ELSEVIER

Automation in Construction 16 (2007) 232–241

AUTOMATION IN CONSTRUCTION

www.elsevier.com/locate/autcon

Evaluation of standard product models for supporting automated erection of structural steelwork

Vineet R. Kamat [a,*], Robert R. Lipman [b,1]

[a] Department of Civil and Environmental Engineering, University of Michigan, 2340 G.G. Brown, 2350 Hayward, Ann Arbor, MI 48109-2125, USA
[b] Computer Integrated Building Processes Group, National Institute of Standards and Technology, 100 Bureau Drive,
Stop 8630, Gaithersburg, MD 20899-8630, USA

Accepted 8 May 2006

Abstract

Automation is being increasingly explored as a possible solution for safely increasing productivity in structural steelwork erection. A piece of automation equipment has no intrinsic knowledge of the steel erection process. Thus, geometric and spatial member information, and the motion sequences that must be executed to install steel members must both be programmed into the equipment. This research investigates the extent to which the CIMsteel Integration Standards (CIS/2) can support automated steelwork erection. Algorithms to interpret steel member geometry and spatial configuration from CIS/2 files were designed. Then, using inverse kinematics principles from robotics literature, a kinematically smart crane capable of accepting robot-like instructions was implemented in 3D virtual reality. The crane was programmed to utilize the algorithms to automatically extract member information from CIS/2, and to use that information to compile assembly instructions for erecting the structure in the virtual world. Based on the obtained results, it was found that CIS/2 does encapsulate the basic geometry and position and orientation of steel members in a format that, after geo-referencing, can be used to support automated steelwork erection. However, several processing steps are necessary to compute the information needed from the CIS/2 model for the process description of erection needed by the automation equipment.

Keywords: Automation; CIS/2; Construction; Computer graphics; Emulation; Structural steelwork

1. Introduction

Structural steel frame erection, especially in high-rise construction, requires a significant amount of skilled labor, and is inherently a very dangerous occupation. According to the American Institute of Steel Construction (AISC), a 25% reduction in the time required to erect a steel frame structure is needed for the steel construction industry to remain competitive [1]. Since any desired increase in productivity must be accompanied by increases in worker safety and reliability, automation of structural steelwork erection is being actively explored as a possible solution [2].

A piece of automation equipment such as a robotic crane has no intrinsic knowledge of the process (e.g., steel erection) it automates. Thus, geometric and spatial information about a component (e.g., steel member), and the motion sequences that must be executed to move that component from a staging area to its installed final location must both be programmed into the equipment. The equipment must minimally know where a steel member in question is currently staged, and what the final installed position and orientation (pose) of the member is in the erected structure. Based on these two pieces of information, automation equipment can use inverse kinematics (IK) algorithms to first move its grippers to the current location of the member, and then transport it to the pose where it is to be installed.

In automated steel construction, the position and orientation of steel members in a temporary staging area is project and site dependent, and thus cannot be automatically determined beforehand. User intervention is required to interpret and communicate such information to automation equipment. The final in-place spatial configuration (position and orientation) of a steel member, however, can be conceptually extracted automatically from a three-dimensional product model of the

---

* Corresponding author. Tel.: +1 734 764 4325; fax: +1 734 764 4292.
E-mail addresses: vkamat@umich.edu (V.R. Kamat), robert.lipman@nist.gov (R.R. Lipman).

[1] Tel.: +1 301 975 3829; fax: +1 301 975 5433.

0926-5805/$ - see front matter © 2006 Elsevier B.V. All rights reserved.
doi:10.1016/j.autcon.2006.05.001

structure being erected [3]. The presented research evaluates this hypothesis and investigates the extent to which the CIMsteel Integration Standards (CIS/2) can specify product descriptions capable of supporting automated erection of structural steelwork.

2. CIMsteel integration standards release 2

The CIMsteel Integration Standards Release 2 (CIS/2) is a product data model for structural steel. It is the result of the pan-European Eureka EU130 CIMsteel Project and has been endorsed by the American Institute of Steel Construction (AISC) as the technical basis for their Electronic Data Interchange initiative. The goal of CIS/2 is to create a seamless and integrated flow of information among all project participants involved in the construction of steel framed structures. These participants include steel designers, analysts, detailers, fabricators, and erectors.

The CIS/2 product data model deals with information about the steel structure throughout its analysis, design, detailing, and fabrication life cycle. The geometry of the structure is only one property of the product data model. Other attributes of the structure include how parts are combined into assemblies and structures, analysis loads and reactions, material types, connection details, associations between members and drawings, and modification history.

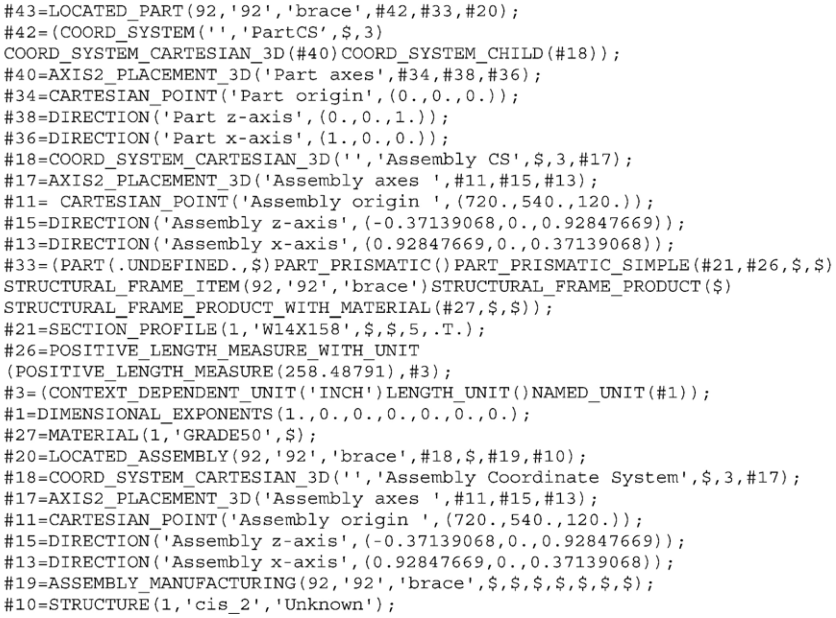
The product data model is defined by a schema that details how all the information in the CIS/2 standard is represented and how each entity relates to each other. It also defines rules for using various entities and the information those entities contain. STEP (ISO 10303:1992), the Standard for Exchange of Product Model Data, is the technical basis used to represent CIS/2 information. STEP is a worldwide effort to develop mechanisms for the exchange and sharing of engineering data. The format of a CIS/2 file that is imported or exported by steel-related software packages is specified in a standard way to facilitate the exchange of information among various applications. The STEP Part 21 implementation method defines the physical structure of a file.

2.1. Structure of the CIS/2 product data model

A CIS/2 file can support three primary models of structural steel information—the analysis model, the design model, and the manufacturing model. Information on any of these models can coexist in the same CIS/2 file. An analysis model of a steel structure consists of nodes and elements and supports several static and dynamic analysis methods. A design model represents a steel structure as a design assembly to allow member and connection design. A design assembly can be partitioned into other simpler design assemblies and eventually into design parts and design joint systems. The design parts and joint systems respectively form the conceptual representations of a basic piece of steel and a basic joint system.

A manufacturing model in CIS/2 represents a steel structure as manufacturing assemblies for the purpose of detailing, production planning, and manufacturing. In a manufacturing model, located assemblies are comprised of located parts and located joint systems that respectively represent a basic physical piece of steel and a basic physical joint system. All located items can be combined into larger located assemblies that eventually define a complete structure.

Fig. 1 shows a sample of a CIS/2 file for a part with a location. The file is represented in the standard STEP Part 21 format. Each CIS/2 entity instance is assigned a number indicated by a pound (#) sign. The name of the CIS/2 entity then appears in upper case letters. Every entity has a number of fields that contain text strings, numeric values, Boolean values, references to other entities, or null values indicated by a dollar sign. The indentation has been added to show the hierarchy and relationships between the various entities.



```
#43=LOCATED_PART(92,'92','brace',#42,#33,#20);
#42=(COORD_SYSTEM('','PartCS’,$,3)
COORD_SYSTEM_CARTESIAN_3D(#40)COORD_SYSTEM_CHILD(#18));
#40=AXIS2_PLACEMENT_3D('Part axes',#34,#38,#36);
#34=CARTESIAN_POINT('Part origin',(0.,0.,0.));
#38=DIRECTION('Part z-axis',(0.,0.,1.));
#36=DIRECTION('Part x-axis',(1.,0.,0.));
#18=COORD_SYSTEM_CARTESIAN_3D('','Assembly CS',$,3,#17);
#17=AXIS2_PLACEMENT_3D('Assembly axes ',#11,#15,#13);
#11= CARTESIAN_POINT('Assembly origin ',(720.,540.,120.));
#15=DIRECTION('Assembly z-axis',(-0.37139068,0.,0.92847669));
#13=DIRECTION('Assembly x-axis',(0.92847669,0.,0.37139068));
#33=(PART(.UNDEFINED.,$)PART_PRISMATIC()PART_PRISMATIC_SIMPLE(#21,#26,$,$)
STRUCTURAL_FRAME_ITEM(92,'92','brace')STRUCTURAL_FRAME_PRODUCT($)
STRUCTURAL_FRAME_PRODUCT_WITH_MATERIAL(#27,$,$));
#21=SECTION_PROFILE(1,'W14X158',$,$,5,.T.);
#26=POSITIVE_LENGTH_MEASURE_WITH_UNIT
(POSITIVE_LENGTH_MEASURE(258.48791),#3);
#3=(CONTEXT_DEPENDENT_UNIT('INCH')LENGTH_UNIT()NAMED_UNIT(#1));
#1=DIMENSIONAL_EXPONENTS(1.,0.,0.,0.,0.,0.,0.);
#27=MATERIAL(1,'GRADE50',$);
#20=LOCATED_ASSEMBLY(92,'92','brace',#18,$,#19,#10);
#18=COORD_SYSTEM_CARTESIAN_3D('','Assembly Coordinate System',$,3,#17);
#17=AXIS2_PLACEMENT_3D('Assembly axes ',#11,#15,#13);
#11=CARTESIAN_POINT('Assembly origin ',(720.,540.,120.));
#15=DIRECTION('Assembly z-axis',(-0.37139068,0.,0.92847669));
#13=DIRECTION('Assembly x-axis',(0.92847669,0.,0.37139068));
#19=ASSEMBLY_MANUFACTURING(92,'92','brace',$,$,$,$,$,$,$);
#10=STRUCTURE(1,'cis_2','Unknown');
```

Fig. 1. Located part in a CIS/2 file.

The top-level entity is a LOCATED_PART (#43) that associates a PART (#33) with a coordinate system (#42). The LOCATED_PART also refers to a LOCATED_ASSEMBLY (#20). The PART refers to a SECTION_PROFILE (#21), a LENGTH (#26), and a MATERIAL (#27). The LOCATED_ASSEMBLY also refers to a COORD_SYSTEM (#18), an ASSEMBLY (#19), and a STRUCTURE (#10). These statements describe that the part, which is a W14X158 wide-flange section with a given length, has a location in an assembly that in turn is located in the structure.

According to the CIS/2 schema, all located parts must be unique. However, there can be multiple references to a single part. For a simple framed structure, each beam or column is a located part. However, there need be only one referenced part for members that have the same section profile and length. Multiple located parts can refer to the same located assembly. For example, a beam and the clip angles and gusset plates at its ends can all be part of the same located assembly. Features such as hole layouts, copes, and miters can be applied to a part with the LOCATED_FEATURE_FOR_LOCATED_PART entity. This entity refers to the specifications and dimensions of the feature, the coordinate system to locate the feature on the part, and the located part on which they are applied. A joint system can be either a bolted or welded connection and is located on an assembly in a similar manner.

3. Research initiative

By knowing where a steel member to be erected is currently located, and what its final installed position and orientation in the frame is going to be, a piece of automation equipment can use inverse kinematics algorithms to first move its grippers to the current location of the member, and then transport it to the location in the frame where it is to be installed. This concept of autonomous pick and place of a steel beam has been demonstrated using the NIST RoboCrane and a simple two-column frame with ATLSS connectors, though the necessary part transformations were not derived from CIS/2 data [4]. Typically, additional spatial information such as the geometry of the partially completed structure and other existing obstructions is also required to be input to the equipment so that interference detection algorithms can work with the IK algorithms to compute a collision free path for the equipment and the member [5].

The location of a steel member in its staging area on a jobsite is specific to the site itself and to the contemplated construction plan. This information cannot be determined automatically by automation equipment beforehand. The final position and orientation of each steel member in the completed structure can, however, be conceptually deciphered from a rich geometric product model of the structural frame. CIS/2, for instance, can allow the definition of all steel parts, prefabricated assemblies, and connecting joint systems that are to be erected in a steel structure.

The nested coordinate systems can define the location of steel parts in assemblies that in turn can be located in larger assemblies, ultimately culminating in defining the location of the entire frame in a local coordinate system. The spatial configuration (position and orientation) of each steel member in its final installed location can then be computed by transforming the model's coordinate frame into a global geo-referenced coordinate system where the structure is to be erected. The CIS/2 model can thus allow the automatic extraction of spatial steel member information that is required to format instructions to automation equipment [3].

In creating a 3D animation of a construction operation with process level detail, the encountered problem is very similar to that faced by robotics engineers. Conceptually, each piece of equipment on a virtual construction site can be thought of as a robot performing a certain construction task. Just as real robots perform specified tasks using real resources, virtual pieces of equipment are required to perform construction tasks on the computer using virtual resources. The striking analogy between the two arises from the fact that both robots and virtual equipment pieces do not have any intrinsic knowledge about performing assigned tasks and need to be programmed in order to perform particular motion sequences.

In terms of information needs, therefore, virtual pieces of construction equipment are very similar to industrial robots. The two may differ in context (real vs. virtual) and shape, but in both cases, engineers are basically interested in trying to manipulate a multiply articulated structure (i.e., a kinematic chain) and move a particular component (e.g., steel member) from one pose to another. The design and implementation of geometric information extraction algorithms and their evaluation in a 3D virtual world thus presented an effective way of validating the applicability of CIS/2 for supporting automated erection of structural steelwork.

4. Extraction of spatial member information from CIS/2 product models—technical approach

A CIS/2 product model contains two pieces of information that can directly support automation tasks: the geometry of steel members comprising a structure, and the pose (position and orientation) of each steel member in the structure. Both pieces of information are necessary to formulate accurate instructions to automation equipment erecting steel members. The final pose of a member dictates the target pose of the equipments' end effecter as well as the equipments' target articulation. The geometry of a member in turn is required to determine the grip locations on a member and to compute a collision free path for the member and equipment from the staging area to the installed location. The focus of this research was thus on evaluating the extent to which these two pieces of information can be extracted from a CIS/2 file, and investigating the level of data modification required before the extracted information can be used to formulate member installation instructions for automation equipment.

A two-step approach was adopted to achieve these objectives. In the first step, the NIST CIS/2 to VRML translator [6] was used to convert CIS/2 files to their corresponding VRML (Virtual Reality Modeling Language) representation. Then in the second step, algorithms were designed to extract member geometry and pose information from the VRML files.

The intermediate conversion of CIS/2 files to VRML was adopted for the following reasons:

1) Although CIS/2 supports explicit description of geometry, most CIS/2 parts are expressed using implicit descriptions of geometry based on a transverse section profile and a longitudinal length through which the profile is swept [3]. The VRML translator converts all implicit geometry to regular geometry facilitating member shape extraction.
2) Implementation of a VRML file parser that can traverse the described scene graph to extract member geometry (shape) and pose (transformation) information was relatively straightforward compared to the implementation of a full-fledged CIS/2 file parser.
3) The VRML translator provides a visual interface to the underlying CIS/2 data thereby providing a means to visualize the represented steel structure in a 3D virtual world.

4.1. Steel shape geometry

In a CIS/2 manufacturing model, an assembly is a collection of located parts and joint systems. For instance, a column, base plate, and welded connection could be located in the structure as a collection using the LOCATED_ASSEMBLY entity. There are no CIS/2 entities that can indicate which located parts and joint systems are contained in a located assembly. Instead, the located parts and joint systems indicate the located assembly they are part of. Using these relationships between defined CIS/2 entities, the CIS/2 to VRML translator generates a scene graph defining the parent–child relationship using corresponding VRML nodes.

Any VRML node can have a user-defined name using the DEF construct. Once a node is defined, multiple instances of it can be used in the scene graph with the USE construct. Using DEF and USE, geometry and other nodes can be reused, greatly reducing the size of a VRML file and the time required to process it inside a browser for display. Since reusing geometry nodes is more efficient than explicitly creating the geometry for an object that already exists, this method is extensively used in mapping CIS/2 entities to a VRML scene graph [7].

While this approach is favorable from the perspective of file size and browser processing time, it is not very amenable to individual member geometry extraction required for supporting automation. When defined unit length geometry nodes are reused to instantiate specific instances of steel members, the geometry of the members is effectively being defined implicitly in terms of the stack of scaling transformation nodes that appear between the member and the referenced geometry node. This presents an identical problem to that encountered when attempting to deduce three-dimensional geometry coordinates of a member from the implicit description of the geometry in a CIS/2 file defined using a section profile and swept longitudinal length.

In order to address this issue and represent member geometry in a way that facilitates the extraction of individual three-dimensional member coordinates, an option was added to the CIS/2 to VRML translator which, when selected, collapses the scaling transformation nodes that would otherwise appear above a member's geometry. The shape of each member is entirely encapsulated in individual geometry nodes using VRML Indexed Face Sets. Since the geometry of each steel member is represented separately even if two members are identical (i.e., same section and length), the size of the resulting file is relatively larger compared to the former scenario. However, this does not present any particular constraints for the purposes of this study.

The geometry of each member is described relative to a local origin. Fig. 2 graphically presents the geometry descriptions for an arbitrary steel column and beam. In both cases, the profile of the section is coplanar with the *YZ* plane and the length is represented along the positive *X*-axis. In the case of columns, the local origin corresponds to the lower end of the column when located in the overall structure. In the case of beams, the local origin in all tested files corresponds to the beam end closer to the global origin when located in the overall frame. However, this is only a convention and not a requirement of CIS/2.

4.2. Steel shape position and orientation

Similar to geometry, CIS/2 defines a hierarchical system of steel member locations (positions and orientations) such that an entity is located with respect to another parent-level entity to which it belongs. For example, a part may be located relative to an assembly that in turn could be located with respect to another higher level assembly. The hierarchical description can continue until the location of all entities in the complete structure is defined relative to a chosen global coordinate system [8].

In CIS/2 manufacturing models, each location is typically defined using the COORD_SYSTEM_CARTESIAN_3D entity whose attributes are a three-dimensional point defining the origin, and a set of vectors defining the three-dimensional orientation of the local coordinate system. The final position and orientation of each member in the overall structure is thus a function of the member's geometry in the leaf coordinate system and the combined effect of all parent coordinate systems higher up in the hierarchy.

In the case of a converted VRML file, this same effect is achieved using a stack of geometric transformation nodes that each corresponds to a specific local coordinate system. The geometry of a member is located with respect to a leaf transformation node that in turn has another transformation node as its parent and so on until the top level transformation node represents the global coordinate system in which the represented structure is located.

The position of a part in the overall structure defined in a VRML file can thus be determined by computing the combined

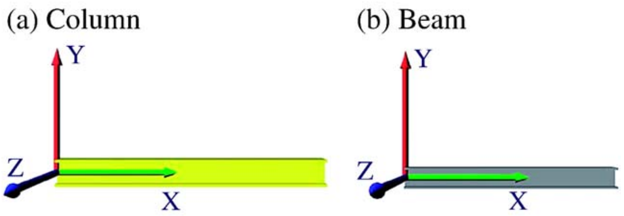


Fig. 2. VRML geometry in a local coordinate system for arbitrary column and beam.

effect of all translation fields that appear in the hierarchy of transformation nodes above the definition of the member's geometry node. The important point to note is that the combined effect of all translation fields appearing above a member's geometry only places the geometry's local origin (Fig. 2) at the computed location in global space. It does not necessarily orient the member correctly in global space.

In order to determine a part's correct global orientation, the combined effect of all rotation fields in the transformation hierarchy is computed. The vectors defining rotations in a CIS/2 file are represented in a converted VRML file using the axis-angle representation [9]. In this format, a rotation is described by an arbitrary axis in three-dimensional space and the amount by which the local coordinate system is to be rotated about that defined axis. The axis itself is defined as a vector constructed from the local origin to the three-dimensional coordinate specified in the description.

The CIS/2 to VRML translator was modified to collapse as many hierarchical geometric transformations (positions and orientations) as possible during the conversion process to facilitate the parsing of the generated VRML file. The structure of the resulting VRML scene graph is graphically presented in Fig. 3. In the case of a CIS/2 manufacturing model, for instance, all geometric transformations were collapsed down to the individual assembly level, i.e., each primitive CIS/2 assembly was directly positioned and oriented in the VRML file's global space. This facilitated the parsing and interpretation of member position and orientation.

4.3. Modification of extracted data to support automation

As described above, the parsing and interpretation of a VRML file converted from a CIS/2 model provides the basic information on member geometry and pose that is required to support automation. However, the extracted information must be pre-processed in several steps before it can be input in instructions to automation equipment, and in this case, to a virtual crane.

4.3.1. Reconciliation of steel member and automation equipment coordinate systems

The position of steel members in a CIS/2 file's global space refers to the top-most coordinate system of that file only. In other words, the top-most coordinate frame in a CIS/2 model is arbitrary and the CIS/2 capability for geo-referenced coordinate systems is typically not used. Thus, recalling the previous example, if we input to a robotic crane that a particular steel beam is located at position (8,9,2) in the CIS/2 file's global space, the crane will not be able to resolve that information into an actual location on the jobsite unless the crane's local coordinate system is reconciled with the file's global coordinate frame.

Such reconciliation is necessary to locate both the equipment and the steel members in the same coordinate space to allow the equipment to determine the position of members relative to itself. In this study, the reconciliation was achieved by coinciding the interpreted CIS/2 global coordinate frame with the coordinate system of the crane's three-dimensional virtual world. Once the crane and the steel members were placed in the same coordinate space, the computation of the relative distance and orientation between them was straightforward to compute.

4.3.2. Coincidence of member origin with automation equipment gripper

Another subtle but important step that must be accomplished to convert the extracted product model data into a format that can be readily used to compile automation instructions is the reconciliation of the local coordinate systems of the automation equipment's gripper(s) and the member being erected. The computations in this step are specific to the geometry of the equipment, the shape of the member being erected, and the grip locations on the member.

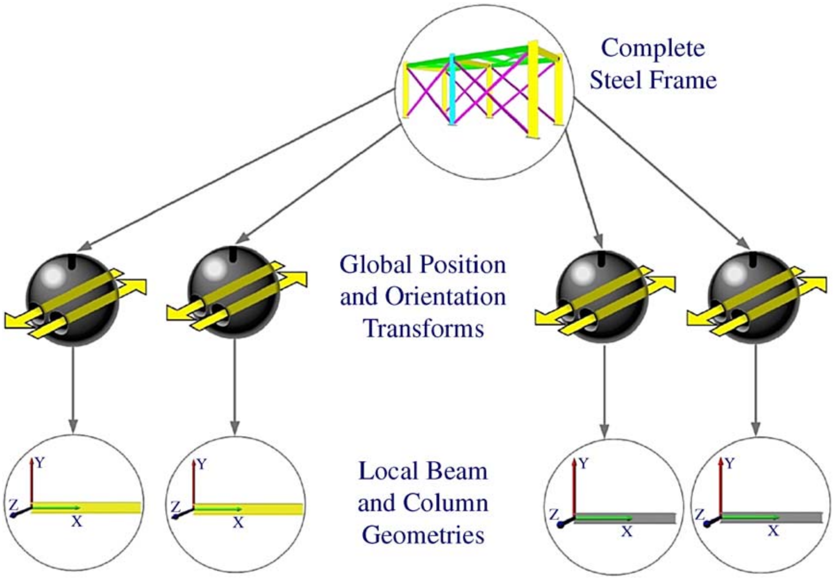


Fig. 3. Local VRML geometry nodes with collapsed transformations.

For example, consider the VRML geometry of a CIS/2 column shown in Fig. 2(a). As pointed out earlier, the local origin of the geometry corresponds to the lower end of the column when located in the overall structure. However, the grip location for a crane on a steel column is the upper end of such a member, which is located along the *X*-axis in the geometry description. When a column is gripped at its upper end and hoisted, it gradually attains a vertical orientation after which it can be moved to its installation location and lowered into place and secured. Such an erection sequence is graphically presented in Fig. 6.

Since a steel member can be arbitrarily oriented when lying flat and ready to be picked in a staging area, moving an automation equipment gripper to a location other than the member's geometric local origin introduces additional computational complexities. Moreover, any algorithm that can be designed to offset the discrepancy cannot be generalized because of the arbitrariness in the orientation of members in the staging area.

This issue can be solved by modifying a steel member's geometric representation so that its local origin coincides with the centroid of its grip locations. In the case of a column, this is achieved by translating the geometry along the negative *X*-axis by an amount equal to the column length. This moves the upper end of the column (the grip location) to the geometry's local origin and is graphically presented in Fig. 4(a).

In the case of a steel beam, the grip location for a crane is typically in the center of the member. A beam is hoisted, moved, and lowered into position while maintaining its horizontal orientation. Such a sequence is presented in Fig. 7. Since the VRML geometry nodes of CIS/2 beams orient geometry in a way similar to that of columns (Fig. 2(b)), the required translation to coincide the grip location on a beam with the local geometry origin is equal to half the beam length along the negative *X*-axis. This is graphically presented in Fig. 4(b).

4.3.3. Conversion of axis-angle member rotation to Euler angles

As noted in Section 4.2, the rotations of all hierarchical parts and assemblies in CIS/2 and the corresponding geometry and transformation nodes in the converted VRML are defined using the axis-angle notation. Collapsing all the rotation values appearing in the node hierarchy above the geometry of a member, the global axis-angle rotation of a member can be computed. While the axis-angle notation allows for a compact, unambiguous description of member orientation, it cannot be used promptly in formatting commands for automation equipment [10].

Any piece of automation equipment cannot be directly instructed to move an object from one arbitrary rotation to another. The IK algorithms that compute a path for a robot's gripper from an object's source position to its desired destination accomplish this by computing a set of concatenated axial rotations that are applied to individual links in the robot's kinematic chain. In order to allow IK algorithms to compute elemental axis-aligned rotations for automation equipment linkages, the source and target orientation of a member being erected must also be defined or converted to three individual rotation values in the *XYZ* coordinate system. Rotations described using this notation are commonly called Euler angles [9]. In this study, the algorithm described in [11] was adopted to convert the global orientation of a steel member from an axis-angle notation to corresponding Euler angles.

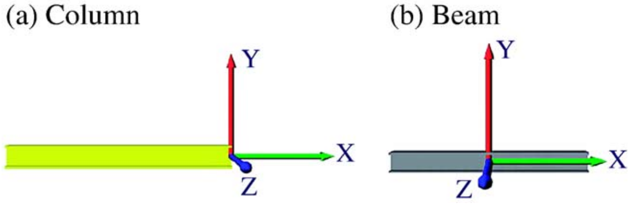


Fig. 4. Coinciding member origin with equipment gripper.

Thus, the parsing and interpretation of CIS/2 information contained in a converted VRML file, and its subsequent modification, yield the following information for each steel member contained in the represented structural steel frame:

- The name of the member
- The geometry of the member in local space
- The position of the member in global space
- The orientation of the member relative to each coordinate axis in global space

Together, these automatically extracted data comprise a significant portion of the information required by a robotic piece of equipment to automatically erect a steel member by transporting it from a staging area to its installed location along a computed collision-free path.

5. Graphical simulation of automated steelwork erection

In order to validate the designed approach, a three-dimensional articulated crawler-mounted crane was programmed to receive extracted CIS/2 geometry and pose information as input and utilize IK algorithms to erect a described steel structure inside a virtual world. In particular, the articulated crane implemented in the KineMach add-on for the VITASCOPE visualization system [12] was used to validate the efficacy of the proposed approach.

5.1. Implementation of virtual robotic crane emulator

KineMach implements "smart," generic pieces of virtual construction equipment and provides simple parametric text statements that can be used to issue task-level instructions to that equipment to visually depict the performance of construction work. Currently implemented generic pieces of equipment include a tower crane, a crawler mounted lattice boom crane, a crawler mounted backhoe, and a highway dump truck. KineMach's statements are designed to represent the common construction tasks that the corresponding real pieces of construction equipment perform on real construction sites. Many statements are designed using standard, documented, and commonly used terminology. For instance, in the case of the implemented crawler crane, most KineMach statements have a direct one-to-one correspondence with standard crane hand signals used in real crane operations.

KineMach deciphers a communicated instruction and uses IK techniques to compute the elemental motions involved in performing a construction task. KineMach then applies the computed elemental motions to the equipment's components (e.g., boom, cabin, etc.) to depict the performance of the requested task in a virtual world. For instance, each time a steel shape is to be placed, the following statement can be communicated to KineMach requesting the instantiated virtual crane (Crane1) to perform the assigned task.

Crane1. PutThatThere LA2550 6 (0,6,0) (8,9,2) 45 90 4 60;

The PutThatThere statement instructs an instantiated crawler-mounted crane (Crane1) to perform a lift inside the virtual world. In particular, the statement instructs a crane to pick up the specified object (LA2550) and install it at the indicated target location (8,9,2) in the given amount of time (60). The statement also requires that the height of the object being lifted (6) be provided for clearance computations, and the hook attachment point (0,6,0) of the object be specified for identification of the grip locations. In addition, the object's target horizontal (45) and vertical (90) rotations (i.e., orientations) and the minimum clearance to maintain (4) during the animated operation must also be indicated in the statement's arguments.

When KineMach processes each such statement, it first computes the elemental crane motions necessary to accomplish the task and apportions the total task time (last argument in the statement—60) to the individual elemental motions using proportions determined empirically. KineMach then generates elementary motion—describing geometric transformations and forwards them to the visualization engine to graphically depict the operation in the virtual world. Thus, as noted in Section 3, the virtual crane has no intrinsic knowledge about the construction task being animated. Similar to a piece of automation equipment, the crane needs to be programmed with specific geometric and spatial information in order to perform particular motion sequences.

5.2. Emulated erection of steel column and beam

The efficacy of the proposed information extraction approach was evaluated by obtaining the argument values for the virtual crane's PutThatThere statements from the geometry and pose information of steel members interpreted from a CIS/2 VRML file. As described in Section 4, the name, geometry, final position, and final orientation (horizontal and vertical) of each steel member comprising a structure can be extracted from its CIS/2 representation. This information was used to formulate the numerical input arguments for several PutThatThere instructions issued to the instantiated virtual crane. This mapping of CIS/2 information to KineMach argument values is graphically presented in Fig. 5.

In particular, the name of a member (e.g., LA2550) to be erected is directly available from the extracted CIS/2 data. The height of the member can easily be computed by constructing a bounding box around the CIS/2 geometry extracted as described in Section 4.1. The grip location(s) can be computed by reconciling two local coordinate systems as described in Section 4.3.2. The target position and the orientations for the member are obtained from the CIS/2 data as described in Section 4.2. Thus, a significant portion of the information required to formulate installation instructions for the virtual crane can be readily extracted from a CIS/2 model.

Based on the arguments provided to the PutThatThere statement, KineMach computes a collision-free path for a member

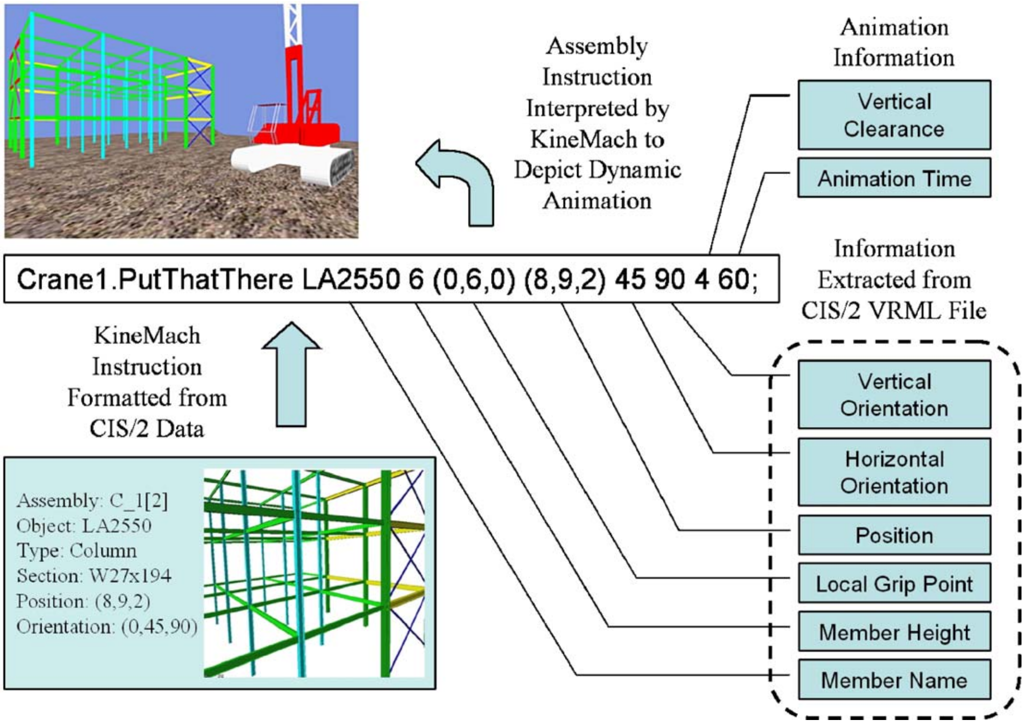


Fig. 5. Mapping CIS/2 geometry and pose information to KineMach arguments.

from its staging area to the location where it is to be installed. The vertical clearance argument (4 units in Fig. 5) is used in the path computation. For example, KineMach will raise the hook of the crane so that the member being hoisted is at least "vertical clearance" units higher than the other CAD objects it will pass over in the virtual environment.

It must be pointed out that KineMach cranes are only "smart" enough to compute an interference-free path for a member being virtually erected. The tool does not encapsulate any other domain intelligence. For instance, if one or both columns that support a particular beam have not yet been erected in the virtual world, and an instruction to install the pertinent beam is issued to the crane, KineMach will transport the beam from its staging location to its destination regardless of the supporting columns. If the columns are missing, the beam will erroneously appear to float without any support.

As noted in Section 6 ahead, the order in which steel members described in a CIS/2 file are to be erected cannot be readily deciphered from the files unless erection sequences are accurately captured during authoring. Thus, the order in which steel members are to be erected by KineMach inside the virtual world must be explicitly indicated by the user. Figs. 6 and 7 present animation snapshots displaying the erection of a steel column and a steel beam, respectively. The snapshots presented are not successive computer frames observed during animation. The captured frames are displayed as a filmstrip merely to convey a sense of motion.

Video clips of the animations depicted in the figures are available for download from the first author's website at http://pathfinder.engin.umich.edu/.

6. Future work

The KineMach crane used to validate the proposed information extraction techniques in a three-dimensional virtual world is a regular, crawler-mounted, lattice boom crane (Figs. 6 and 7). A real robotic crane that automates structural steelwork erection is likely to have a different shape and form than a regular crane such as that animated in this study. While the use of a regular crane model to demonstrate the utility of information extracted from CIS/2 does not discount the merits of this research in any way, an interesting and natural extension of this work would be the implementation of a virtual crane that resembles a real robotic crane. An example of such a piece of equipment to emulate would be the NIST ROBOCRANE [13].

As demonstrated in this study, the CIS/2 product model encapsulates the final installed configuration (position and orientation) of steel members in a structure. This information is useful in instructing automation equipment about where to install a particular member. However, as noted in Section 1, the CIS/2 model captures no information on where steel members are temporarily staged on the jobsite. Such information would be useful in automatically providing source information for robotic equipment. One way this problem could be addressed is by extending the CIS/2 model to encapsulate project specific construction planning and site layout information. Another parallel direction to address the same problem would be in the area of local positioning and sensing so that automation equipment can detect the position and orientation of steel members that exist in the staging area and formulate appropriate instructions to pick the members from this temporary location for

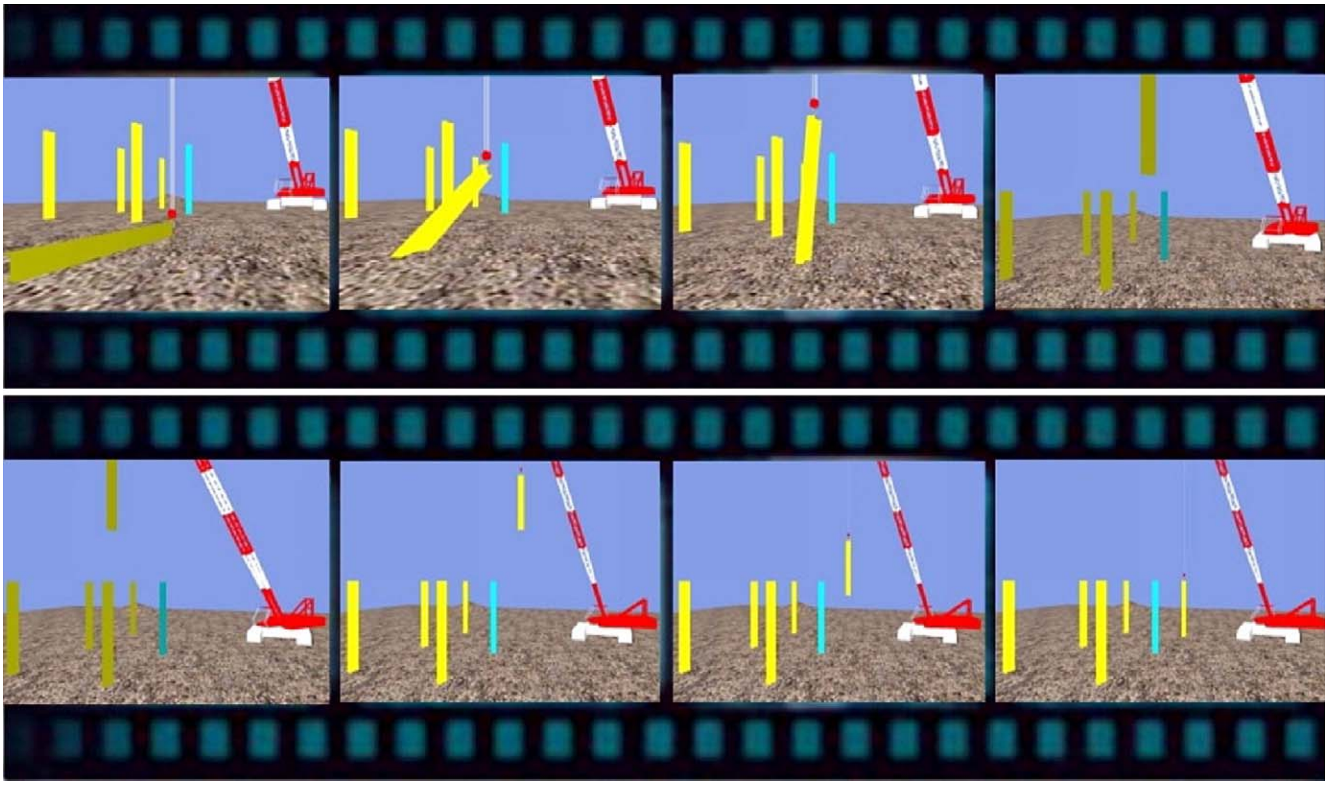


Fig. 6. Animation snapshots of steel column erection.

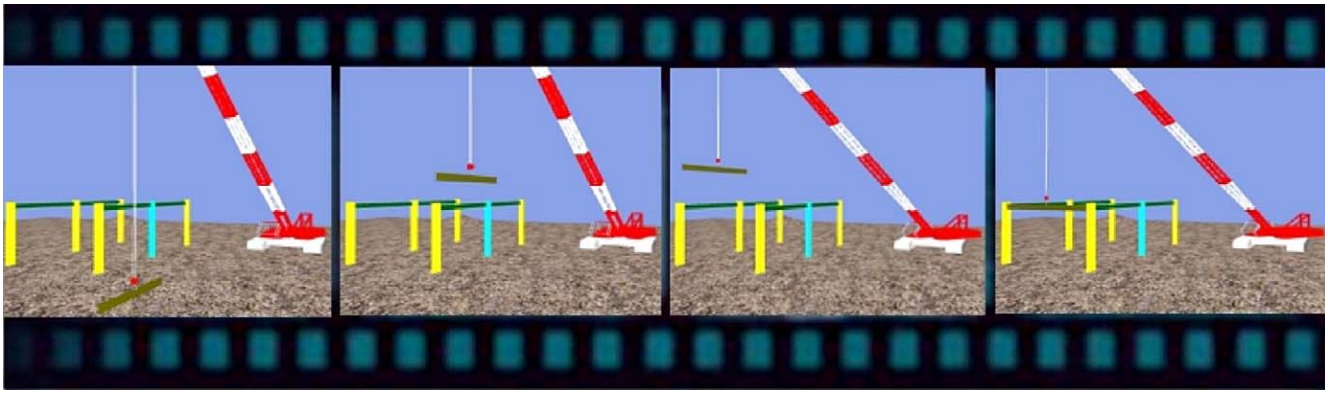


Fig. 7. Animation snapshots of steel beam erection.

installation. Ongoing work at NIST is attempting to address the problem using this second approach [2,14]. Information from the automatic detection of the position and orientation of steel members could be used to populate an extended CIS/2 model for representing the temporary staging of members on the jobsite.

The CIS/2 product model contains many weakly specified uses of character strings as identifiers, labels, and descriptions [3]. For example, in the definition of steel parts and assemblies, there are several placeholders in definitions that could be used, for instance, to indicate the structural function (e.g., beam, column, brace, etc.) of a defined part. Such information could be very useful for automation equipment and the algorithms that control their functions. However, because the use of such CIS/2 strings is not enforced (i.e., weakly specified), no assumptions can be made by the file parsing algorithms. Two possible ways to address this issue would be to strongly specify the use of information rich string tags in future versions of the product model, or to develop a set of recommended tagging practices at the user level so that misinterpretation of expected information is minimized [3].

Recommended user-group level practices could also be developed to capture erection sequences accurately during the authoring of CIS/2 files. Such information could be extracted and used by robotic equipment to erect the encapsulated structure in proper sequence. Sequence information can be expressed in CIS/2 with the ZONE_OF_STRUCTURE entity. Information from fabrication software such as what material has been shipped to the job site could also be used to determine the erection sequence. In addition, CIS/2 allows the indirect determination of the parts that a LOCATED_JOINT_SYSTEM entity connects through the LOCATED_PART_JOINT entity. This information could be used to determine which assemblies are connected together in the structure and could help further automate the erection process and eliminate the problem of not having columns in place before a beam connected to them is erected.

Future research could also explore CIS/2's ability to support automated erection of larger steel components comprising of several assemblies. For instance, a large truss in CIS/2 can be represented as a collection of assemblies, even though there may not be any single CIS/2 entity that encapsulates these assemblies together into a cohesive truss. Interpreting the physical characteristics of the truss and formatting automation instructions based on extracted information on individual assemblies would be an interesting avenue for future work.

7. Summary and conclusions

Automation equipment such as a robotic crane has no intrinsic knowledge of any construction processes such as those involved in steel erection. Thus, in automated steelwork erection, geometric and spatial information about a steel member to be erected, and the motion sequences that must be executed to move that member from a staging area to its installed location must both be programmed into the equipment. The position and orientation of steel members in a temporary staging area is project and site dependent, and thus cannot be automatically determined beforehand. However, the final in-place spatial configuration (position and orientation) of a steel member can be conceptually extracted automatically from a three-dimensional product model of the structure.

The presented study evaluated this hypothesis and investigated the extent to which the CIS/2 standard model can specify product descriptions capable of supporting automated steelwork erection. A kinematically "intelligent" crane capable of accepting robot-like instructions was implemented in a three-dimensional virtual world. Algorithms to automatically interpret steel member geometry and their spatial configuration from CIS/2 files were designed. The virtual crane was then programmed to use these algorithms to automatically extract steel member information from a CIS/2 file. The extracted information was used to compile automated assembly instructions required to erect the structure inside the virtual world using the crane.

Based on the emulation results, it was found that CIS/2 does encapsulate the basic geometry and pose of steel members in a format that, after appropriate geo-referencing, can be readily used to support automated erection of structural steelwork. However, several intermediate information processing steps were found to be necessary before the extracted data can be used to program specific instructions for the automation equipment. In addition, it was found that while CIS/2 itself defines an information-rich product model and not intended to have a role in automation, the semantics of many statements could be improved to strengthen the standard's applicability for supporting structural steelwork automation.

Acknowledgments

The presented work has been supported by the National Science Foundation (NSF) through Grant CMS-0408538. The authors gratefully acknowledge NSF's support. Any opinions, findings, conclusions, and recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of the NSF.

References

[1] A. Lytle, K. Saidi, W. Stone, J. Gross, Report of the NIST Workshop on Automated Steel Construction, 19th International Symposium on Automation and Robotics in Construction (ISARC), NIST, Gaithersburg, MD, 2002, pp. 247–254.

[2] A.M. Lytle, K.S. Saidi, N.A. Scott, Developments with the NIST automated construction testbed, 21st International Symposium on Automation and Robotics in Construction (ISARC), September 21–25, 2004, Jeju, Korea, 2004.

[3] K.A. Reed, Role of the CIMsteel integration standards in automating the erection and surveying of constructional steelwork, 19th International Symposium on Automation and Robotics in Construction (ISARC), NIST, Gaithersburg, MD, 2002, pp. 15–20.

[4] K.S. Saidi, A.M. Lytle, N.A. Scott, W.C. Stone, Developments in automated steel construction, NISTIR 7264, National Institute of Standards and Technology, Gaithersburg, MD, 2005.

[5] Y.K. Cho, C.T. Haas, Rapid geometric modeling for unstructured construction workspaces, Journal of Computer-Aided Civil and Infrastructure Engineering, Blackwell Publishers, Malden, MA, 2003, pp. 242–253.

[6] R.R. Lipman, K.A. Reed, Visualization of structural steel product models, Electronic Journal of Information Technology in Construction, vol. 8, Royal Institute of Technology, Stockholm, Sweden, 2003, pp. 43–50.

[7] R.R. Lipman, Mobile 3D visualization for construction, 19th Intl. Symposium on Automation and Robotics in Construction (ISARC), NIST, Gaithersburg, MD, 2002, pp. 53–58.

[8] ISO 10303-42 (2000), Industrial automation systems—Product data representation and exchange—Part 21: Integrated generic resource: Geometric and topological representation, ISO/IEC, Geneva, Switzerland (with Technical Corrigendum 1, 2001).

[9] H. Goldstein, Classical Mechanics, 2nd ed., Addison-Wesley, Reading, MA, 1980.

[10] J.J. Craig, Introduction to Robotics: Mechanics and Control, 2nd ed., Addison-Wesley, Reading, MA, 1989.

[11] K. Shoemake, Euler angle conversion, in: Paul S. Heckbert (Ed.), Graphics Gems IV, Academic Press Professional, Toronto, Canada, 1994.

[12] V.R. Kamat, J.C. Martinez, Dynamic 3D Visualization of Articulated Construction Equipment, Journal of Computing in Civil Engineering, vol. 19, No. 4, American Society of Civil Engineers, Reston, VA, 2005.

[13] J.S. Albus, R.V. Bostelman, N.G. Dagalakis, The NIST ROBOCRANE, A Robot Crane, Journal of Robotic Systems, Wiley, New York, NY, July 1992.

[14] D.E. Gilsinn, G.S. Cheok, C. Witzgall, A. Lytle, Construction Object Identification from LADAR Scans: An Experimental Study Using I-Beams, NISTIR, vol. 7286, National Institute of Standards and Technology, Gaithersburg, MD, 2006.