Overview of Assembly Modeling, Planning, and Instruction Generation Research at the Advanced Manufacturing Lab

Satyandra K. Gupta
Director, Advanced Manufacturing Lab
University of Maryland, College Park, MD 20742

Technical Report, Advanced Manufacturing Lab, University of Maryland, College Park, MD 20742

December 2012

1. Introduction

Most products require some form of assembly. Hence assembly operations are a very important part of the manufacturing process. Performing assembly operations requires time and effort and hence it contributes to manufacturing cost. In many applications, it is desirable to reduce the number of assembly operations by consolidating multiple parts into a single part. Processes such as multi-material 3D printing and in-mold assembly can be used to reduce the assembly operations.

In many applications assembly operations cannot be eliminated. Designers and process engineers need software tools to manage assembly deign and generate instructions to perform assembly operations. Developing these tools requires developing new geometric reasoning algorithms. Our research is developing algorithmic foundations for realizing assembly design and planning tools. Many assembly operations are performed by humans. Humans require training to perform these assembly operations. We have developed a virtual environment based training tool that enables humans to learn complex assembly operations in the virtual environment. This report presents a summary of work in the assembly area in the Advanced Manufacturing Lab.

2. Assembly Modeling and Search

Computer-Aided Design (CAD) systems have become very popular in the industry. These CAD systems are being used to generate models of parts and assemblies. These models are used as a basis for engineering analysis and generate manufacturing plans. CAD models also allow virtual prototyping. This reduces the need for physical prototyping. Nowadays, organizations routinely set up databases of CAD models to enable all participants in the product development process to have access to 3D data to support their functions. Specifically, manufacturing and service engineers are expected to greatly benefit from these databases. These databases are updated with the latest versions of parts and assemblies. This significantly improves information dissemination. CAD databases for even moderate size companies are expected to be large in size. A product assembly can contain many subassemblies and each subassembly can contain many parts. Therefore, even a small organization that has multiple product lines may add hundreds of assemblies to their database every year.

In addition to supporting downstream manufacturing and service operations, assembly databases can be very useful during the design phase as well. There are two main uses of an assembly database during the design stage. The first possible usage is to locate existing assemblies that can be reused in a new product. Such reuse reduces product development time by eliminating the need for modeling, analysis, and process planning for the assembly being reused. The second possible usage is to provide access to existing design knowledge. Once designers manage to find an assembly with the desired characteristics, they can also access associated data such as cost, reliability, and failure reports.

Currently, designers have access to several types of search tools. If assemblies are stored in hard drives, they can use file name based search tools. This strategy only works if a meaningful file naming convention based on assembly content is adopted. Unfortunately, many organizations do not name assembles in a meaningful way. For example, a research lab that we visited names assemblies by the product numbers. Any designer who has not worked on a particular product will not be aware of the existence of that product number. A molding shop that we visited names mold designs by dates and customer names. Moreover, when a company acquires another company, often the file naming convention used by two companies are not consistent. Such inconsistencies may require significant amount of manual labor to rename files. These problems suggest need for content-based search tools.

Another way to index assemblies is to attach text notations to assemblies and store them in a Product Data Management (PDM) database. In this indexing scheme, assemblies need to be manually annotated or text labels need to be extracted from CAD models. This scheme provides search capabilities based on text. However, text based search is not always the best way to search for assemblies.

Increasingly, limitations of text-based search are being recognized and methods are being developed to perform search based on the content itself. Such methods are able to exploit rich domain dependent information present in the data. For example, in the recent past, several part geometry-based search tools have emerged. However, these tools, although useful for part searches, are not very effective for assemblies. They can only account for the overall shape of assemblies and do not take into consideration the relationships and structures in assemblies. Currently, content-based search tools do not exist for searching assemblies based on these aspects. Therefore, designers locate assemblies by manually opening various files and browsing through them using a CAD system. This is a highly inefficient use of designer’s time. This becomes a serious problem as the numbers of assemblies in the database grow.

Assemblies contain a very large amount of information and complex relationships. Performing computationally efficient and easy-to-specify search on assemblies will require careful selection of the search criteria and associated definitions. Our central premise is that users will utilize distinguishing characteristics of the assemblies to initiate search to locate the desired assemblies from the database. In order to perform efficient searches, these distinguishing characteristics will need to include information other than text labels associated with the assemblies. Moreover, these distinguishing characteristics will be different in different domains. A large number of assemblies might be present in the database. But often certain distinguishing characteristics can be recalled by the designers to get a small number of desired results.

We have developed a system for performing content-based searches on an assembly database. We discuss the main requirements for the content based assembly search system. The high-level concept behind the content-based assembly search is shown in Figure 1. We present templates for defining a wide variety of assembly search queries that we have identified after studying user needs. We also describe in detail new algorithms for performing mating relationship based search and characterize their performance. Finally, we describe our system implementation and show several examples to illustrate possible usage of content-based assembly search system.

We conducted a survey involving total of seven users to help us identify assembly search system requirements. These included engineers at a government laboratory, mold designers at a molding shop, engineers working at a small company building custom electronics hardware, and students designing robots. All of these users use assembly design modules. Based on the user responses, we derived the system requirements listed below.

- *Comprehensive Search Criteria*: Different users are likely to remember different characteristics based on the context and their prior familiarity with the assembly that they are looking for. For example, a person who has not seen the assembly but only heard about it may recall different things about the assembly than the person who has actually seen the physical assembly. In order to ensure that the system caters to the needs of a wide range of users and different search contexts, it must support comprehensive search criteria for defining queries. Possible search criteria include information used to define assembly using a CAD system. In addition, any information or characteristic that can be extracted from the information present in assembly models and is likely to be utilized by users during the initiation of the assembly search must also be supported.

- *Ease of Use*: A typical assembly modeling module in a commercial CAD system records a lot of information about the assembly. Hence, comprehensive search criteria requirement is likely to result in hundreds of different criteria. In presence of such a large number of criteria, it might be difficult for users to locate the criteria they are looking for. Hence, they must be organized in a navigation taxonomy that groups similar search criteria together. Navigation taxonomies usually form a tree. We are interested in grouping search criteria such that the navigation tree is not very deep and yet at any level there are not too many branches. Furthermore, such grouping should be intuitive to navigate so that a new user familiar with assembly modeling concepts can locate the desired search criterion with four or fewer mouse clicks.
- *Search Flexibility*: Often users like to initiate search with bare minimum pieces of information and adjust the search criteria based on the search results. Hence, the system should support an iterative search refinement process. In other words, if the search results are too few then the user should be able to reduce the strictness of the search criteria by increasing the cut-off values. Also, if the search results are too many then the user should be able to introduce additional criteria in the form of more assembly characteristics in subsequent searches. Each consecutive search with additional constraints should be performed on only the set of assemblies identified during a previous search.
- *Search Efficiency*: Ultimately for the system to be useful, it should produce search results quickly. Hence, it should employ efficient search algorithms. Our goal is to be able to return search results within one minute for a 10,000 assembly database. The rationale for this target was the following. Many small and medium sized organizations have assembly databases smaller than 10,000 in size. In order to use assembly search in iterative refinement mode, the search time has to be smaller than one minute.

Our goal was to gather main system requirements to develop a research prototype. Specific contributions of our work are described below:

- We have identified a comprehensive set of search criteria to define assembly search queries. As illustrated by five different example scenarios, the identified search criteria allow the users to define a wide variety of search queries. These criteria were selected after a careful consideration of the search algorithm and ease of use. These criteria expand the search options available to users in a typical PDM system. Our proposed scheme also preserves the search options available to users in current PDM systems.
- We have categorized identified search criteria into logical groupings to make it convenient for users to locate the desired criteria.
- We have developed a new algorithm for performing mating graph-based searches. This new algorithm allows checking of compatibility of a query graph containing wild card entries with a fully specified mating graph of an existing assembly. Our algorithm will work well

when the number of node types is larger than (1) the number of nodes in the database mating graph, and (2) the number of nodes in the query mating graph. Both of these conditions are easily met in our application. We have characterized the computational performance of this algorithm and shown that this algorithm works well for typical small query graphs encountered in the assembly search.

- We have successfully developed a proof of the concept system to demonstrate the feasibility of our ideas and algorithms. We have shown using five real-life inspired examples that the system is capable of meeting all the system requirements. Most of the search criteria have been put to use in one or more of the examples.

We expect that this system will serve two purposes. First, it will allow designers to reuse existing assemblies by giving them a tool to identify assemblies with the desired characteristics. Second, it will provide designers access to the design for manufacturing and assembly knowledge contained in the assembly database, and hence transfer best practices to new designs.

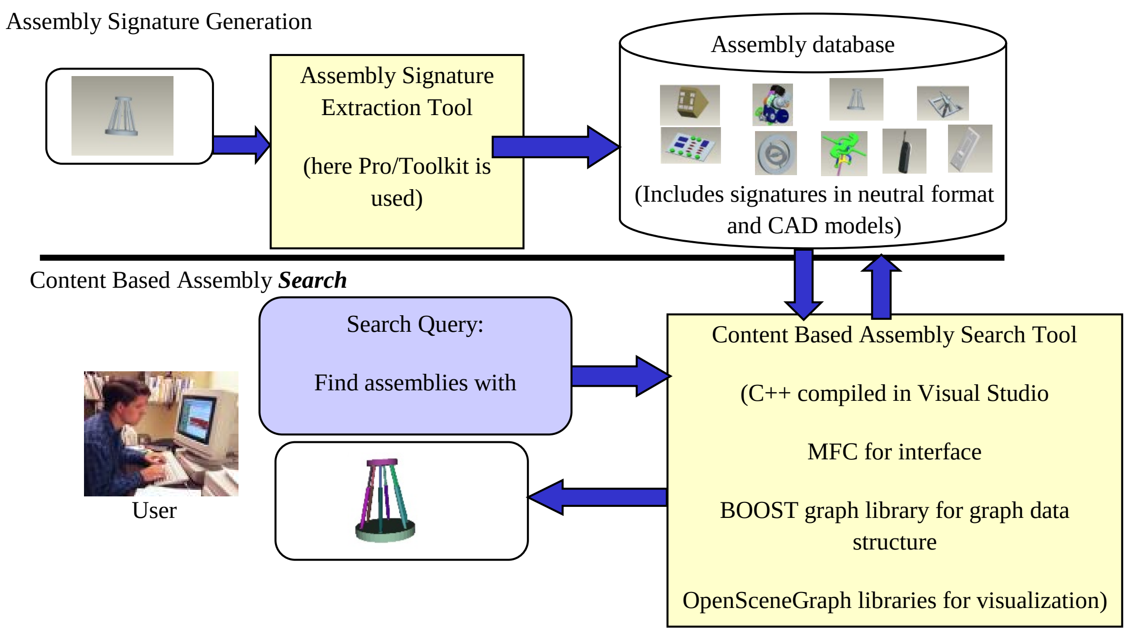

Figure 1: Overview of assembly search system

3. Virtual Environment-Based Training System for Mechanical Assembly Operations

Assembly and disassembly operations are a crucial part of service, maintenance, and manufacturing operations. A large number of such operations are performed by humans. Training a human operator to perform these operations often requires generating training instructions. The following three are representative applications where detailed high-quality assembly and disassembly instructions are highly desirable:

- Complex products (e.g., furniture, grills) that need to be assembled at home by inexperienced end-customers.

- On-site emergency repair and service of complex systems (e.g., aircrafts, ships, tanks) by mechanics with limited experience.
- Assembly operations involving small batch productions of complex artifacts (CAD/PAD, medical diagnostic equipment).

Currently, generating high-quality training instructions is a highly time consuming task. The time is incurred not only in generating the instruction but also in validating them and making changes to them in response to design changes. Moreover, generating multi-media instructions (e.g., 3D animations) is even more expensive than the traditional text-based instructions.

Currently, many organizations are not able to spend the time and money to generate adequate instructions. So often the person performing the assembly and disassembly operations takes a long time to complete the task and/or makes crucial mistakes in performing the tasks.

Our goal is to develop a highly automated authoring capability using which high quality instructions can be generated through a virtual demonstration of the task, without any programming effort by the author (see Figure 2). We plan to utilize the virtual environments as the user interface to assist in the creation of the training instructions. In our envisioned framework, the person authoring training instructions will not be required to perform any programming or be familiar with geometric transformation operations. This work will be automatically performed behind the scenes to speed up the creation time of training instructions and allow anyone familiar with the task to easily author instructions.

Our ideas for generating training instructions by exploiting virtual environment are as follows. First, the training instructor will need to define a work environment. This step will be performed by selecting the necessary artifacts from the library of artifacts. These artifacts will be placed in the scene by simply moving them to appropriate locations. The next step will be to import three dimensional CAD models of the device to be assembled/serviced into the scene and label them. These models will be placed at initial locations in the scene. After this step, the instructor will demonstrate various tasks by selecting the parts and performing the necessary motion steps to execute the desired task. The gross motion will be concerned with correctly orienting parts. The fine motion will be concerned with performing assembly or disassembly operations. The motions carried out by the instructor will be recorded and converted into motion primitives to be used in the training instructions. As illustrated in this scenario, the training instructions will be built by simply enacting the spatial manipulating tasks and no programming will be necessary.

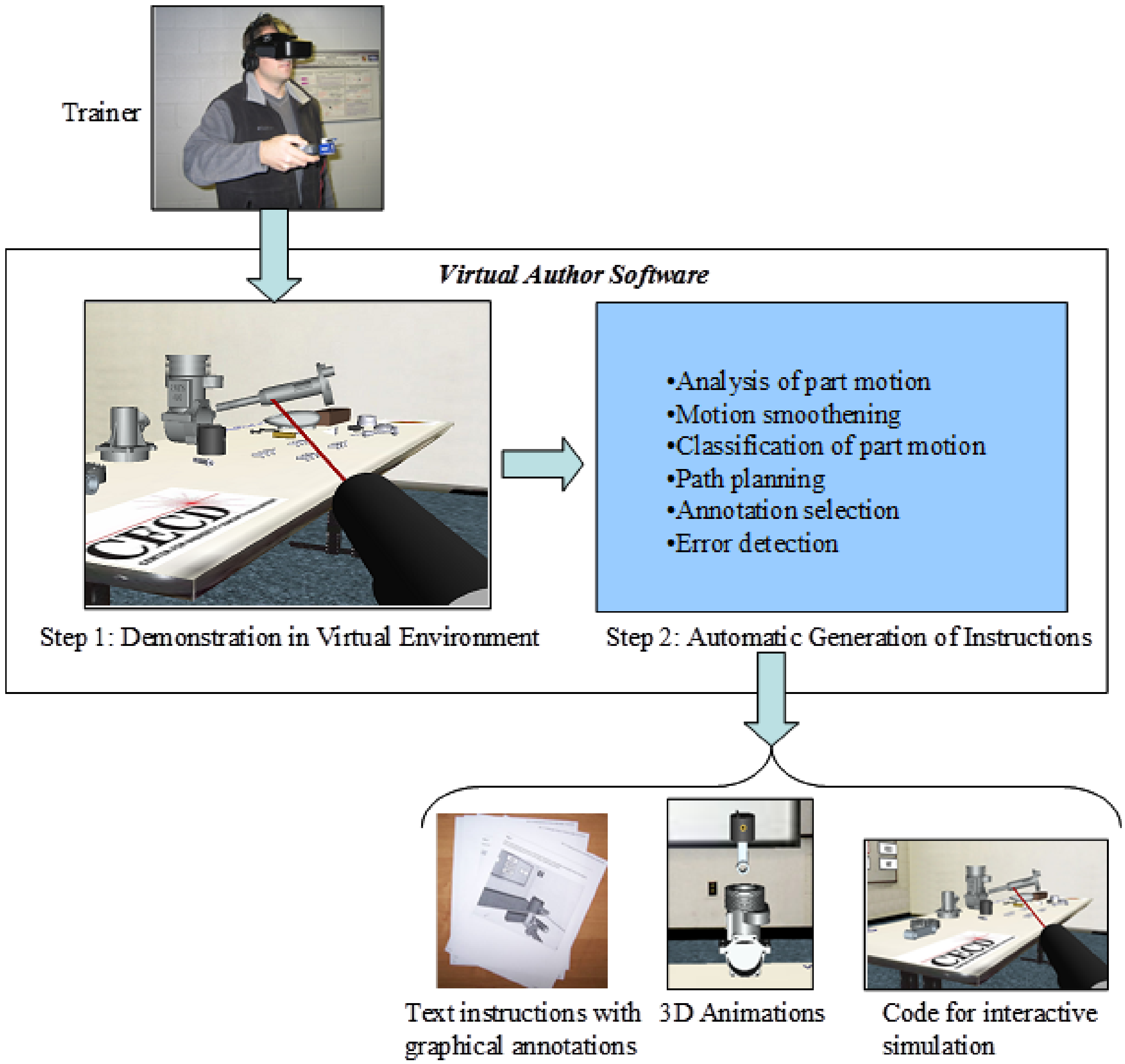

Figure 2: VTS System

Due to the rapid influx of new and changing technologies and their associated complexities, accelerated training is a necessity in order to maintain an advanced and educated manufacturing workforce. We believe that existing training methods can be significantly improved in terms of cost, effectiveness, time expenditure and quality through the use of digital technologies such as virtual environments. The advent of personal virtual environments (PVEs) offers many new possibilities for creating accelerated training technologies.

We have developed a virtual environment based training system called Virtual Training Studio (VTS). The VTS aims to improve existing training methods through the use of a Virtual Environment based multi-media training infrastructure that allows users to learn using different modes of instruction presentation while focusing mainly on cognitive aspects of training as opposed to highly realistic physics based simulations.

The VTS system has two main goals. The first goal is to ensure that virtual environment based instructions for training personnel in the manufacturing industry can be created quickly so that an overall training cost reduction can potentially be realized by the use of our system. The second goal is to accelerate the training process for the trainees through the use of adaptive, multi-modal instructions. With this system, training supervisors have the option of using a wide variety of multi-media options such as 3D animations, videos, text, audio, and interactive simulations to create training instructions. The virtual environment enables workers to practice instructions using interactive simulation and hence reduces the need for practicing with physical components. Our current system is mainly geared toward cognitive skills: training workers to recognize parts, learn assembly sequences, and correctly orient the parts in space for assembly. The VTS was designed to be an affordable training tool. Hence we developed a low cost wireless wand and use an off the shelf head mounted display (HMD).

The VTS system consists of the following three modules:

- ***Virtual Workspace***: The goal of this component of the VTS is to provide the basic infrastructure for multimodal training and to incorporate the appropriate level of physics-based modeling consistent with the operation of a low cost PVE. Virtual Workspace houses the necessary framework to allow manipulation of objects, collision detection, execution of animations, and it integrates the hardware with the software to provide the user an intuitive, easy to use interface to the virtual environment. Virtual Workspace offers three primary modes of training: 3D Animation Mode, which allows users to view the entire assembly via animations; Interactive Simulation Mode, which is a fully user driven mode that allows users to manually perform the assembly tasks; and Video Mode, which allows users to view the entire assembly via video clips. Trainees can switch between these modes at any time with the click of a button.
- ***Virtual Author*****:** The goal of the Virtual Author is to enable the user to quickly create a VE-based tutorial without performing any programming. The Virtual Author package includes a ProEngineer assembly import function. The authoring process is divided into three phases. In the first phase, the author begins with a complete assembly and detaches parts and subassemblies from it, creating an assembly/disassembly sequence. In the process of doing this, the instructor also declares symmetries and specifies the symmetry types. In the second phase, the instructor arranges the parts on a table. In the third and final phase, the instructor plays back the generated assembly/disassembly sequence via animation. During this final phase, text instructions are generated automatically by combining data about collision detection and part motion.
- ***Virtual Mentor*****:** The goal of the Virtual Mentor is to simulate the classical master-apprentice training model by monitoring the actions of the user in the Virtual Workspace and assisting the user at appropriate times to enhance the trainee's understanding of the assembly/disassembly process. If users make repeated errors, then the system will attempt to clarify instructions by adaptively changing the level of detail and inserting targeted training sessions. The instruction level of detail will be changed by regulating the detail of text/audio instructions and regulating the detail level of visual aids such as arrows, highlights, and animations.

We conducted a detailed user study involving 30 subjects and two tutorials to assess the performance of our system. During the first study involving a rocket motor, overall 94.4% steps were performed correctly by the users during the physical demonstration after completing the training. During the second study involving a model airplane engine, overall 97.3% steps were performed correctly by the users during the physical demonstration after completing the training. None of the users tested during these two studies had assembled either a rocket motor or a model airplane engine similar to the ones used in these experiments prior to participating in this study. These results clearly show that our system can be successfully used for training of assembly operations.

We foresee a number benefits from the use of the Virtual Training Studio. First, we believe that the number of personnel required to become trainers will be reduced, since the system will perform the bulk of the training. Second, the system will provide a convenient mechanism for depositing assembly process knowledge into a central repository for later retrieval by trainees, trainers and managers. Third, the VTS will assist instructors in the creation of VE based tutorials and paper based documentation by generating an instruction set from the animation sequence and by providing a convenient way of generating 2D illustrations. Fourth, the system will provide an affordable Personal Virtual Environment. Finally, we believe that the most important benefits will be an accelerated training process as well as a reduced probability of worker error.

4. Automated Generation of Training Instructions for Human Operators

We have developed a system to generate text instructions, including graphical annotations and 3D animations, for human operators at each assembly step. An assembly sequence was considered as input to the system. These instructions were displayed on a large monitor situated at an appropriate viewing distance from the human carrying out assembly operations. By utilizing a speech-recognition module, the system allows the human to use a hands-free interface to sequence the flow of instructions easily. Figures 3(a) and (b) show snapshots from a live footage of a human viewing an animation of placing a central roll bar onto the main chassis and the human subsequently performing the assembly operation, respectively.

Figure 3: (a) Human operator views an assembly instruction; (b) Implementing the instruction

The input to the instruction generation system is a linearly ordered assembly sequence represented in the extensible markup language (e.g., plan.xml). The contents of each step in the plan.xml file will be used to generate multimodal instructions in the form of text, images, and animations.

The language for text instructions will consist of simple verbs such as Identify, Attach, Position, Rotate, Push, Pull, Lift, Lower, Check, Pick, Place, Use, etc. Part features can be used during instruction presentation to guide the human worker in mating the correct panel face with the engine hood before proceeding with attachment. This framework then generates the following text instructions:

Press YELLOW BUTTON **to reset** ROBOT2

Apply FLUX **on edges of** VENT-PANEL-BOTTOM (special instruction)

Pick up VENT-PANEL MANUALLY

Position VENT-PANEL **so that** VENT-PANELNOTECHED-EDGE **aligns with** HOOD-FRONT-EDGE (View Animation)

Attach VENT-PANEL **to** HOOD **so that** VENT-PANELBOTTOM **mates with** HOOD-TOP

Press ORANGE BUTTON **to start** ROBOT2

Press BLACK BUTTON **to switch off** ROBOT2 **when** ROBOT2 **halts**

The information extracted from plan file includes details about the initial scene, labels of the parts and/or subassemblies involved in the assembly operation, and their initial/final postures. This data from each step of the plan is used to invoke a simulation of the assembly operation in a virtual visualization environment, which was developed based on Tundra software. An automated motion planning module interacts with the visualization environment and computes a collision-free motion of a part from its initial posture (e.g., the hood lying in a shelf) to its final posture (e.g., placing of the hood onto the engine compartment of the space frame). Visualization of this computed motion in the visualization environment results in animations that will be appropriately labeled and saved as video clips in a local computer directory. We use multiple random trees, a variation of rapidly-exploring random trees, for the purpose of motion planning.

Usually, when a set of parts is presented to a human worker, he/she can easily distinguish among most of them. However, a few may look similar to each other, leading to confusion about which to pick. We have developed a part identification tool to determine automatically the presence of such similar looking parts and present them in a way that allows a worker to identify and pick the correct part. For this purpose, a similarity metric between two parts is constructed based on the following attributes:

- Part volume and surface area
- Basic shape statistics, such as surface types and their corresponding areas
- Gross shape complexity
- Detailed shape complexity that includes surface area and curvature information.

To present instructions to the human worker, a webpage coded with .php scripts is generated by using the data extracted from plan.xml, filling the appropriate language constructs with this data, and querying the corresponding videos stored in the local folders.

We report the instruction generation results for the chassis assembly. We consider the following assembly sequence as input to the instruction generation system:

1. Position MAIN CHASSIS at POSTURE 1 on ASSEMBLY TABLE
2. Position CENTER ROLL BAR at POSTURE 2

3. Position REAR BRACE at POSTURE 3
4. Position RADIO BOX 8 at POSTURE 4
5. Position RADIO BOX 4 at POSTURE 5

We assume that the assembly location (Posture 1) is selected by the user. Postures 2 to 5 are computed by combining information about the posture 1 and the relative/absolute reference frames extracted from the assembly model. Figure 4 shows snapshots of the instructions — text, graphical annotations, and 3D animations — generated by the system, for each assembly step. Note that a part identification instruction precedes every assembly operation in which a new part must be picked up and attached to the current subassembly.

Figure 4: Generated chassis-assembly instructions

We employed the Tundra environment to generate an augmented reality animation. We built and introduced into Tundra 3D models of assembly-critical resources, including welding robot, welding table, part shelves, sheetmetal-bending machine, waterjet-cutting machine, and laser-cutting machine. Figure 5 shows a snapshot of one perspective on this environment.

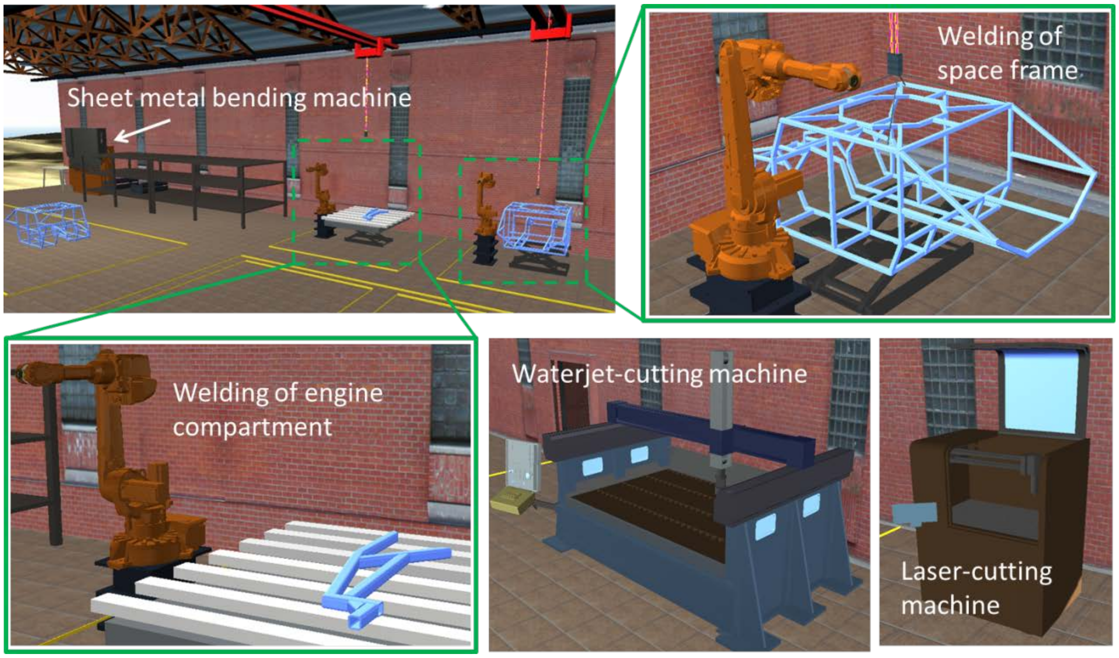

Figure 5: Visualization of different manufacturing and assembly setups

References

1. A. Ananthanarayanan, L. Ehrlich, J.P. Desai, and S.K. Gupta. Design of revolute joints for in-mold assembly using insert molding. *ASME Journal of Mechanical Design*, 133(12):121010, Dec 2011.

2. J.E. Brough, M. Schwartz, S.K. Gupta, D.K. Anand, C.F. Clark, R. Pettersen, and C. Yeager. Virtual Training Studio: A step towards virtual environment assisted training. *IEEE Virtual Manufacturing Workshop*, Alexandria, Virginia, March 2006.

3. J.E. Brough, M. Schwartz, S.K. Gupta, D.K. Anand, R. Kavetsky, and R. Pettersen. Towards development of a virtual environment-based training system for mechanical assembly operations. *Virtual Reality*, 11(4):189-206, 2007.

4. A.S. Deshmukh, S.K. Gupta, M. V. Karnik, and R. Sriram. A system for performing content-based searches on a database of mechanical assemblies. In *ASME International Mechanical Engineering Congress & Exposition*, Orlando, FL, November 2005.

5. A.S. Deshmukh, A.G. Banerjee, S.K. Gupta, and R. Sriram. Content-based assembly search: A step towards assembly reuse. *Computer Aided Design*, 40(2):244-261, 2008.

6. S.K. Gupta, C. J. Paredis, C.H. Wang, R. Sinha, and P.F. Brown. An intelligent environment for simulating mechanical assembly operations. *ASME Design for Manufacturing Conference*, Atlanta, GA, September 1998.

7. S.K. Gupta, C.J. Paredis, and P.F. Brown. Micro planning for mechanical assembly operations. *IEEE Robotics and Automation Conference*, Leuven, Belgium, May 1998.

8. S.K. Gupta, C.J. Paredis, R. Sinha, and P.F. Brown. Intelligent assembly modeling and simulation. *Assembly Automation*, 21(3):215-235, 2001.

9. S.K. Gupta, D.K. Anand, J.E. Brough, R.A. Kavetsky, M. Schwartz, and A. Thakur. A survey of the virtual environments-based assembly training applications. *Virtual Manufacturing Workshop*, Turin, Italy, October 2008.

10. S.K. Gupta, D.K. Anand, J.E. Brough, M. Schwartz, and R.A. Kavetsky. *Training in Virtual Environments: A Safe, Cost-Effective, and Engaging Approach to Training*. CALCE EPSC Press, College Park, July 2008.

11. K. Kaipa, C. Morato, B. Zhao, and S.K. Gupta. Instruction generation for assembly operation performed by human operators. *ASME Computers and Information in Engineering Conference*, Chicago, IL, August 2012.

12. C. Morato, K. Kaipa, and S.K. Gupta Assembly sequence planning by using dynamic multi-random trees based motion planning. *ASME Computers and Information in Engineering Conference*, Chicago, IL, August 2012.

13. A.K. Priyadarshi, S.K. Gupta, R. Gouker, F. Krebs, M. Shroeder, and S. Warth. Manufacturing multi-material articulated plastic products using in-mold assembly. *International Journal of Advanced Manufacturing Technology*, 32(3-4):350--365, March 2007.

14. A.K. Priyadarshi and S.K. Gupta. Algorithms for generating multi-stage molding plans for articulated assemblies. *Robotics and Computer Integrated Manufacturing*, 32(3/4):350-365, 2009.

15. B. Russ, M. M. Dabbeeru, A. S. Chorney, and S. K. Gupta. Automated assembly model simplification for finite element analysis. *ASME Computers and Information in Engineering Conference*, Chicago, IL, August 2012.

16. M. Schwartz, S.K. Gupta, D.K. Anand, J.E. Brough, and R. Kavetsky. Using virtual demonstrations for creating multi-media training instructions. *CAD Conference*, Hawaii, June 2007.

17. M. Schwartz, S.K. Gupta, D.K. Anand, and R. Kavetsky. Virtual mentor: A step towards proactive user monitoring and assistance during virtual environment-based training. *Performance Metrics for Intelligent Systems (PerMIS) Workshop*, Gaithersburg, MD, August 2007.

18. R. Sinha, C.J. Paredis, S.K. Gupta, and P.K. Khosla. Capturing articulation in assemblies from component geometry. In *ASME Design Automation Conference*, Atlanta, GA, September 1998.

19. R. Sinha, S.K. Gupta, C. J. Paredis, P.K. Khosla. Extracting articulation models from cad models of parts with curved surfaces. *ASME Journal of Mechanical Design*, 124(1):106--114, 2002.

20. A. Thakur and S.K. Gupta. Improving performance of rigid body dynamics simulation by removal of inaccessible regions from geometric models. *Computer Aided Design*, 44(12):1190-1204, December 2012.