Journal of Machine Engineering, Vol. 10, No. 4, 2010

trajectory generation,
object detection and identification,
multi-axis machines, collision avoidance

Rafiq AHMAD[1]
Stephane TICHADOU[2]
Jean-Yves HASCOET[3]

INTEGRATION OF VISION BASED IMAGE PROCESSING FOR MULTI-AXIS CNC MACHINE TOOL SAFE AND EFFICIENT TRAJECTORY GENERATION AND COLLISION AVOIDANCE

Demand of high speed production not only increases the complexity of today's CNC process, but also increases the risks and possibility of collision because of the difference between real (machining) and virtual scenes (CAD/CAM process). Idea here is to make this process more intelligent by processing image taken from the real or virtual machine scenes. Identify objects (already known in the CAD database), obtain safe and efficient trajectories that will modify the previous known trajectories from the CAM systems and will be used finally in real machining environments. This work more focuses to improve trajectory generation, collision avoidance and communication in CAD/CAM systems by image processing technique. Safe and Efficient Trajectory (SET) algorithm for point trajectory is discussed along with its extended version for object trajectory known as Rectangular Enveloped object Safe and Efficient Trajectory (RESET) algorithm that will perfectly generate safe un-functional trajectories [3] for multi-axis machine tool envelop. Meanwhile scene objects are detected and identified by image processing tool while trajectory and setup is optimized and improved accordingly in order to avoid collision. This generated trajectory can be used for setup correction before and after production or for "real times/online" production. Finally work has been validated through real and virtual machine scene images.

1. INTRODUCTION

CNC multi-axis and multi-functional machine tool complexity and demands of high production with high accuracy increases the possibility of risk of interference. Generally, machining interferences are classified into two categories as explained by Wang et al. [14], local gouge and collision. According to him, "Local gouge refers to the interference between cutter tip and work piece while collision is the global interference between cutter or cutter holder and machining environment". The main focus of this paper is collision

[1] IRCCyN, Ecole centrale Nantes, Rafiq.Ahmad@irccyn.ec-nantes.fr, Tel.: ++ 33 2 40 37 69 62
[2] Ecole centrale Nantes, Stephane.Tichadou@irccyn.ec-nantes.fr, Tel.: ++ 33 2 40 37 69 06
[3] Ecole centrale Nantes, Jean-Yves.Hascoet@irccyn.ec-nantes.fr, Tel.: ++ 33 2 40 37 69 49

avoidance for getting safe and efficient trajectory by image processing as very little work has been done in multi-axis CNC systems for this purpose.

Current CNC technology lacks to avoid collisions in real times and even it's difficult to communicate the real changes from the real systems to its virtual models for any possible modifications of trajectory. This work will more focus on Mill-Turn multi-axis machine tool safe and efficient trajectory generation for online and offline setup correction because it's the new generation of machine tools with high complexity. The goal is to give bit intelligence to machine in order to avoid online collision for generating safe and efficient trajectory meanwhile minimizing the gap between CAD/CAM and its real machine counterparts. Although this work more focuses on virtual setup correction and optimization for simplicity but results are also applicable in online production. This paper is considering a 3-axis Mill-Turn CNC machine as a case study with more focus on its virtual machine model image. Image generated from virtual scene are easier to process and developed methodology can be extended to the real systems image processing in future.

The context of work presented here is to improve the trajectory generation process by avoiding collision in CAD/CAM system in order to minimize the differences and enhancing the communication between real and virtual machine scene and to perfectly integrate the image processing technique in CNC technology. A machine must be able to take the right decision at right time in "real times" for avoiding collision and generating safe and efficient trajectories or it must have perfect preparation. Machine vision can solve this problem by taking real or virtual image and processing it to generate online corrected trajectories or optimized trajectory for preproduction setup correction. In order to obtain intelligent machining and perfect preparation before production one need to provide a two way communication between virtual model and real machine tool counterparts.

This paper is organized as follows: Section 2 will explain context of the presented work. Section 3 will include some related work. In Section 4, the proposed approach will be discussed. Section 5 will apply the proposed approach and discuss the algorithm developed. Section 6 will show the results for applied criteria. Finally, the conclusion will be given at the end.

2. RELATED WORK

A lot of work has been done for vision system in robotics but minimum efforts has been done till moment for integrating vision based image processing in CNC machines for collision detection and intelligent trajectory. This work is restricted to un-functional trajectories (tool displacement); where as functional trajectories (during machining) are not treated here. In functional trajectories one can avoid collision by changing tools orientation while in un-functional trajectories one needs to avoid tools collision by changing tool path [3]. CNC multi-axis pre-production functional trajectory collision detection strategies for virtual scene are discussed by [14],[7], and [12] for multi-axis machine tools. C-space collision detection was introduced by [10]. Interference analysis between a manually pre-defined tool and an arbitrary work piece has been done by [6]. Tian [13] had developed a 3D vision system for real and virtual image comparison in CNC environment that explains the problematic of differences between reality and virtuality. Ata [1] used a circular obstacle concept for to avoid robotic arm collision that can be used only if obstacles are circular or if one consider spheres around objects in CNC machines.

Remarkable work has been done for object identification and recognition in an image. DeVel et al. [2] used 2D pixel data for objects recognition using random image lines ina scene. Pal et al. [11] proposed a hybrid approach for non-interacting 3D-feature identification from CAD database. A skeleton based graph matching method is proposed for object recognition by He [5]. 3D object recognition is proposed in Lee [9] that can recognize a real object from its virtual model set. Lee [8] used the concept of Familiarity which according to him is a measure of the resemblance of local features extracted from the input image to features of trained object models stored in a database. These references give the motivation for object identification for multi-axis CNC intelligent trajectory process in order to know objects and its mission in the scene. This paper doesn't include details of algorithms used for this purpose because image processing software is used to do this process automatically that could be integrable to CNC. Whereas object detection is a part of proposed algorithm that will be discussed later in the coming sections.

3. APPROACH

The approach discussed here leads to development of an algorithm based on machine vision that can generate safe and efficient trajectory for a CNC multi-axis machine tool by avoiding obstacles and strengthening the communication between CAD/CAM andreal machining. The approach also provides some other links of vision system with CNC

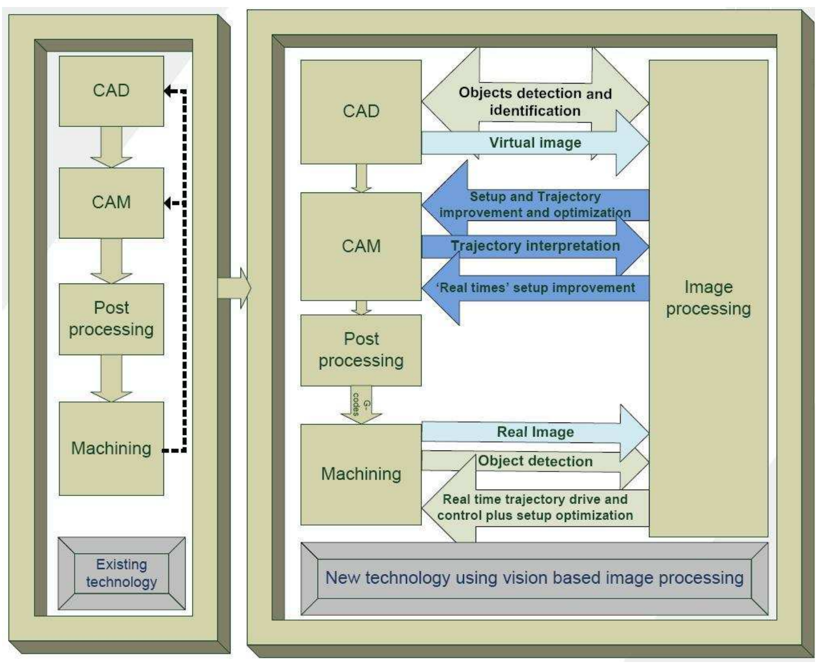

Fig. 1. Technology evolution based on Image processing

processes. In current CNC systems, one can always move forward to the production but it's difficult to move backward and to check for any possible corrections because virtual and real scenes may always have some differences. To minimize the differences and enhance communication between real and virtual machine tools and to integrate the image processing technique in CNC technology proposed work has been divided into many phases of communications as shown in Fig. 1. Objects in real or virtual image are detected and identified in order to avoid collision for generating safe and efficient trajectory.
Fig. 1 shows that existing technology rely on virtual models but if there is slightly change in real machine tool dimension or position the feedback system is unable to communicate it to the virtual model and hence it leads to undefined production risks. Image processing is therefore useful to take a real image from the actual machine scene and identify and localize objects in the scene for getting points of trajectory or avoiding collision for a perfect setup preparation or "real times" correction and optimization. Image taken from real system will identify real objects and its position in the scene and will communicate this data to algorithm proposed for safe trajectory generation.

Some important steps shown in Fig. 1 are discussed below.

3.1. OBJECT DETECTION

Detection and localization of objects in a scene are important in order to know the exact position for avoiding collision. Two types of image objects detection can be useful to discuss here. Detection by Matrox inspector (Software: copyright Matrox Electronic Systems Ltd) and RESET algorithm detection. Objects can be detected and localized in an image using Matrox inspector while RESET algorithm detects tool collision with obstacles at pixel level. This paper more focuses on RESET algorithm collision detection, discussed in coming sections. While if exact localization of objects present in scene are required then Matrox inspector can be used as a tool.

3.2. OBJECT IDENTIFICATION

Objects identification in the virtual or real scene is necessary to know the exact type of object and its mission in the scene. Object identification is mostly done between Matrox inspector and CAD database. Tools and work piece are identified for getting un-functional trajectory initial (tool start point) and final points (production point), while other objects are identified for avoiding collision. Tool model already defined from virtual image is identified in real image for the same scene as shown in Fig. 2.

This process will eliminate the possibility of error in position between real and virtual machine tool. Also this will provide the tool position as input to RESET algorithm for trajectory generation.

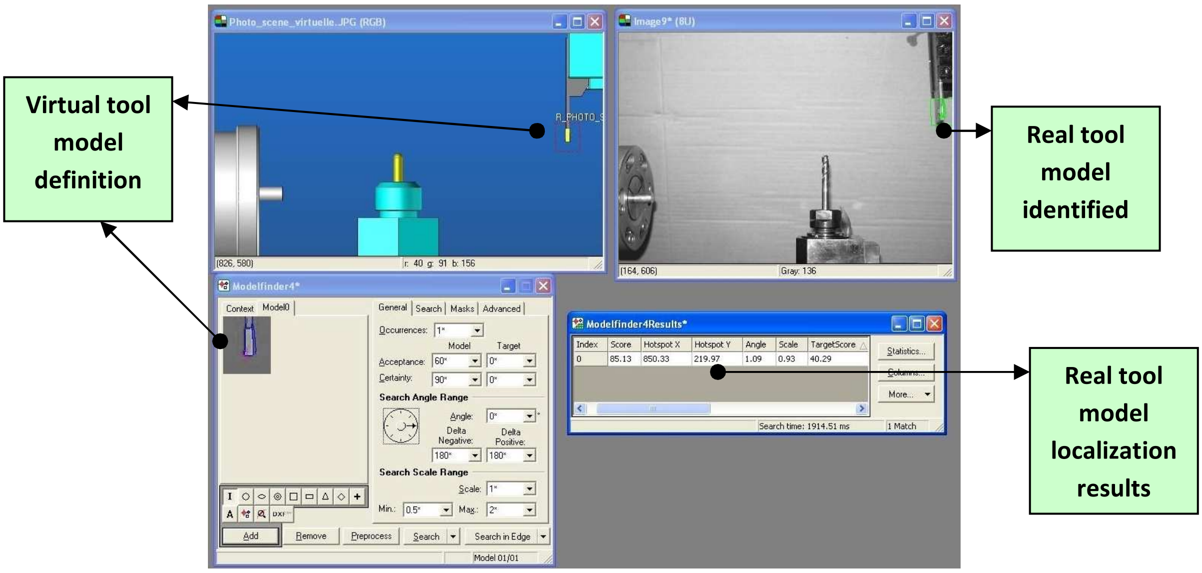

Fig. 2. Model identification in real image

3.3. TRAJECTORY INTERPRETATION

Trajectory can be interpreted from CAM and post-processor (G-code interpretation) directly and can be used as input by image processing unit. For the case presented in this paper, trajectory interpretation is the straight line from tool initial point to the point of production. Generally interpreted trajectory data from CAM and post-processor can be used to generate safe trajectory directly on image. It just needs a proper camera calibration [13] for on machine setup. Interpreted trajectories may consist of straight line or different curves that depend on specific machine type. This process is done between vision system and CAM/Post-processor.

3.4. TRAJECTORY IMPROVEMENT AND OPTIMIZATION

The next step is to improve the trajectory by detecting and avoiding collision and finally optimization. Trajectory improvement and optimization is the current focus of our paper which will be discussed later in proposed algorithms.

3.5. “REAL TIMES” TRAJECTORY CORRECTION

“Real times/online” trajectory correction and improvements can be treated for two cases, “real times” trajectory correction for a real scene and “real times” setup correction. “Real times” trajectory correction for real image will not be treated here as the main focus is on virtual setup because of simplicity. “Real times” safe trajectory is achieved for a virtual machine tool by detecting and avoiding obstacles at pixel level in a virtual image for "real times" setup correction (detailed in next section). Finally this "real times" trajectory is optimized to get trajectory correction for setup preparation.

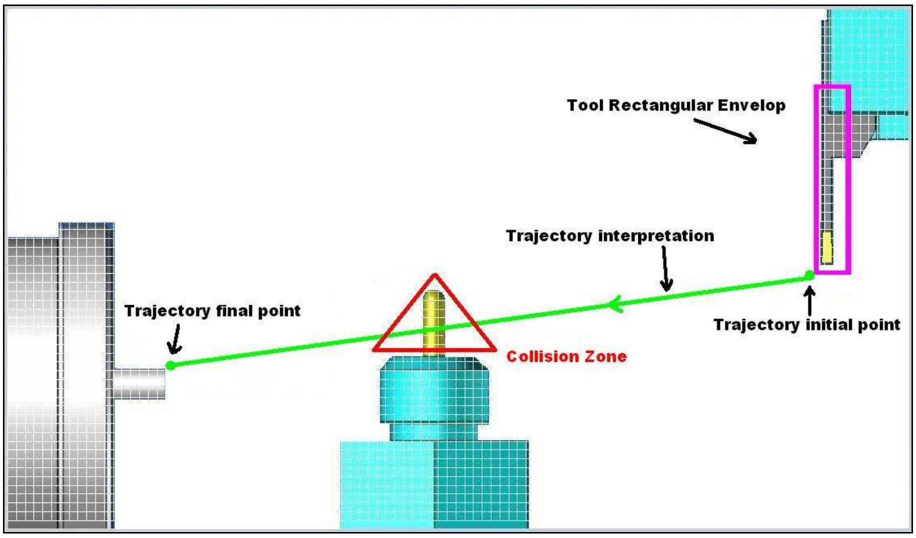

Fig. 3. Straight line trajectory

4. PROPOSED ALGORITHMS

The proposed algorithms can be used by CAM or intelligent machine system for generating collision free pre-process trajectories for real or virtual scene. Two possible solutions arise while working on the problem of collision in CNC systems. Either removes the possibility of collision or make a system that can avoid collision in "real times". Algorithm presented here could be used for both purposes. This algorithm can efficiently remove the possibility of collision by pre-processing real or virtual image to find an optimal and safe trajectory for a machine tool or can be used for online safe trajectory setup correction. It can also be used for online drive and control while work presented here is the initial step to achieve this objective. A simple corner detection algorithm is presented that will be used to provide safe points in space to SET and RESET algorithms.

4.1. CORNER DETECTION ALGORITHM

Corner detection algorithm is used to provide safe point for trajectory near the corner points of obstacles present in the prescribed area. Algorithm usually scan image from left to right and from top to bottom for all pixel values that will be treated for trajectory as per procedure shown in Fig. 4. To avoid unnecessary calculations, algorithm can be applied by defining a region of interest in an image. This algorithm is very fast because it searches each pixel only once.

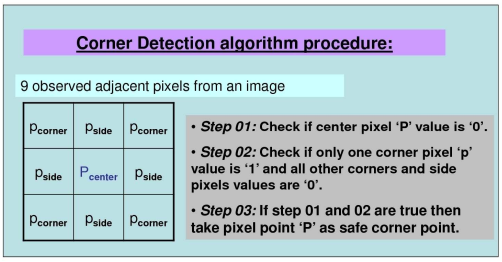

Fig. 4. Corner detection algorithm procedure

4.2. SAFE AND EFFICIENT TRAJECTORY ALGORITHM (SET ALGORITHM)

SET algorithm generates safe and efficient trajectory for a multi-axis machine tool tip point using image processing. Computing software (Matlab) programming is used for the development of this algorithm. SET algorithm takes a real image from the machine environment or a virtual image from CAD model as its input. Process this image for object detection. Generate trajectory using interpreted trajectory and after detecting collision, it takes corrective action accordingly for improving and optimizing trajectory. Image pre-processing is required for foreground and background clarity before applying SET algorithm. Algorithm first convert input image to binary by providing certain threshold value, depending on the clarity of objects in the image. This binary image consists of two colours, white and black set of pixels, with values “1” and “0” respectively. White colour represents objects in the scene while black colour is assigned to the background. Next step is to extract pixel values matrix from the 2D image scene that will represent “1” for pixel containing object and “0” for free area calling it matrix “A”. After getting binary matrix from the original image the algorithm tries to find shortest straight line trajectory points in matrix from initial position to production point. Algorithm verifies each point of the straight line trajectory and if it founds a point inside obstacle, it changes that point to nearest safe corner detected by Corner detection algorithm. Algorithm then keep the last points already known for the safe trajectory and make the new corner point as initial point. Algorithm repeats it’s self, finding all safe points and at the end save all safe trajectory points known as “real times” safe trajectory. A simple program script for algorithm is given below while the detailed process will be discussed in next section as RESET algorithm uses the same principle. Note that “x” and “y” represents the relative pixel coordinates in 2D image which are different from machine axis.

```
for     Checking all trajectory points
        Matrix A [% Image pixel values]
        Matrix B [% Same size as A, just used for computing]
        M=A+B  [% Assign value of A+B to Matrix M]
        If  M(x,y) = = 2 [% (x,y) is the current tool trajectory point for observation]
                [% Collision point detected, therefore take a new safe trajectory point]
```

(x,y) = = safe corner point [% Assign a safe corner to the current observed point from Corner detection algorithm]

else Mark (x,y) as safe trajectory point

end

end

While doing pixel value analysis through matrices some definitions for collision and non collision points are as under:

• *Confirmed collision:* A collision may be detected if a pixel point gives a value “2” in final matrix “M” (Fig. 05). This collision point is later discarded and another corner point is assigned to the trajectory that can make the trajectory more safe and efficient.

• *Collision free:* A pixel that gives an odd value in final resulted matrix “M” is always a collision free point that can be used in trajectory if needed.

• *Possible collision:* Possibility of collision is there if a pixel point gives a value “4” in final matrix “M”.

4.3. RECTANGULAR ENVELOPED OBJECT SAFE AND EFFICIENT TRAJECTORY ALGORITHM (RESET ALGORITHM)

RESET algorithm is the extension of SET algorithm for a rectangular enveloped tool. As SET algorithm is used only for the tool tip point trajectory, this extension will instead provide safe and efficient trajectory for a rectangular tool. Trajectory and setup is improved and optimized according to the detected objects in order to avoid collision using RESET algorithm. RESET algorithm usually focuses on trajectory of four corner points of the rectangular tool and at every pixel point it checks for collision. Initially before applying RESET algorithm, one need to identify the tool as shown in Section 04 (Object identification) in order to get initial point for trajectory. Pixel values are extracted from image same as SET algorithm and object collisions are detected after checking the derived matrices. Detected collision points are modified accordingly using corner detection algorithm and safe points in space generated by RESET algorithm.

Same definitions discussed for SET algorithm also applies to RESET algorithm with a bit modified confirmed collision definition as below:

• *Confirmed collision:* A collision may be detected if a rectangular tool tip pixel point gives a value “2” OR any of other tool three corners gives a value “7” in final matrix “M” (Fig. 05).

Algorithm flow chart is shown in Fig. 5:

Important steps of algorithm:

- Step 01: Image pre-treatment.
- Step 02: Detection of rectangular enveloped tool corner points.
- Step 03: Removal of unnecessary points obtained in Step 02.

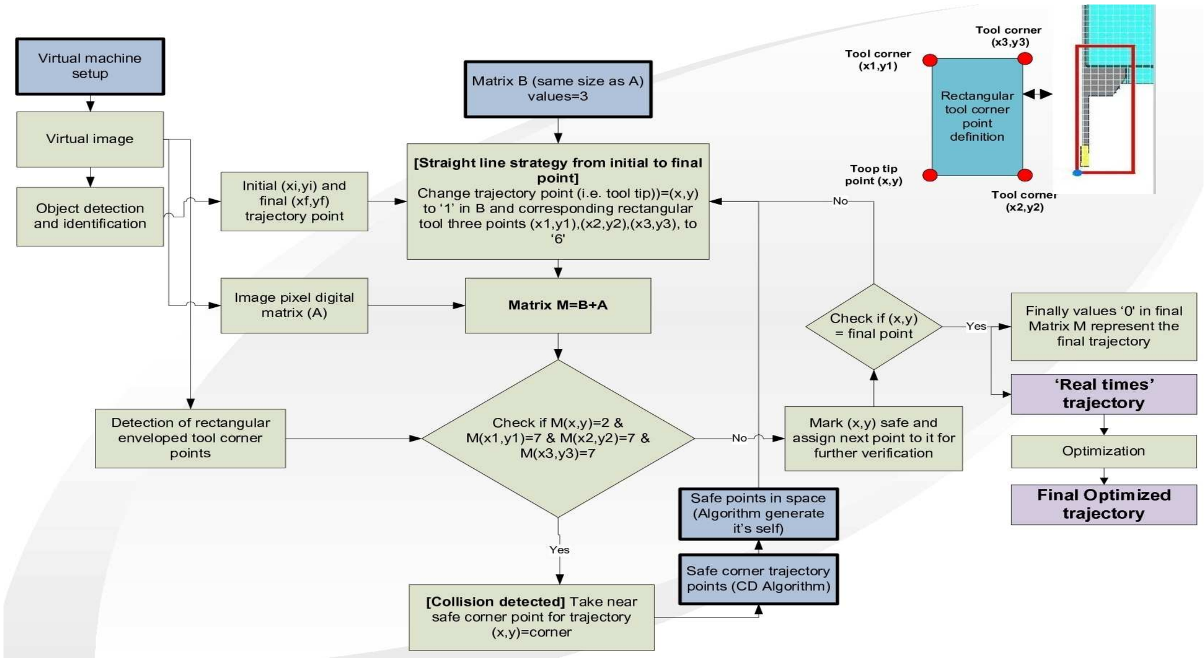

Fig. 5. RESET algorithm flow chart

- Step 04: Replace the tool body pixels value “1” with background value “0” so that self collision detection points must be avoided because it’s impossible in case of hard objects.
- Step 05: Definition of area of interest for corner detection and detection of safe corner points for objects present in that area.
- Step 06: Define some extra safe points in the space because the detected safe corner’s points are not enough to provide a safe trajectory for rectangular tool. Unlike SET algorithm that works only with safe corners for point trajectory.
- Step 07: Now go straight from initial to final point for tool tip and verify all new positions of the rectangular tool corner points for collision. If collision is detected for any of the tool corner points then replace the tool tip point with already detected closer safe corner or space point.
- Step 08: Repeat step 07 till reaching the final point. If current goal is “real times” safe trajectory then stop at step 08 that gives the safe trajectory otherwise moves to next step.
- Step 09: For setup preparation and efficient trajectory, optimize the trajectory obtained in step 08, using a number of defined criteria.

Results of RESET algorithm are shown in the next section.

5. RESULTS

RESET algorithm is applied on virtual image taken from a virtual setup as shown in Figures below. Algorithm gives four different trajectories depending on tool envelop and optimization. Two types of enveloped tool results are shown here, big and small. For each envelop two types of trajectories are obtained, one for “real times/online” setup correction and another optimized trajectory that can be used for setup correction and improvement. “Real times” trajectories for setup correction as shown in Fig. 6 and Fig. 8 are not always efficient in terms of certain criteria like time, trajectory length, straight line trajectory changing points and speed. Trajectories shown in Fig. 06 and Fig. 8 are optimized while minimizing the criteria; resulted trajectories are shown in Fig. 7 and Fig. 9.

Comparison of all the trajectories obtained for tool envelops are shown in table below for an ideal straight line trajectory length of 330mm. Straight line trajectory is assumed ideal just for comparison purpose by supposing that we are not considering collision so tool will go straight from initial point to the point of production directly with straight line as shown in Fig. 3.

Results are plotted to give more expression as shown in Fig. 10. Smilies are shown for respective trajectory path representing satisfaction to criteria like minimum trajectory turning points and minimum trajectory length with no collision. From graph it’s clear that straight line is a better option for criteria but it causes collision while all other trajectories are safe. Fig. 7 and Fig. 9 are better satisfying the criteria therefore they are safe and efficient while Fig. 6 and Fig. 8 are not that much efficient for defined criteria but are safe trajectories and better for “real times” correction.

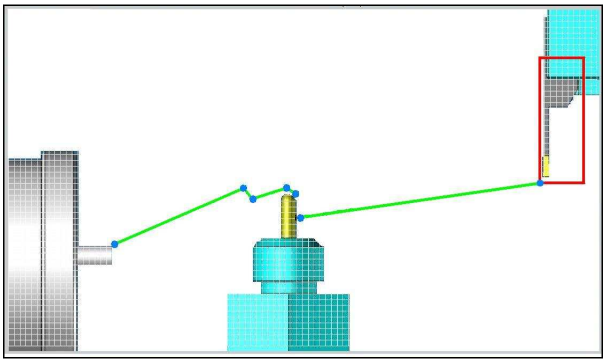

Fig. 6. “Real times” setup correction relative big tool envelop

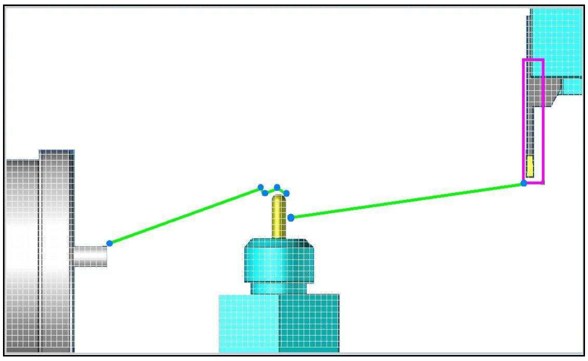

Fig. 8. “Real times” setup correction for relative smaller tool envelop

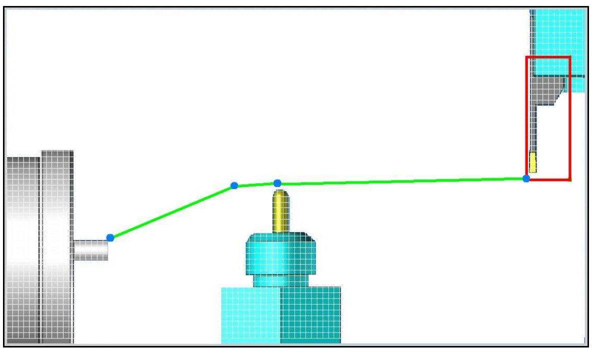

Fig. 7. Setup correction after optimization for relative big tool envelop

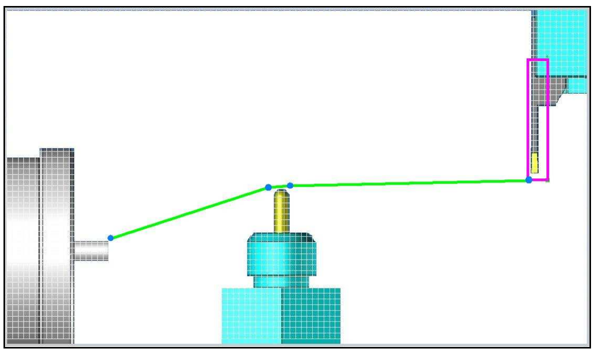

Fig. 9. Setup correction after optimization optimization for relative smaller tool envelop

Table. 1. Trajectories comparison

S.no	Trajectory names	Tool trajectory turning points (pts)	Trajectory length (mm)
1	Straight line trajectory (Fig. 3)	0 pts	330 mm
2	"Real times" setup correction for big envelop (Fig. 6)	5 pts	360.43 mm
3	"Real times" setup correction for small envelop (Fig. 8)	5 pts	359.17 mm
4	Setup correction after optimization for big envelop (Fig. 7)	2 pts	335.11 mm
5	Setup correction after optimization for small envelop (Fig. 9)	2 pts	333.17 mm

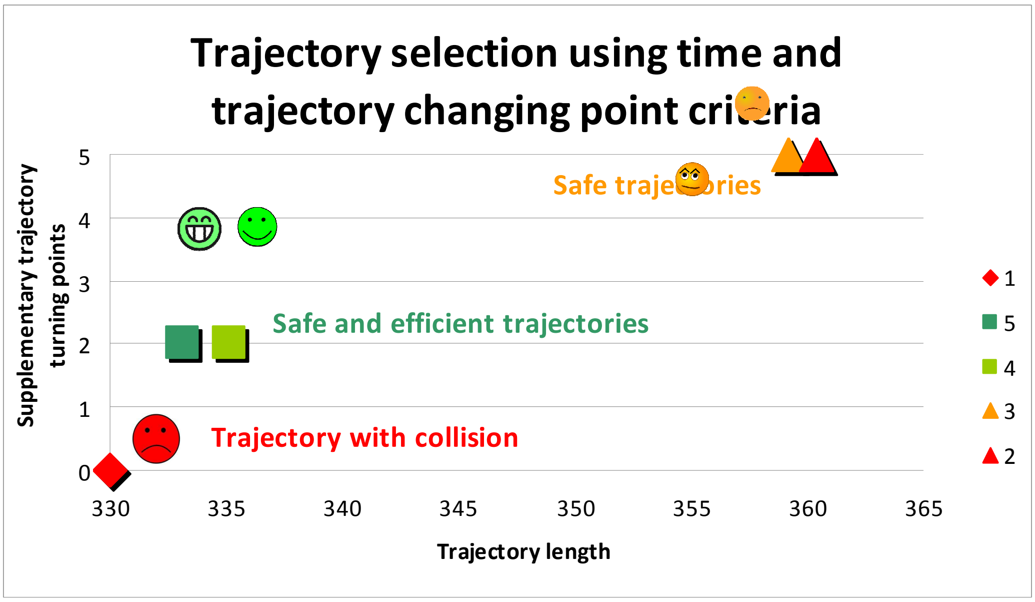

Fig. 10. Trajectories comparison

Hence the best trajectory is the one in Fig. 8 (Number 5 in Fig. 10) which gives an efficient trajectory for satisfying all criteria. This paper doesn't consider criteria like speed and acceleration but speed is more inversely proportional to the trajectory turning points. Increasing trajectory turning points will decrease the speed of machine tool. Also if there are more trajectory turning points then there will be more acceleration and hence trajectory will not be smooth that is not good for machine tool. Hence trajectory obtained in Fig. 7 and 09 are good for defined criteria but if real times trajectories are required then off course it's not possible to do the optimization in "real times" therefore the trajectories obtained in Fig. 6 and Fig. 8 will be the best options especially the one in Fig. 8.

6. CONCLUSIONS

This paper gives three ways of improvement in multi-axis production process. Approach presented here enhances the communication gap between virtual and real scenes for identifying and localizing tool and obstacles in the scene, by image processing taken from real machining or virtual CAD/Cam model. This approach also provides intelligent trajectory generation process in order to improve and optimize the multi-axis machine tool trajectory for "real times" setup correction. Corrected trajectory is optimized for certain criteria like time, length and trajectory changing points for getting preprocess setup correction. Results are plotted and tabulated for virtual machines images but it can be extendible to real images, note that this paper used virtual images for the sack of simplicity and providing links with CAD and image processing unit. These results are based on virtual images because they are relatively clean for processing. This process can lead to work on real environment while staying in the context of virtual for increasing and facilitating communication between reality and virtuality. This work can be integrated in STEP-NC technology [4] in future which is the broad context of presented work. Other consideration like mobile and immobile, known and unknown plus controllable and non controllable obstacles and machine tools can be considered in future.

REFRENCES

[1] AA ATA and TR MYO., 2006, *Collision-free trajectory planning for manipulators using generalized pattern search*, International Journal of Simulation and Modelling, 5/4, 145–154.

[2] O. DE VEL and S. AEBERHARD., 2000, *Object recognition using random image-lines. Image and Vision Computing*, 18/3, 193–198.

[3] J-Y. HASCOET, F. BENNIS and F. GUERIN., 1993, *Optimization of tool trajectory for five axis control machining*, Int. Conf. Iasted Robotics and Manufacturing, 120–122.

[4] J.Y. HASCOET, M. RAUCH and S.H. SUH., 2007, *Relevance of step-nc standard for high level toolpaths generation.* Int. Conference on High Speed Milling.

[5] LEI HE, CHIA Y. HAN, BRYAN EVERDING and WILLIAM G., 2004, Wee. *Graph matching for object recognition and recovery,* Pattern Recognition, 37/7, 1557–1560.

[6] S. HO, S. SARMA and Y. ADACHI., 2001, *Real-time interference analysis between a tool and an environment.* Computer-Aided Design, 33/13, 935–947.

[7] OLEG ILUSHIN, GERSHON ELBER, DAN HALPERIN, RON WEIN and MYUNG-SOO KIM., 2005, *Precise global collision detection in multi-axis nc-machining.* Computer-Aided Design, Special Issue: Product Design, Integration and Manufacturing, 37/9, 909–920.

[8] SEUNGJIN LEE, KWANHO KIM, JOO-YOUNG KIM, MINSU KIM and HOI-JUN YOO., 2010, *Familiarity based unified visual attention model for fast and robust object recognition.* Pattern Recognition, 43/3, 1116–1128.

[9] YANG-LYUL LEE and RAE-HONG PARK., 2002, *A surface-based approach to 3-d object recognition using a mean field annealing neural network.* Pattern Recognition, 35/2, 299–316.

[10] KOICHI MORISHIGE, KIWAMU KASE and YOSHIMI TAKEUCHI., 1997, *Collision-free tool path generation using 2-dimensional c-space for 5-axis control machining.* The International Journal of Advanced Manufacturing Technology, 13, 393–400.

[11] PRALAY PAL and ANJANI KUMAR., 2002, *A hybrid approach for identification of 3d features from cad database for manufacturing support.* International Journal of Machine Tools and Manufacture, 42/2, 221–228.

[12] T.D. TANG, ERIK L.J. BOHEZ and PISUT KOOMSAP., 2007, *The sweep plane algorithm for global collision detection with workpiece geometry update for five-axis nc machining.* Computer-Aided Design, 39/11, 1012–1024.

[13] XIAODONG TIAN, XI ZHANG, KAZUO YAMAZAKI and ADAM HANSEL., 2010, *A study on three dimensional vision system for machining setup verification. Robotics and Computer- Integrated Manufacturing,* 26/1, 46–55.

[14] QING-HUI WANG, JING-RONG LI and RU-RONG ZHOU., 2006, *Graphics-assisted approach to rapid collision detection for multi-axis machining,* The International Journal of Advanced Manufacturing Technology, 30, 853–863.