Science Arts & Métiers (SAM) is an open access repository that collects the work of Arts et Métiers ParisTech researchers and makes it freely available over the web where possible.

This is an author-deposited version published in: https://sam.ensam.eu
Handle ID: http://hdl.handle.net/10985/10280

To cite this version:

Any correspondence concerning this service should be sent to the repository Administrator: archiveouverte@ensam.eu
The development of a physics and constraint-based haptic virtual assembly system

Germanico Gonzalez-Badillo and Hugo Medellin-Castillo
Facultad de Ingeniería, Universidad Autónoma de San Luis Potosí, San Luis Potosí, Mexico

Theodore Lim
Department of Mechanical Engineering, Heriot-Watt University, Edinburgh, UK

James Ritchie
School of Eng and Phys Sciences, Heriot-Watt University, Edinburgh, UK, and

Samir Garbaya
Institut Image, Arts et Metiers ParisTech, Chalon sur Saone, France

Abstract
Purpose – This paper aims to report the development and key features of a novel virtual reality system for assembly planning and evaluation called Haptic Assembly and Manufacturing System (HAMS). The system is intended to be used as a tool for training, design analysis and path planning.

Design/methodology/approach – The proposed system uses the physics-based modelling (PBM) to perform assemblies in virtual environments. Moreover, dynamic assembly constraints have been considered to reduce the degrees of freedom of virtual objects and enhance the virtual assembly performance.

Findings – To evaluate the effectiveness and performance of HAMS, the assembly of various mechanical components has been carried out, and the results have shown that it can be effectively used to simulate, evaluate, plan and automatically formalise the assembly of complex models in a more natural and intuitive way.

Research limitations/implications – The collision detection performance is the bottleneck in any virtual assembly system. New methods of collision shape representation and collision detection algorithms must be considered.

Originality/value – HAMS introduces the use of dynamic assembly constraints to enhance the virtual assembly performance. HAMS also uses features not yet reported by similar systems in the literature. These features include: automatic or manual definition of assembly constraints within the virtual assembly system; the implementation of control panels and widgets to modify simulation parameters during running time to evaluate its influence on simulation performance; assembly data logging such as trajectories, forces and update rates for post-processing, further analysis or its presentation in the form of chronocyclegraphs to graphically analyse the assembly process.

Keywords Assembly, Virtual reality, Assembly constraints, Haptics, Physics-based modelling, Virtual assembly

Paper type Technical paper

1. Introduction

Assembly remains the most studied processes in manufacturing industry because it represents up to 60 per cent of the total cost of the product (Boothroyd, 1992). Computer aided design (CAD) and computer aided assembly planning (CAAP) are still de facto systems to create assembly plans. These systems use various algorithms to automate assembly planning such as feature recognition. In these systems human experience and knowledge are difficult to support intuitively and some factors – such as quality testing, shop floor layout and human ergonomics – cannot be easily taken into account during assembly evaluation (Xia et al., 2013). The lack of physical constraints that the user experiences in the real world, such as collision and interference among objects, is another issue with such systems.

Technologies such as virtual reality (VR) can be used as a tool to enhance the assembly planning and evaluation process by providing a virtual environment (VE) where users can get the feeling of immersion in a real environment and the use of more intuitive cues such as collisions between virtual objects, collisions with obstacles, friction, inertia, restitution, 3D rendering and sound, etc. (Gutiérrez et al., 1998). The evaluation of assembly/disassembly processes in a VE during the early stages of design helps to dramatically reduce the time, cost and material associated with the construction of physical prototypes (Liu et al., 2010).

VEs can be enhanced with haptic technologies to provide the sense of touch. Force-reflecting haptics create a close-loop system, linking visceral data with human motor and cognition. Haptics is therefore ideally suited to understand the interactions and procedures associated with assembly tasks. It allows natural manipulation of objects by enabling the user with the feeling of virtual object collisions, weight and inertia.

The authors acknowledge the financial support from CONACyT (National Science and Technology Council of Mexico) research grant CB-154430 and EPSRC/IMRC grants 113946 and 112430.
This paper describes the research and development of a VR system for assembly evaluation and planning named as Haptic Assembly and Manufacturing System (HAMS). The configuration and flexibility of HAMS allows its use as a platform for evaluating the influence of various simulation parameters on the performance of assembly simulation.

2. Related work

2.1 Virtual assembly
Virtual assembly (VA) is defined as:

(….) utilizing VR technology, computer graphics, artificial intelligence, assembly theory and method, to construct the virtual model of the product and the VE of the assembly layout, and then interactively analyse and simulate the product design result and assembly operation process (Xia et al., 2013).

Several authors have developed VA systems using different methodologies such as automatic feature matching recognition (Jacob et al., 2011; Sato et al., 2011; Viganò and Osorio-Gómez, 2012; Xu et al., 2012); constrained motion (Tching et al., 2010; Zaldivar-Colado and Garbaya, 2009; Gutiérrez et al., 2010; Xia et al., 2012); reuse of CAD assembly constraints (Jayaram et al., 1999; Chamaret et al., 2010; Chen et al., 2010; Cheng-jun et al., 2010); physics-based modelling (PBM) (Garbaya and Zaldivar-Colado, 2009; Gupta et al., 1997; Aleotti and Caselli, 2011; Wan et al., 2004; Seth et al., 2006; Lim et al., 2007a); hybrid approaches (Seth, 2007; Xia et al., 2011); and the use of haptic feedback (Coupee et al., 2001; Ji et al., 2011; Christiaan and Yoon, 2011; Ladeveze et al., 2010; Liu et al., 2010; Bordegoni et al., 2009; Lim et al., 2007a, b). A brief description of the main characteristics of these platforms is presented in Table I.

2.2 Force feedback importance
The importance of force feedback during VA process has been demonstrated by several authors, such as Gupta et al. (1997), Lim et al. (2007b) and Xia et al. (2013), who proved that the assembly time can be reduced by the use of force feedback. An evaluation of two different collision feedback modalities, visual and force feedback, for VA verification was carried out by Sagardia et al. (2012). The results revealed a clear and highly significant superiority of force over visual feedback.

According to Xia et al. (2013), haptics improves the VA performance by reducing completion time, increasing the accuracy to position virtual objects and guiding steadier hand motions along 3D trajectories.

2.3 Key aspects of VA
Four main applications of VA are identified:
1. path planning and optimisation for robotic or human assembly task;
2. design for assembly analysis;
3. maintenance analysis and evaluation; and
4. assembly training.

According to Zhu et al. (2010), the general steps of virtual assembly process planning (VAPP) are: product CAD modelling, interactive VA, automatic generation of a standard assembly process plan based on knowledge and, finally, the assembly plan to be used in the real process. The key activity related to VAPP is the interactive assembly.

The works reported in the literature have revealed that interactive VA can be used as a tool to reduce the DFM/A cycle time. Various techniques and systems have been developed to perform VA. HAMS has been developed as a VA platform to analyse methodologies and algorithms used in VA.

3. System description
HAMS comprises three main modules: physics, haptics and graphics modules, Figure 1. The key features and characteristics of HAMS are described in the following paragraphs.

3.1 Integration
The three main modules of HAMS have been integrated using Microsoft Foundation Classes (MFC) of Visual Studio 2010. The virtual scene 3D rendering is carried out by the graphics module using the Visualization Toolkit libraries (VTK 5.10). This module is responsible of creating the virtual scene and rendering all the objects and information needed in the virtual world.

The physics module enables physical based behaviour of virtual objects. This module uses three physics simulation engines (PSEs): Bullet, PhysX v2.8 and PhysX v3.1. The user is able to select any of them during the system operation. Finally, the haptic module provides the force feedback to enable the sense of touch and kinaesthesia for the user to recognize and manipulate virtual objects. HLAPI from OpenHaptics (v3.0) is used for haptic rendering. Dual haptic interaction is possible via a pair of Phantom Omni haptic devices (Figure 2).

3.2 Model creation
In HAMS virtual objects are imported as STL, OBJ or VTK file formats. An object unique identifier (ID) and material are assigned to each virtual object. The user can select from four different materials: lead, steel, wood or plastic, each with different density. The triangular mesh data describing an object is used to create three different models: the graphics, the physics and the haptics models.

To create the graphics model, a triangle mesh mapper fits the data in such a way that it can be rendered graphically by using the VTK commands. The physics model is required by the PSE for collision detection and dynamic behaviour of virtual objects; it is invisible to the user and can be vastly different from the graphic model. However, for assembly purposes, the physics model must be as geometrically accurate as the graphic model (Gonzalez et al., 2012). The algorithms used in HAMS to generate the collision shape (physics model) from a triangular mesh are listed in Table II.

Finally, for haptic rendering, HAMS uses the higher level graphics attributed methods (HLAPI) from OpenHaptics. HLAPI captures the geometry specified by VTK-OpenGL commands (graphic model) and uses it to perform haptic rendering of virtual objects (haptic model).

3.3 Haptic manipulation and force feedback
During the simulation of the assembly process HAMS has three user motion modalities:
1. the wander;
2. the touch; and
3. the control modes.

Wandering refers to the user’s movements around the virtual scene but without touching or manipulating any object. The touching mode is activated when the user touches an object with the haptic device to explore its shape by force feedback.
In the controlling mode the objects are manipulated by the user through the haptic devices. The movements of the manipulated objects are created as follows:

- The haptic shape is coupled to the physics model trough a mass-spring-damper (MSD) system defined as:

\[
m \ddot{x} = -kx - cx
\]

- The haptic model is moved directly by the position and orientation of the haptic device.
- When the haptic model changes its position, a force \(m \ddot{x}\) is computed using the MSD system.
- The resulting force is then applied to the physics model, producing its movement.

- Finally, the graphics model is updated through a transformation matrix using the position and orientation of the physics model (Figure 3).

### 3.4 Assembly methodology

According to Xia et al. (2013) and Seth et al. (2011), the two most common methodologies to model an assembly process in VEs are: PBM and constraint-based modelling (CBM). In PBM, the virtual objects are dynamic and interact with each other by means of collision response, resulting in a physics behaviour similar to the real world. The contact response between objects prevents the overlapping of virtual objects, enabling the assembly of components (Figure 4). HAMS relies on the PSE to enable PBM in virtual assemblies.
However, PBM has the disadvantage of high computational cost, particularly for non-convex objects, caused by the complexity of computing collisions among such objects.

In CBM the assembly is performed by reducing the degrees of freedom (DOF) of the manipulated object relative to the final assembly position, resulting in a motion constrained part (Figure 5). This methodology has the advantage of low computational cost. When CBM is integrated into a PBM system it is possible to improve its performance and at the same time maintain the realism and intuitive behaviour. In order to improve the performance of VA, dynamic assembly constraints (DAC) have been developed and implemented in HAMS. DAC helps to reduce the computing load of collision detection during the assembly and it also aids the user to reach the final assembly position and orientation of the manipulated part.

3.5 Dynamic assembly constraints
Two types of assembly constraints have been developed and implemented in HAMS:
1 Cylindrical constraints, which are applied to cylindrical features of objects (Figure 4). When this type of constraint is active, the manipulated object can only rotate or translate along the constraint axis.

2 Planar constraints, which are defined by planar faces of an object. When this constraint is active, the manipulated object can only move or rotate on the constraint plane.

In order to use constraints during the VA process, it is required to define a base part where the rest of the parts, named as “manipulated parts”, will be assembled. The assembly constraints of the manipulated parts and the base part must be defined previously.

3.5.1 Assembly constraints definition
DACs can be automatically or manually defined as described in the following paragraphs.

(a) Automatic definition of cylindrical assembly constraints. Cylindrical features of a model can be recognized by analysing the object’s edges. An edge is composed of two indices: \(A\) and \(B\). A closed edge is identified when the index \(A\) of the first analysed edge and the index \(B\) of the last analysed edge are the same. If a closed edge is identified then its size is analysed...
in the three coordinate axes. If two dimensions are similar, within a pre-defined tolerance, then a circular feature has been found and is defined by its diameter, centre and orientation.

Once all the circular features of a model have been identified, a second analysis is performed to find if two circles have the same position and orientation. If this happens, a cylindrical feature has been recognized and its position, depth and orientation are used to define a new cylindrical assembly constraint. Table III presents the results when applying the algorithm to the objects shown in Figure 6.

(b) Manual definition of cylindrical assembly constraints. Cylindrical constraints that are not automatically recognized by the previous algorithm can be manually defined by selecting four points on the cylindrical feature using the haptic device. The first three points must be selected on one edge of the cylindrical feature to compute the centre of the cylinder and the fourth point must be selected on the opposite face of the cylindrical feature to compute the depth and orientation. The assembly constraints are saved as a text file in order to be used whenever the model is loaded.

(c) Manual definition of planar assembly constraints. A planar constraint is defined by selecting four points on a face of the model; these points define the plane where the assembly constraint is intended to be applied. The manually defined planar constraints are also saved as a text file.

3.5.2 Application of assembly constraints
When a virtual object is manipulated by the haptic device, it moves freely. However, when the manipulated object approaches the base part the system evaluates if any assembly constraint is close enough to a similar constraint of the base part. If one pair of constraints is coincident the DAC is activated, the manipulated part is then reoriented and located according to the base part constraint. The DOF of the manipulated part are then reduced according to the type of constraint.

When a manipulated part is released at the desired position and the DAC is active, a kinematic joint between the manipulated part and the base part is created, allowing the creation of subassemblies.

3.6 Data logging
The data generated during the VA process is logged for further processing. HAMS uses three modules for data logging:

1. in the first module the object position, orientation, trajectories, name, haptic cursor position and time are logged;

<table>
<thead>
<tr>
<th>Table III</th>
<th>Results of the automatic assembly constrain definition algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td>Object</td>
<td>Original cylindrical features</td>
</tr>
<tr>
<td>Bearing puller base</td>
<td>5</td>
</tr>
<tr>
<td>Oil pump housing</td>
<td>26</td>
</tr>
<tr>
<td>Valve throttle</td>
<td>3</td>
</tr>
<tr>
<td>Valve housing</td>
<td>8</td>
</tr>
</tbody>
</table>

Figure 6 Automatic cylindrical feature recognition evaluated parts

Notes: (a) Puller base; (b) pump housing; (c) valve throttle; (d) valve housing
in the second module the torques and forces applied to manipulated objects are logged; and
3 the third module logs the system performance parameters such as haptics, physics and graphics update rates.

Each module logs the information during each simulation step. Once the assembly is completed the logged data can be saved as a *.txt or *.csv file format.

(a) Chronocyclegraphs
A method to visualize the logged data is the use of chronocyclegraphs (Lim et al., 2007b; Ritchie et al., 2008; Lim et al., 2010), which are graphic representations of the trajectories followed by the user when manipulating the haptic device. The user movements are symbolized by coloured spheres which represent the modality of the VA (Table IV), with the distance between spheres representing the speed of the movement.

3.7 System increased functionality
HAMS offers different functions that provide increased flexibility and robustness to the system:

• Multimodal graphic rendering that allows the modification of the object rendering type, i.e. the virtual object can be rendered as solid, transparent, invisible, by points, by edges, by wireframe or textured. Another functions includes full camera manipulation around the virtual scene (pan, zoom and rotate), and pre-defined camera viewpoints (front, rear, top, left, right and isometric).
• A haptic properties control panel that includes options to enable or disable the haptic device, select a haptic rendering method or modify the haptic cursor size. In this panel haptic properties such as stiffness, damping, static and dynamic friction can be modified. DAC can be also defined, modified and activated in this panel.
• A physics properties control panel that allows the modification during run-time of physics simulation parameters such as mass, collision shape tolerance, restitution, gravity, friction, simulation time step and PSE selection.
• DACs parameters can be modified through the use of widgets, Figure 7. These widgets can be manipulated via the mouse or the haptic device.

4. System evaluation
Four VA tasks were selected to evaluate the functionality of HAMS, Table V. In order to minimize the complexity of the assembly process, these tasks were performed using one haptic device (one hand configuration). Due to its stability and performance in a preliminary evaluation, Bullet was selected as the PSE (Gonzalez-Badillo et al., 2012). The real assembly of the components was also carried to compare the real and virtual task completion times (TCT). The real objects were manufactured using rapid prototyping techniques in order to create objects with the same features and tolerances than virtual objects. At least five trials were carried out for each assembly tasks (virtual and real).

4.1 Bearing puller
The disassembled bearing puller is shown in Figure 8(a). The VA of this component was carried out without DACs (Figure 8(b)) and with DACs (Figure 8(c)). The task is completed when the last pin is assembled (Figure 8(d)). During the assembly the housing remained static whilst the rest of the objects were dynamic. The chronocyclegraphs corresponding to the assembly of this component are shown in Figure 8(e). The same assembly procedure was carried out with the real parts (Figure 8(f)).

The results of the bearing puller assembly are summarized in Table VI. From these results it is observed that the use of DACs drastically reduce the assembly time and the TCT. The mean haptic force feedback rendered to the user is also presented in Table VI, where it can be observed that force feedback is similar when using PBM and DACs.

4.2 Gear oil pump
The disassembled parts of the gear oil pump are shown in Figure 9(a). The VA was carried out without DACs (only using PBM) (Figure 9(b)) and with DACs (Figure 9(c)). The task is completed when the top bearing is assembled (Figure 9(d)). During the assembly the housing remained static whilst the rest of the objects were dynamic. The chronocyclegraphs of this task are shown in Figure 9(e). The same assembly task was carried out with the real parts (Figure 9(f)).

Table VII shows the results of the gear oil pump assembly. Similarly to the bearing puller assembly, the TCT and individual assembly times are smaller and closer to the

<table>
<thead>
<tr>
<th>Spheres</th>
<th>Modality</th>
</tr>
</thead>
<tbody>
<tr>
<td>Green</td>
<td>Wandering mode, scene recognizing</td>
</tr>
<tr>
<td>Blue</td>
<td>Touching mode, virtual object recognizing</td>
</tr>
<tr>
<td>Red</td>
<td>Controlling mode, virtual object manipulation</td>
</tr>
</tbody>
</table>
real assembly when DACs are used. The force rendered to the user during the assembly of the large gear without DACs is shown in Figure 10(a), whilst the assembly force when using DACs is shown in Figure 10(b). From these results it is observed that when DACs are used, the forces rendered to the user are smaller and more stable.

Figure 11 shows the graphics, physics and haptics simulation update times corresponding to the gear oil pump assembly task. It can be observed that haptics and graphics simulation times tend to remain constant during the whole assembly process. However, the physics simulation time increases with the assembly progress; this is caused by the increment of contact points needed for collision detection in each simulation step. It is also observed that the physics simulation time is smaller when DACs are used.

### 4.3 Pneumatic cylinder

The parts used in this assembly are shown in Figure 12(a). The VA was carried out without DACs (Figure 12(b)) and with DACs (Figure 12(c)). The task is completed when the forth screw is assembled (Figure 12(d)). The rear cap, cylinder and front cap remained static once assembled, whilst the screws and plungers were dynamic. The chronocyclegraphs corresponding to this assembly tasks are shown in Figure 12(e). The same assembly task was carried out with the real parts (Figure 12(f)).

The results of the pneumatic cylinder assembly are summarized in Table VIII. Similarly to the previous assembly tasks, the individual assembly time and TCT are smaller when using DACs. On the other hand, the mean force feedback is similar for both cases (PBM and DACs).
Figure 9 Oil pump assembly

Notes: (a) Virtual objects; (b) HAMS assembly using only physics object; (c) HAMS assembly using DACs; (d) HAMMS assembly; (e) assembly chronocyclegraphs; (f) real assembly

Table VII Results of the gear oil pump assembly

<table>
<thead>
<tr>
<th>Part</th>
<th>Assembly time PBM (min)</th>
<th>Assembly time DACs (min)</th>
<th>Real TCT (min)</th>
<th>Mean force PBM (N)</th>
<th>Mean force DACs (N)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Housing</td>
<td>0:05.2</td>
<td>0:01.9</td>
<td>–</td>
<td>0.27</td>
<td>0.25</td>
</tr>
<tr>
<td>Bottom bearing</td>
<td>0:15.0</td>
<td>0:07.5</td>
<td>–</td>
<td>0.68</td>
<td>0.32</td>
</tr>
<tr>
<td>Large gear</td>
<td>0:13.5</td>
<td>0:04.0</td>
<td>–</td>
<td>0.62</td>
<td>0.47</td>
</tr>
<tr>
<td>Short gear</td>
<td>0:31.0</td>
<td>0:07.8</td>
<td>–</td>
<td>0.78</td>
<td>0.34</td>
</tr>
<tr>
<td>Top bearing</td>
<td>0:11.4</td>
<td>0:09.7</td>
<td>–</td>
<td>0.88</td>
<td>0.35</td>
</tr>
<tr>
<td>TCT</td>
<td>1:29.3</td>
<td>0:41.7</td>
<td>0:13.1</td>
<td>0.65</td>
<td>0.35</td>
</tr>
</tbody>
</table>

Figure 10 Logged force when manipulating and assembling the large gear

Notes: (a) PBM without DACs; (b) with application of DACs
4.4 Manual bench vice

The disassembled bench vice is shown in Figure 13(a). The VA of the bearing puller was carried out using only the PBM (Figure 13(b)). The task is completed when the pin is assembled (Figure 13(c)). The objects remained static after being assembled. The chronocyclegraphs of this assembly are shown in Figure 13(d). The real assembly was also performed, Figure 13(e). Table IX summarizes the results of the bench vice assembly. The results show that virtual TCT is greater than real TCT as it was expected.

4.5 User perception

In order to validate the system functionality, a set of experiments was carried out by several users, which were asked to perform the gear oil pump assembly following a specific order, Figure 14. First, all the users were trained in the system during 20 min to minimize learning effects. When using only PBM, a mean TCT of 3:29.4 min with a standard deviation of 57.8 s was obtained. When using DACs, a mean TCT of 2:20.7 min with a standard deviation of 31.0 s was measured. It can be observed that the use of DACs enhances the VA process by reducing the TCT.
The users were asked to qualitatively evaluate the system and the VA experience by assigning values to the following parameters:

- **Ease of controlling the haptic device**, from 0 if controlling the device was very difficult, to 4 if it was very easy and comfortable.
- **Ease of performing the VA**, from 0 if the assembly process was very difficult, to 4 if the process was very easy.
- **Virtual objects stability**, from 0 if the virtual objects were jumping or flying in the scene, to 3 if the objects were very stable.
- **Realism of the process**, from 0 if virtual objects behaviour is far from real objects, to 3 if virtual objects behave like real objects.
- **Collision feedback**, from 0 if the collision feedback through the haptic device is not perceptible, to 3 if all the collisions of manipulated objects are accurately rendered by the haptic device.
- **TCT perception**, from 0 if the user perceived VA TCT to be larger than real TCT, to 3 if virtual TCT is perceived to be smaller than real TCT.

The results of the user’s qualitative evaluation are shown in Figure 15. It can be observed that collision feedback is the best evaluated parameter, which means that collisions are accurately rendered by means of the haptic device. The TCT of the VA is perceived to be larger than the TCT of the real assembly, which is confirmed by the experimental measurements. The rest of the
qualitative parameters have a satisfactory evaluation, which means that users have a good overall perception of HAMS. Regarding the overall perception of HAMS, most of the users considered that:

- they felt confident while doing the VA;
- the system is useful and interesting;
- the virtual objects move according with the movements of the hand; and
- virtual objects have a good accuracy respect to collision response.

Finally, the users suggested that the simulation process must be faster and the haptic interaction must be improved by using haptic devices with more DOF and contact points, e.g. a haptic glove.

5. Discussion

The key features of HAMS and other haptic VA systems proposed by different authors are presented in Table X. It can be observed that HAMS includes several characteristics that make it different from similar systems. One of the outstanding characteristic is a hybrid approach (PBM and CBM) to perform the assembly process with DACs that can be manually or automatically created. Unlike similar systems, HAMS is the only system where the user can manipulate the camera viewpoint using the haptic device, i.e. without the need to release the device and use the mouse or keyboard to perform this operation. Another important feature of HAMS is the possibility to automatically create a subassembly without the need of manually updating parameters such mass, inertia, etc. The subassembly is automatically created once the manipulated part is assembled in the target position.

6. Conclusions

A new HAMS has been presented and described. The system comprises three main modules: haptics, graphics and physics. The outstanding characteristics of HAMS are: the use of DACs to enhance the VA process; automatic or manual definition of assembly constraints within the VA system; control panels and widgets to modify simulation parameters during run-time; assembly data logging, different collision shape representation algorithms and the possibility to select the PSE to perform the VA. These characteristics make HAMS a complete, flexible and robust VA platform for planning, evaluation, simulation and training of assembly tasks in a more natural and intuitive way.
The case studies have proved the effectiveness and functionality of HAMS to perform haptic VA process of any object previously designed in a CAD system and exported as STL, OBJ or VTK format file. Also, the results have shown that in the VE the TCT is greater than the corresponding real assembly TCT. However, the results obtained in the multi-users tests have probed that the use of DACs can reduce the virtual TCT.

Future work considers the evaluation of the algorithm used to create the collision shape, and the effect of using different PSEs on the VA performance. The evaluation of HAMS as a tool for assembly planning and training is also considered as part of the future work.

Table X Comparative of HAMS with similar VA systems

<table>
<thead>
<tr>
<th>Features</th>
<th>HAMS</th>
<th>SHARP (Seth et al., 2006)</th>
<th>HVAS (Xia et al., 2011)</th>
<th>HIVex (Bhatti et al., 2009)</th>
<th>HIDRA (Coutee et al., 2001)</th>
</tr>
</thead>
<tbody>
<tr>
<td>End purpose</td>
<td>Training, path planning,</td>
<td>Path planning and</td>
<td>Path planning and</td>
<td>Cognitive learning</td>
<td>Assembly and path planning</td>
</tr>
<tr>
<td></td>
<td>assembly evaluation and simulation analysis</td>
<td>assembly evaluation</td>
<td>assembly evaluation</td>
<td>and training</td>
<td></td>
</tr>
<tr>
<td>Assembly modelling technique</td>
<td>PBM and CBM (hybrid)</td>
<td>PBM and CBM (hybrid)</td>
<td>PBM</td>
<td>PBM</td>
<td>PBM</td>
</tr>
<tr>
<td>Method used to create assembly constraints</td>
<td>Manual and automatic based on model features</td>
<td>Manual and automatic Form CAD assembly constraints</td>
<td>Not supported</td>
<td>Manually defined only in orthogonal axis</td>
<td></td>
</tr>
<tr>
<td>Logged data</td>
<td>Manipulated part position,</td>
<td>Record and play module</td>
<td>Positions, forces and</td>
<td>Completion time and</td>
<td>Not mentioned</td>
</tr>
<tr>
<td></td>
<td>orientation, time, force, torque</td>
<td></td>
<td>torques and simulation update rates</td>
<td>percentage of correct parts selected</td>
<td>For demonstration purposes</td>
</tr>
<tr>
<td>Visualization of resulting assembly path</td>
<td>Chronocyclegraphs</td>
<td>Swept volumes</td>
<td>Discrete points or swept volumes</td>
<td>Not supported</td>
<td>Dual haptics but only one hand manipulation</td>
</tr>
<tr>
<td>Dual handed haptic device configuration</td>
<td>Supported</td>
<td>Supported</td>
<td>Not supported</td>
<td>One handed, data glove attached to phantom</td>
<td></td>
</tr>
<tr>
<td>Import of CAD models</td>
<td>Geometry imported using STL,</td>
<td>Geometry imported using voxel models or B-Rep</td>
<td>Automatic data integration interface</td>
<td>VRLM, geometry, assembly and physical data</td>
<td>Import geometry using VRLM format file</td>
</tr>
<tr>
<td></td>
<td>OBJ and VTK file formats</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Method to create the collision shape</td>
<td>Trimesh, convex decomposition, convex elements and primitives</td>
<td>B-Rep, VPS</td>
<td>Convex decomposition</td>
<td>Smart collision</td>
<td>Convex decomposition</td>
</tr>
<tr>
<td>Complex models</td>
<td>Supported</td>
<td>Supported</td>
<td>Supported</td>
<td>Supported</td>
<td>Primitives or convex elements</td>
</tr>
<tr>
<td>Collaborative assembly</td>
<td>Local</td>
<td>Supported, network</td>
<td>Not mentioned</td>
<td>Not supported</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>module</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Possibility to define materials</td>
<td>Supported</td>
<td>Not mentioned</td>
<td>From CAD model</td>
<td>Supported</td>
<td></td>
</tr>
<tr>
<td>Full camera manipulation</td>
<td>By haptic device or mouse</td>
<td>Not mentioned</td>
<td>Not mentioned</td>
<td>By keyboard</td>
<td></td>
</tr>
<tr>
<td>Possibility to create subassemblies</td>
<td>Supported</td>
<td>Supported, needs the redefinition of model properties</td>
<td>Not supported</td>
<td>Not supported</td>
<td>Not supported</td>
</tr>
</tbody>
</table>

The references for the text include:


References


**Corresponding author**

**Germanico Gonzalez-Badillo** can be contacted at: germanico.gonzalez@uaslp.mx