Building Simulation 2005

Ninth International IBPSA Conference
Montréal, Canada
August 15-18, 2005

POSITION UNCERTAINTY IN SPACE SCENE RECONSTRUCTION FOR SIMULATION-BASED LIGHTING CONTROL

Georg Suter, Oguz Icoglu, Ardeshir Mahdavi and Bojana Spasojevic
Department of Building Physics and Building Ecology
Institute of Architectural Sciences
Vienna University of Technology
Vienna, Austria

ABSTRACT

Dynamic, three-dimensional models of existing facilities that are updated automatically based on data provided by advanced sensing technologies appear increasingly feasible. Simulation-based lighting control is one potential application domain, where, besides information regarding external obstructions and sky conditions, accurate, up-to-date models of space boundaries, openings, and, possibly, furniture and light fixtures are required to support controller decision-making. In this paper, we describe a procedure that uses spatial reasoning and constraints to support the reconstruction of space scene models that are sufficient for lighting simulation. Specifically, the procedure addresses the problem of position uncertainty in sensor data. Its potential is demonstrated in a test setting. Daylight performance indicators are computed for an as-built model of a test space and compared with those obtained for a model generated with the procedure.

INTRODUCTION

There is a growing interest in the continuous monitoring of building performance, motivated by the need to improve building operation. Recent advances in sensing technologies have made this goal increasingly feasible. Although commercial solutions are available for monitoring and preventative maintenance of mission-critical HVAC, lighting, security or circulation systems, these cover only a small fraction of the information that may be useful in assessing building performance. What is still lacking is a systematic approach to collecting and organizing dynamic building state information into comprehensive models that update themselves automatically and are available to applications, such as quantity surveying, inventory management, or building controls. The context for the work presented in this paper is simulation-based control of lighting systems, where a controller relies on lighting simulations of a target indoor environment to explore multiple control options (Mahdavi 2001). Most detailed lighting simulation involves the definition of space scenes, that is, space boundary geometries (walls, floors, ceilings, and openings) as well as corresponding surface attributes such as absorptance and reflectance. Optionally, equipment such as furniture or light fixtures may be included to either increase accuracy or to consider artificial lighting. An important requirement concerns the validity of input models. Lack of validity may cause system failure or misleading feedback. Conditions include space boundary faces forming a valid polyhedron and equipment disjoint from each other and the space boundary. Furthermore, certain relations must be maintained. For example, windows should be properly contained within their walls.

In the following, we describe a procedure for the reconstruction of space scenes based on sensed object location information. Tracked objects include space boundaries and equipment, such as tables, cabinets, or light fixtures. The procedure employs spatial reasoning and object contact constraints to address position uncertainty in object location data and generate models that are valid for lighting simulation and minimize deviations from as-built space scenes. To demonstrate the potential of this approach for simulation-based lighting control, daylight performance indicators are computed for an as-built model of a test space and compared with those obtained for a test space model generated with the procedure.

OBJECT LOCATION SENSING

Detailed sensor-driven space scene models imply the capability to track objects of interest over space and time. Current location-sensing systems often rely on tags, that is, small markers mounted objects. In contrast to tag-less technologies such as laser scanners, one benefit of tags is the encoding of object identification and, sometimes, additional information. We use TRIP, a vision-based location sensing that relies on a combination of visual markers or tags and digital cameras as sensors (Lopez de Ipina et al. 2002, Icoglu et al. 2004). TRIP uses computer vision techniques to derive tag locations. One limitation is that it requires line-of-sight between tags and sensors. Tags are inexpensive as they can be generated on a black-and-white laser printer. Circular tag partitions define the ID code, position of synchronization sector (starting point), actual length of the radius of the tag in millimeters and even-parity bits. Location information provided by TRIP includes object position $\boldsymbol{p}$ and normal vector $\boldsymbol{n}$. The latter is perpendicular to the tag

plane and points towards the camera. Object location data is generally prone to measurement errors. In case of TRIP, these are due to factors such as image resolution and the angular accuracy of a pan-tilt unit, on which the camera is mounted to increase the scanning range. Errors include deviations in position (ε_p) as well as orientation or normal readings (ε_n) (Ellis 1991). In the following, we only consider position errors. Position uncertainty is defined as the radius $r=\varepsilon_p$ around a sensed or nominal position within which the true position is likely to lie. In the TRIP system, ε_p for 76% of position readings is 0.2m or less, and 0.3m or less for 88% of readings. Orientation errors are more difficult to address because even small deviations may introduce significant location uncertainty. As a simplification, we thus assume orthogonal space scenes, in which a sensed tag normal $\boldsymbol{n}$ is interpreted as equivalent to the closest basis vector (or its reverse) of a global cartesian coordinate system.

CONVERSION FROM TAG- TO BOUNDARY-BASED SPACE SCENE REPRESENTATIONS

We assume that each sensed object's identity is known and partial models of manufactured objects such as openings, light fixtures and furniture exist. From a maintenance perspective, the number of tags to mark objects should be minimized as well as the a priori information needed to unambiguously define a model. We use the term tag representation to refer to a model composed of a combination of sensed object location and partial models of manufactured objects. By itself, tag representations have little explicit structure and are thus of limited immediate use to lighting simulation and most other applications, which require boundary representations of space scenes. For example, the shape of space boundaries or the relation between space boundaries and openings is unknown in a tag representation. A procedure for the conversion of tag-based to boundary-based building representations is described in Suter (2004) and Suter et al. (2005). The procedure derives space boundary faces as well as relations between openings and space boundary faces by spatial reasoning. It is limited, however, because it assumes perfect location data. To address that limitation, a space object position adjustment procedure is described next that relies on the original procedure and ensures non-interfering space objects.

OBJECT POSITION ADJUSTMENT

Overview

The approach is to generate a preliminary model first, using the original conversion procedure and assuming ideal sensor data. This results in a boundary representation of a space scene where individual objects may interfere with each other. Space boundary faces, related openings as well as equipment are repositioned in a second phase to resolve these interference conflicts. That is, for each sensed position $\boldsymbol{p}_i$, an object position adjustment procedure determines a new position $\boldsymbol{p}'_i$. Furthermore, each is $\boldsymbol{p}'_i$ chosen such that contacts to other objects are maximized. For example, tables are often grouped together and usually in direct contact to the floor. Similarly, openings may be repositioned with respect to the floor, or cabinets with respect to the floor and one or more walls. For clarity, we explain this procedure in two dimensions. The extension to three dimensions is analogous and straightforward.

Position uncertainty zone representation

A rectangular uncertainty zone U with round corners of radius ε_p can be constructed with the sensed position $\boldsymbol{p}$ of equipment E (furniture, light fixture), whose shape is approximated by a bounding box with dimensions w_x and w_y, and the position error ε_p (Figure 1a). The actual equipment E'' at (unknown) position $\boldsymbol{p}''$ is contained in U. The adjusted E' at position $\boldsymbol{p}'$, which is determined by the procedure, needs to be contained in U as well. Rounded corners of uncertainty zones are ignored to facilitate the derivation of the spatial relations between uncertainty zones without sacrificing too much accuracy (Figure 1b).

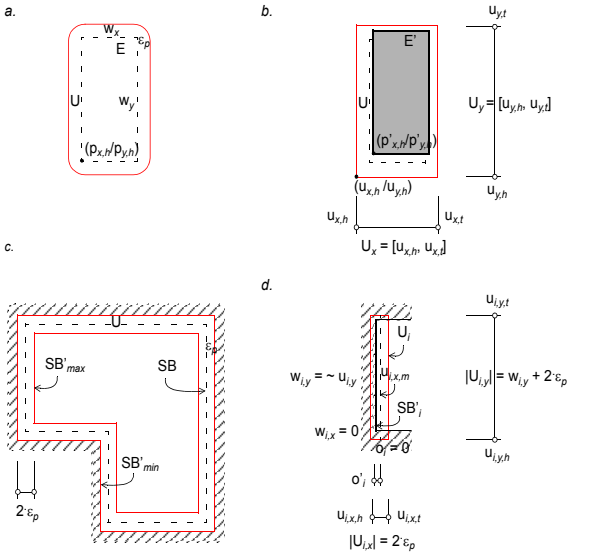

Figure 1. Representations for position uncertainty. a. Space equipment uncertainty zone, b. Approximation of space equipment uncertainty zone, c. Uncertainty zone defined by minimum and maximum space boundary, d. Approximation of a space boundary face.

Uncertainty zones for space boundary faces are determined similar to those for equipment. Each plane, in which a face of a sensed space boundary *SB* is embedded, may be offset parallel to the plane normal by ε_p, resulting in minimum and maximum space boundaries (Figure 1c). These are similar to the minimum and maximum material conditions for

engineering parts in tolerancing theories (see, for example, Requicha 1983). As each face is represented by one tag and sensed independently of other faces, individual space boundary face uncertainty zones are generated based on the face bounding box and ε_p. An offset parameter o'_i is derived by the procedure for each space boundary face (Figure 1d). As offsets are applied to all faces of a space boundary polyhedron simultaneously, the shape of each face is adjusted in addition to its repositioning; offsetting is a common operation in solid modeling systems.

A rectangular uncertainty zone U for an equipment or space boundary object is represented as closed intervals in x and y directions, that is, $U_x = [u_{x,h},u_{x,t}]$ and $U_y = [u_{y,h},u_{y,t}]$ (an h label refers to the head, a t label to the tail of an interval). Representing uncertainty zones as intervals is useful for classifying spatial relations and tightening contact zones formed by the intersection of adjacent uncertainty zones. Note that uncertainty zones and object approximations for equipment and space boundary elements are represented for uniform treatment.

Uncertainty zone relations

The next step is to derive spatial relations among uncertainty zone pairs. These are indicators for actual relations among objects. If the intersection of two uncertainty zones is empty, then the two objects associated with them are guaranteed not to touch each other. The converse is not necessarily true: uncertainty zones may have a non-empty intersection, but the objects contained by them may or may not be in contact. The intersection of two uncertainty zones is referred to as a contact zone. Two objects may touch only in their contact zone.

Table 1. Intervals ($U_{A,x} = [u_{A,x,h}, u_{A,x,t}]$, $U_{B,x} = [u_{B,x,h}, u_{B,x,t}]$) for testing the spatial relation between uncertainty zones U_A and U_B.

	$T_{1,x}$	$T_{2,x}$
$U_{A,x} \cup U_{B,x} =$ $[u_{A,x,h}, u_{A,x,t}]$ or $[u_{B,x,h}, u_{B,x,t}]$	$[u_{A,x,h}, u_{B,x,t}]$	$[u_{B,x,h}, u_{A,x,t}]$
$U_{A,x} \cup U_{B,x} =$ $[u_{A,x,h}, u_{B,x,t}]$	$[u_{A,x,h}, u_{B,x,t}]$	n/a
$U_{A,x} \cup U_{B,x} =$ $[u_{B,x,h}, u_{A,x,t}]$	$[u_{B,x,h}, u_{A,x,t}]$	n/a

Table 2. Classification of relation types based on test outcomes in each direction.

	y-test = true	y-test = false
x-test = true	xy-relation	x-relation

Table 2. Classification of relation types based on test outcomes in each direction.

	y-test = true	y-test = false
x-test = false	y-relation	infeasible

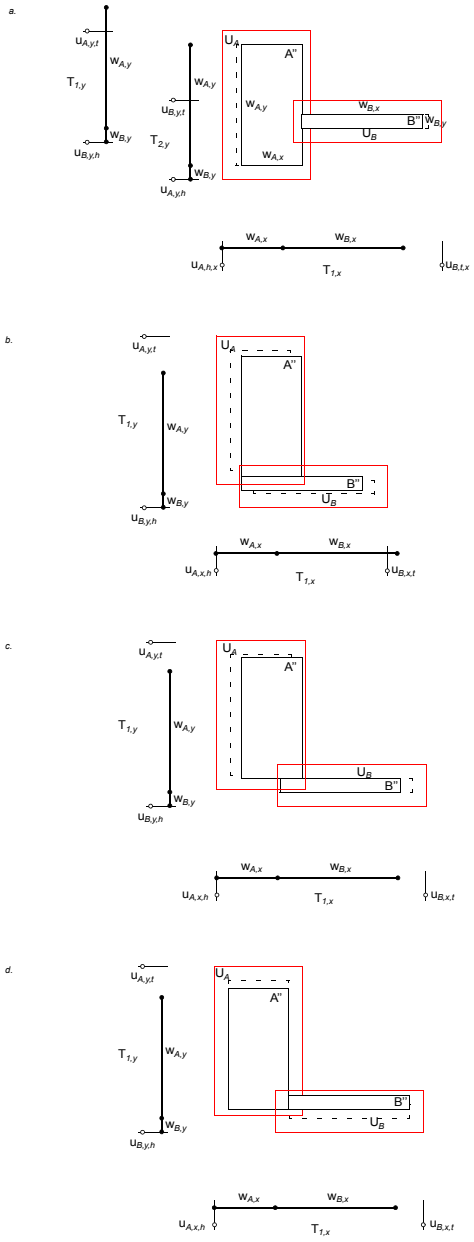

Figure 2. Spatial relations between object pairs. a. x-relation, b. y-relation, c., d. xy-relation.

As we assume orthogonal space scenes, the direction of a spatial relation between uncertainty zones can be classified. Given an uncertainty zone pair U_A and U_B with non-empty intersection, the relations are determined as follows (Figure 2):

- Test intervals are generated for the x direction according to Table 1. Depending on the outcome of $U_{A,x} \cup U_{B,x}$, this results in one or two test intervals, $T_{1,x}$ and/or $T_{2,x}$. Test intervals for the y direction are analogous.

- The length of test intervals $|T_{1,x}|$ and/or $|T_{2,x}|$ are compared with the sum of the object bounding box dimensions, $w_{A,x} + w_{B,x}$. The test fails if $|T_{1,x}| < w_{A,x} + w_{B,x}$ (single test interval), respectively, $|T_{1,x}| < w_{A,x} + w_{B,x}$ and $|T_{2,x}| < w_{A,x} + w_{B,x}$ (two test intervals). The test for the *y* direction is analogous.
- Tests in both directions are evaluated according to Table 2. Possible relations are *x*-relation, *y*-relation, and *xy*-relation.

Note that *xy* relations are relevant only for object position adjustment if neither *x* nor *y* relations are present.

Uncertainty zone adjacency graph

An uncertainty zone adjacency graph can be created for a space scene using the above relation detection and classification procedure. This involves testing each object with all other objects in a space scene. An uncertainty zone adjacency graph is labeled and directed. The graph is connected in case of three-dimensional representations of space scenes, and may be disconnected in two-dimensional representations (Figure 4a). Vertices represent uncertainty zones, edges the type of spatial relation, *x*, *y* or *xy*, between an uncertainty zone pair. The orientation of an edge reflects the ordering of an uncertainty zone pair with respect to the relation direction. When the order is not obvious, as illustrated in Figure 3, a test is performed which is similar to that for determining the relation between uncertainty zones.

Once an uncertainty zone adjacency graph is generated, a set of paths is extracted from it. Each path includes a maximum number of successive vertices that are connected by edges with identical labels (Figure 4b). Paths are generated until each edge in the graph is included in at least one path. These serve as a basis for the tightening of contact intervals, which is described next.

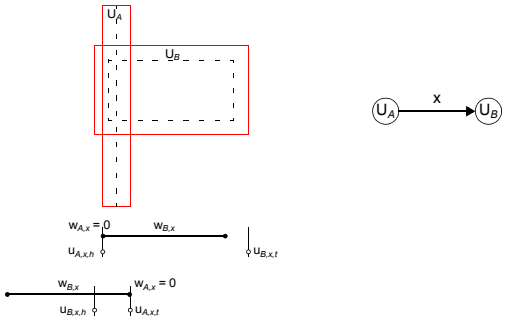

Figure 3. Ordering adjacent uncertainty zones.

Tightening of contact intervals within paths

A contact interval $C_{i,i+1,x}$ parallel to the *x* axis is defined as the non-empty intersection of two adjacent uncertainty intervals $U_{i,x}$ and $U_{i+1,x}$, that is, $C_{i,i+1,x} = U_{i,x} \cap U_{i+1,x}$. Contacts between objects may occur only within such contact intervals. Similarly, a non-contact interval $NC_{i,x}$ parallel to the *x* axis is defined as $NC_{i,x} = U_{i,x} - C_{i,i+1,x} - C_{i,i-1,x}$ (Figure 4c). Contact and non-contact intervals parallel to the *y* axis are analogous.

Contact intervals within a path may be tightened using object dimensions as additional constraints. Contact intervals may also be tightened across paths. The latter step is described in the next section. As not necessarily all contact constraints may be satisfied simultaneously (an overconstrained configuration is shown in Figure 5), constraints are considered incrementally, that is, contact intervals are first adjusted within and then, if successful, across paths.

Both contact interval tightening steps take advantage of variations in contact intervals associated with a path, which are due to measurement errors. Within each path, these are compared with a sequence of intervals based on object dimensions, $W_{1,x}$, $W_{2,x}$,...$W_{n,x}$ or $W_{1,y}$, $W_{2,y}$,...$W_{n,y}$ (Figure 4c, formal definition in Appendix).

A delta interval $D_{i,i+1,x,l}$ is computed next for each $W_{i,x}$ (Figure 4c, formal definition in Appendix). The length of that interval is the distance by which $W_{i,x}$ would need to be translated to reach the lower bound $c_{i,i+1,x,h}$ of a contact interval $C_{i,i+1,x}$. Similarly, a difference interval $D_{i,i+1,x,u}$ reaches the upper bound $c_{i,i+1,x,t}$ of a contact interval $C_{i,i+1,x}$. As each $W_{i,x}$ should at least reach the lower bound $c_{i,i+1,h}$ of its contact interval $C_{i,i+1,x}$, the maximum $d_{i,i+1,x,l} = |D_{i,i+1,x,l}|$ is determined:

$$d_{min} = max(d_{1,2,x,l}, d_{2,3,x,l}, \ldots d_{n-1,n,x,l}); d_{min} < d_{n,x}$$

Similarly, the minimum $d_{i,i+1,x,u} = |D_{i,i+1,x,u}|$ is determined to reach the upper bound $c_{i,i+1,t}$:

$$d_{max} = min(d_{1,2,x,u}, d_{2,3,x,u}, \ldots d_{n-1,n,x,u}); d_{max} < d_{n,x}$$

If $d_{min} \le d_{max}$, then a configuration is overconstrained, that is, not all contacts between objects are feasible (Figure 5). In that case, contacts may be assumed only for every other pair of objects in a path, which is guaranteed to work.

If $d_{min} > d_{max}$, then all contacts are feasible, and d_{min} and d_{max} are used to compute each tightened contact interval $C'_{i,i+1,x}$:

$$C'_{i,i+1,x} = [w_{i,x,t} + d_{min}, w_{i,x,t} + d_{max}]$$

An example of this tightening step is given in Figure 4. Note that all contact intervals $C'_{i,i+1}$ have the same length $d_{max} - d_{min}$.

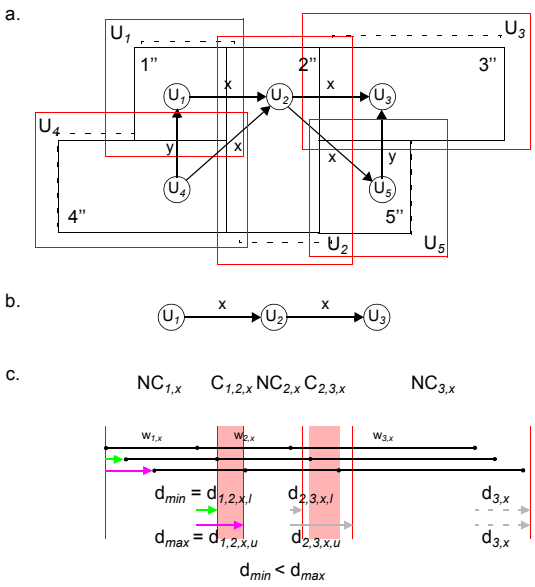

Figure 4. Tightening of contact intervals within paths. a. Configuration with uncertainty zone adjacency graph, b. Example of a path in x direction, c. Sequence of non-contact and contact intervals (tightened contact intervals are shaded).

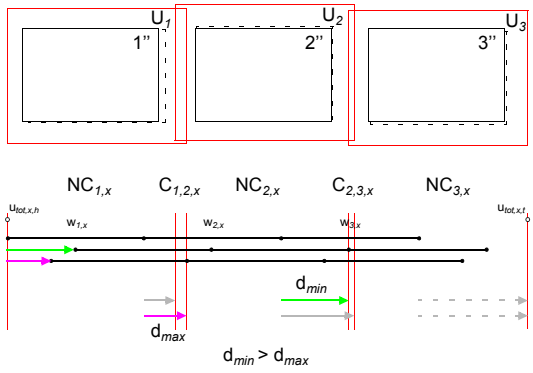

Figure 5. Overconstrained configuration.

Tightening of contact zones across paths

Tables or cabinets are often arranged according to repetitive patterns. Some of these patterns are detected when contact zones are tightened across paths, which, again, is done independently for each direction. Towards that end, all uncertainty zone interval elements from all paths in x or y direction, e.g. $U_{1,x}$, $U_{2,x}$,...$U_{n,x}$, are organized and ordered into a single list (top row of table in Figure 6b). Head and tail elements of a tightened contact interval are detected by scanning the list for a tail immediately succeeding a head element. To determine which paths share a contact interval, lists within each path (bottom rows of table in Figure 6b) are scanned. A path participates only if a head and tail element can be found to the left and to the right, respectively, of the 'partition' formed by the head and tail elements of the tightened contact interval (shown in bold in Figure 6b). Contact and non-contact intervals in each path are adjusted accordingly (Figure 6c).

Note that all paths sharing a contact interval need to be re-tightened with respect object dimensions, that is, the procedure described in the previous section is repeated, to ensure that all contact intervals have the same length.

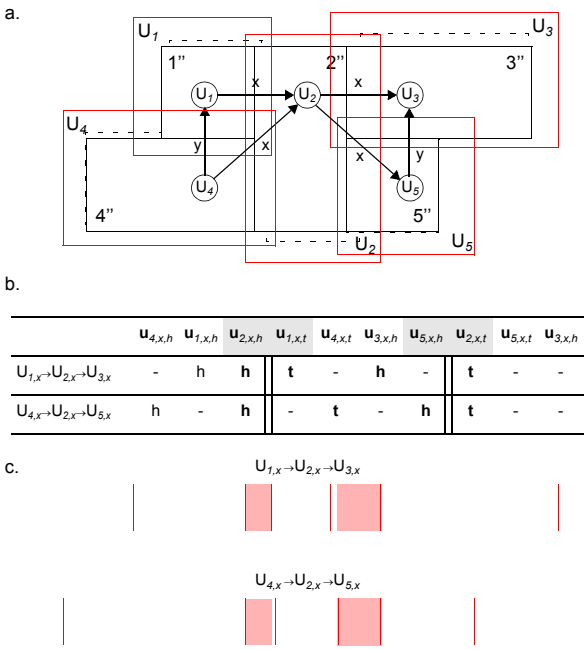

Figure 6. Tightening contact zones across paths. a. Configuration with uncertainty zone adjacency graph, b. Table with list of uncertainty interval elements, c. Sequences of non-contact and contact intervals in the x direction (tightened contact intervals are shaded).

Object position adjustment computation

Object position adjustments are computed after contact interval tightening. Adjusted x coordinates are determined as follows (Figure 1):

$$p'_{1,x,h} = c'_{1,2,x,m} - w_1$$

$$p'_{i,x,h} = c'_{i,i+1,x,m}$$

$$c'_{i,i+1,x,m} = c'_{i,i+1,x,h} + (c'_{i,i+1,x,t} - c'_{i,i+1,x,h})/2$$

Additionally, the offset parameter o'_i is computed for a space boundary face SO_i (Figure 1d):

$$o'_i = p'_{i,x,h} - u_{i,x,m}$$

$$u_{i,x,m} = u_{i,x,h} + (u_{i,x,t} - u_{i,x,h})/2$$

The computation of object position adjustments in y direction is analogous.

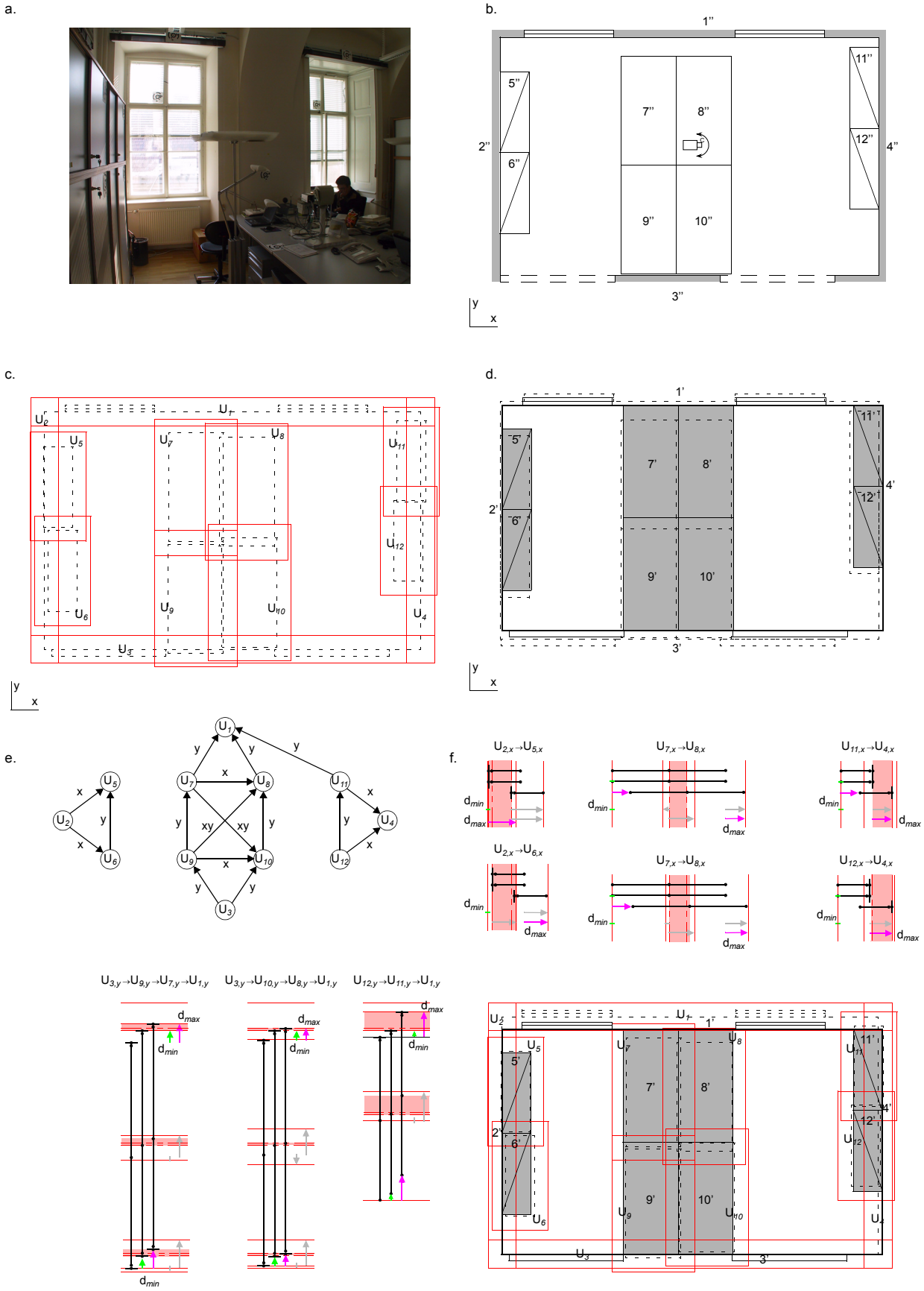

Figure 7. Test space scene. a. As-built (image), b. As-built (plan), c. As-sensed with uncertainty zones, d. Generated space model (dotted lines indicate as-built), e. Uncertainty zone adjacency graph, f. Contact-/non-contact zone and object dimension intervals (dotted lines indicate as-sensed).

EXPERIMENT

The accuracy of lighting simulations for models generated with the above object position adjustment procedure was tested in a real space scene. We used Lumina for the prediction of illuminance levels (Pal and Mahdavi 1999). Lumina utilizes a three component procedure (i.e. direct, externally reflected, and internally reflected components) to obtain the resultant illuminance distribution in indoor environments. A radiosity-based approach is adopted for computing the internally reflected component. Lumina is based on the notion of three-dimensional spaces consisting of a space enclosure and windows.

Lighting simulations were conducted for two space models: one reflecting the space as-built (Figure 7a,b), the other generated with the object position adjustment procedure applied to a boundary representation based on object location data provided by the TRIP location sensing system (Figure 7c-f). The bounding box for windows was modeled to include a sill height of 0.85m. This ensured a *z*-relation with the floor and identical elevation for both windows. Furthermore, recessed window faces were projected to their containing wall (Suter et al. 2005). Although used to constrain space scene model generation, tables and cabinets were not included in the Lumina input model for simplicity. Position uncertainty ε_p was set to 0.2m, which was sufficiently large to correctly detect all object relations (ε_p would need to be enlarged to anchor free-floating objects).

CIE overcast sky conditions were used for the sky modeling in Lumina. Besides average illuminance, the two-point daylight factor DF_{2p} was chosen as performance indicator because it has been shown to be a suitable estimator of the lighting conditions in a space (Seidl 1986). For rectangular spaces, it is defined as the average illuminance measured at two points located 1*m* from the left or right side walls and at half room depth (Figure 8).

Simulation results are summarized in Figure 8 and Table 3. Due to smaller depth, the generated model has illuminance and DF_{2p} values that are ca. 10-25% higher than the as-built model.

Table 3. Comparison of simulation results for as-built and generated space configuration.

Performance indicators	Configuration		$SB_{generated}$ / $SB_{as\text{-}built}$
	$SB_{generated}$	$SB_{as\text{-}built}$	
Average illuminance [lx]	351	311	113 %
P_1 [lx]	385	317	121 %
P_2 [lx]	351	278	126 %
Df_{2p} [%]	3.5	2.8	124 %

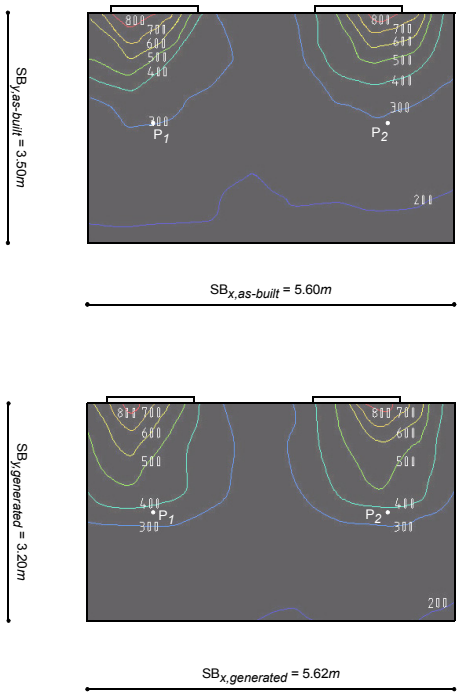

Figure 8. Lighting simulation results for as-built (top) and generated scene model (bottom).

DISCUSSION

The work presented in this paper introduces an algorithm that can be used in the reconstruction of space scenes that are valid and sufficiently detailed for lighting analysis. Uncertainty in location sensing is addressed by contact constraints applied to space boundaries and equipment objects. Initial simulation results obtained from a test space indicate that these models may be sufficiently accurate for applications such as simulation-based lighting control. To validate this approach, additional experiments need to be carried out in test spaces with systematic variations in space boundary geometry (e.g. U- and L-shaped spaces), openings, or space equipment. These factors significantly constrain simulation input models and hence affect the accuracy of simulation results. Furthermore, the effects of daylighting conditions, or combinations of daylighting and artificial lighting, need to be considered. A fully functional prototype system, including a controller, would permit further exploration and validation of simulation-based lighting control in comparison with conventional control schemes. We presently test such a prototype.

Concerning the object position adjustment procedure, extending the algorithm's domain to non-rectangular space scenes would represent a significant improvement. As most space configurations are predominantly orthogonal, the overall structure of the algorithm could remain. Additions would be required in each step to address non-orthogonal features.

APPENDIX

An object dimension interval W_i is defined as follows:

$$W_{i,x} = \left[u_{tot,x,h} + \sum_{j=1}^{i-1} w_{j,x}, u_{tot,x,h} + \sum_{j=1}^{i} w_{j,x} \right]$$

where $u_{tot,x,h}$ refers to the head element of the interval

$$U_{tot,x} = \sum_{i=1}^{n} U_{i,x}$$

Object dimension intervals for y direction are defined similarly.

A delta interval $D_{i,i+1,x,l}$ with respect to the lower bound $c_{i,i+1,x,h}$ of a contact interval $C_{i,i+1,x}$ is defined as follows:

$$D_{i,i+1,x,l} = \left(\sum_{j=1}^{i-1} NC_{j,x} \cup C_{j,j+1,x} \right) \cup NC_{i,x} - \sum_{j=1}^{i} W_{j,x}$$

Similarly, a delta interval $D_{i,i+1,x,u}$ with respect to the upper bound $c_{i,i+1,x,t}$ of a contact interval $C_{i,i+1,x}$ is defined as follows:

$$D_{i,i+1,x,u} = \left(\sum_{j=1}^{i} NC_{j,x} \cup C_{j,j+1,x} \right) - \sum_{j=1}^{i} W_{j,x}$$

ACKNOWLEDGMENT

The research presented in this paper was supported in part by a grant from the Austrian Science Fund, project number P15998-N07.

REFERENCES

Ellis, R. 1991. Geometric uncertainties in polyhedral object recognition, IEEE Transactions on Robotics and Automation 7 (3), pp. 361-371.

Icoglu, O., Brunner, K., Mahdavi, A., Suter, G. 2004. A distributed location sensing platform for dynamic building models, Proceedings Second European Symposium on Ambient Intelligence, Eindhoven, The Netherlands.

Lopez de Ipina, D., Mendonca, P., Hopper, A. 2002. Visual Sensing and Middleware Support for Sentient Computing, Personal and Ubiquitous Computing 6 (3), pp. 206–219.

Mahdavi, A. 2001. Aspects of self-aware buildings, International Journal of Design Sciences and Technology 9, pp. 35-52.

Pal, V., Mahdavi, A. 1999. A comprehensive approach to modeling and evaluating the visual environment in buildings, Proceedings of Building Simulation '99. Sixth International IBPSA Conference, Kyoto, Japan. Vol. II., pp. 579 - 586.

Requicha, A. 1983. Toward a theory of geometric tolerancing, International Journal of Robotics Research 2(4), pp. 45-60.

Seidl, M. 1986. Tageslicht in Innenräumen, Ed. J. Beckert, Gesundes Wohnen, Beton-Verlag, Düsseldorf, Germany, pp. 206-216.

Suter, G. 2004, Tag representations of simple polyhedra, Technical Report, Vienna University of Technology, Vienna, Austria.

Suter, G., Brunner, K., Mahdavi, A. 2005 (to appear). Spatial reasoning for building model reconstruction based on sensed object location information, Proceedings of CAAD Futures 2005, Vienna, Austria.