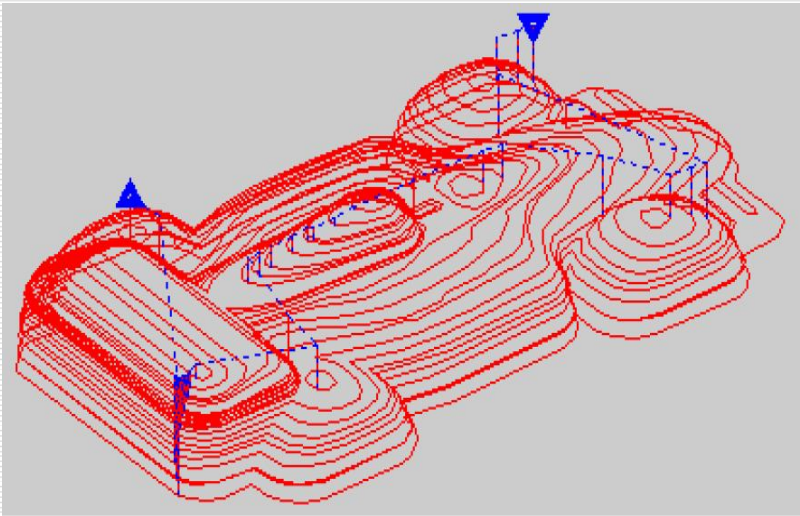




Generating CNC toolpaths from grey level image processing

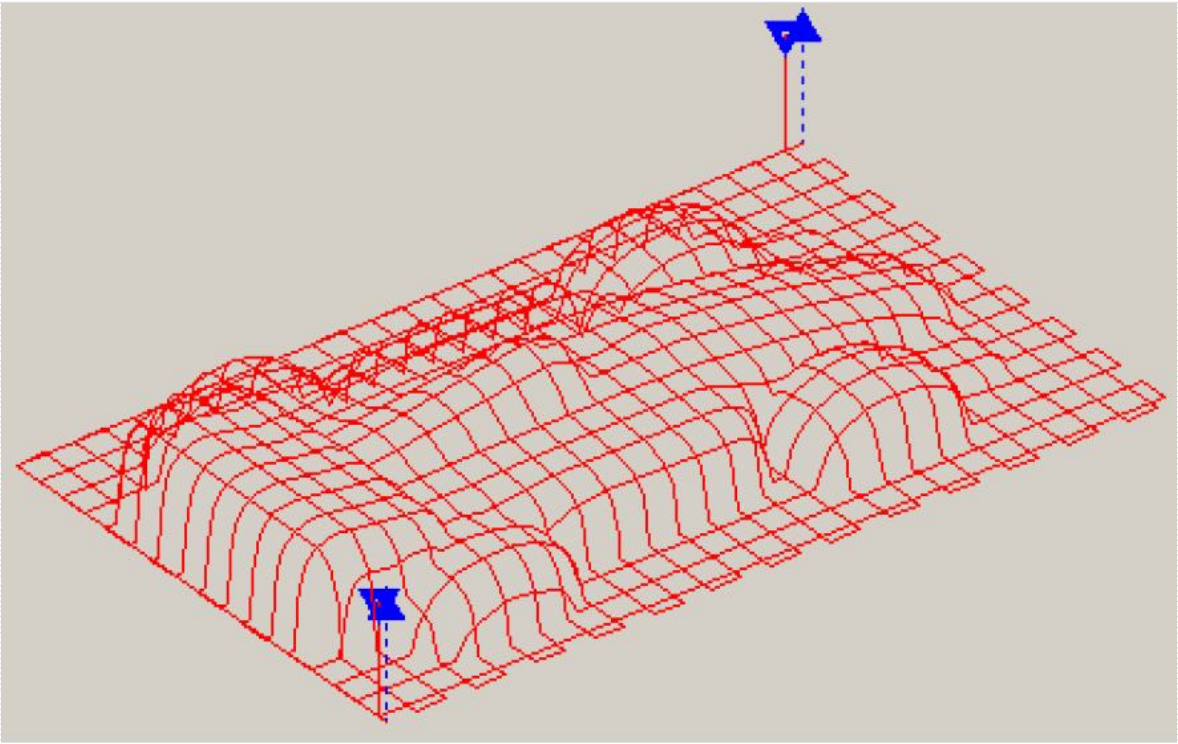
Generating Finishing Toolpaths – 2nd method

- Tool paths are at constant Z levels
- Because of the tool shape, one cannot use 2D compensation any more
- The whole surface needs to be computed!



Generating CNC toolpaths from grey level image processing

- Finishing Toolpaths - Combined



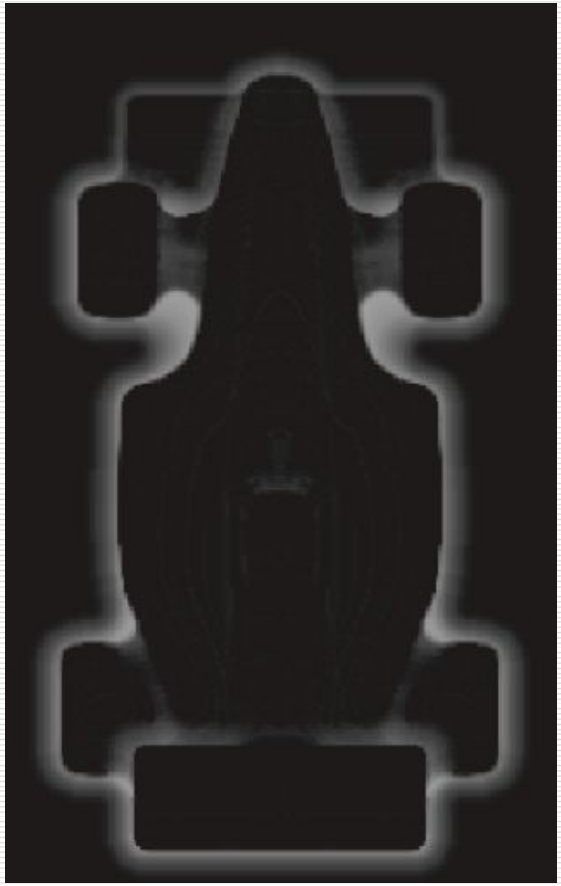
Generating CNC toolpaths from grey level image and pattern robot motion

- **Error Analysis 1**

- A tool can be too big to machine fine details

- At first, one can use a *bigger tool* to machine surfaces without details, and then a *smaller tool* to machine only the small details

- One can simulate one cutting operation and see what could not be machined



Generating CNC toolpaths from grey level image and pattern motion

- **Error Analysis 2**

- Model the toolpath like a greyscale image

- Erode the image using the tool model

- Compare the eroded image with the original model

Generating CNC toolpaths from grey level image and pattern robot motion

ISO CNC Output

- Toolpaths are made of linear segments and circular arcs
- Succesive segments may be approximated with circular arcs
- Toolpath optimization: reduce the time for moving the head without cutting

```
M03
G0 X80 Y7.75
G1 Z-40 F100
G1 X65.5
G0 Z0
G0 X71.75 Y10.75
G1 Z-40 F100
G1 X80
G1 Y13.75
G1 X73
G1 Y16.75
G1 X80
G1 Y19.75
G1 X73.25
G0 Z0
G0 X0 Y0
M05
```

Generating CNC toolpaths from grey level image and pattern robot motion

- **Sample Workpiece**