A vision-based sensing system for sentient building models

O. Icoglu & A. Mahdavi
Department of Building Physics and Building Ecology, Vienna University of Technology, Vienna, Austria

ABSTRACT: The quality and cost effectiveness of services in the building industry possess high potential for improvement. A possible approach to bringing out this potential is to conceive buildings as sentient entities that continuously adapt to changes in environment and occupancy. To achieve real-time building operation support and to avoid bottleneck situations resulting from manual model input and updating activities, the underlying model must possess the capability to autonomously update itself. This requires a versatile sensing mechanism that provides real-time facility state information. The research described in this paper aims to demonstrate the potential of vision-based sensing solutions to support the operations of sentient buildings. Toward this end, a system prototype has been implemented that focuses on building systems control (lighting). The resulting arrangement of required hardware and software components (tied together via internet) provides a flexible and self-adapting structure, which is highly suited to the requirements of control applications for sentient buildings.

1 INTRODUCTION

The research presented in this paper is part of a project to demonstrate the feasibility, scalability, and potential of sentient building technologies. A sentient building is one that possesses a multi-faceted internal representation of its own context, structure, components, systems, and processes. It can use this representation, amongst other things, toward the full or partial self-regulatory determination of its indoor-environmental status (Mahdavi 2004). The realization of a sentient building is, we argue, within reach. However, to achieve this goal, already acquired scientific foundations (theories, methods, and tools) must be translated into a technically mature and industrially promising level. Specifically, the representational core of sentient buildings must integrate static building component class hierarchies (product models) with process-oriented systems controller hierarchies (process models). Moreover, the complexity of buildings (multitude of components, environmental systems, occupancy patterns, contextual influences) implies that the associated representations must be continuously updated, if they are to be applied effectively in the course of building operation and maintenance activities. To avoid bottleneck situations resulting from manual input and updating activities, the underlying product-process model must possess the capability to autonomously update itself. This requires a versatile sensing mechanism that provides real-time facility state information. The research described in this paper aims to demonstrate the effectiveness of vision-based sensing solutions to support the operations of sentient buildings.

2 VISION-BASED SENSING

To deliver a proof of concept for the feasibility, the necessary sensing mechanisms are implemented focusing on the lighting control system for a sentient building (Mahdavi 2001a, 2001b). In such a control system, objects in the space must be identified, their locations must be sensed, changes in reflectance of the objects and surfaces must be monitored, and occupancy information must be obtained. Furthermore, the prospective solution must comply with the building-specific requirements where low-cost, low-maintenance, and scalability are crucial. In our efforts for realizing such a solution, a Vision-based Object and Occupancy Location Assessment System (VIOLAS) is developed that extracts context information from the built environment using vision-based methods. In a technology review performed prior to the implementation of the system (Icoglu et al. 2004), vision-based approaches were found to be preferable in terms of being software supported and system customizable.

The "sensing core" of VIOLAS is comprised of five major blocks, whereby each block performs a distinct function for context data extraction (Fig. 1).

2.1 *Hardware Interface*

The Hardware Interface block allows data acquisition from multiple sensors. Recent developments in embedded computing led to the integration of sensors with processors. This reduced the costs, as dedicated computers were not necessary to enable data compression and communication. Thus, data could be efficiently conveyed over large-scale networks. In vision sensing domain, such developments gave rise to network cameras (netcams). Netcams have embedded computing power that enables image compression and data communication over Internet via standard protocols. They also enable the control of third-party devices like pan-tilt units (P/T) through the same communication channel.

Netcams can be easily adapted to built environments by making use of existing network installations without requiring additional infrastructure. Towards this end, we use network cameras together with pan-tilt units that effectively increase the monitoring range. However, netcams generate relatively low-quality data due to data compression. This results in low resolution and blurred images without sharp details. To enhance the performance, the system applies adaptive sharpening and zooming to the initial netcam images (see the following section).

2.2 *Object Identification and Location Sensing*

The location sensing module is adapted from TRIP system (Lopez de Ipina et al. 2002) that basically uses "pose extraction from circle" algorithm, estimating the pose of a circle in space from a single

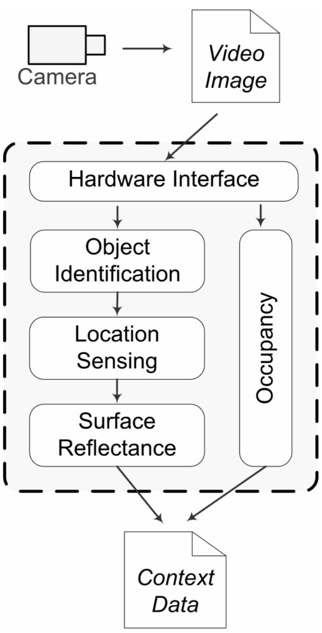


Figure 1. "Sensing core" in VIOLAS

intensity image. The working of the algorithm is illustrated in Figure 2. The target plane generates an ellipse on the image plane of the camera. From the known parameters of the ellipse, it can be back-projected to the original circle, enabling the extraction of the orientation and the position of the target plane with respect to the camera origin (Trucco & Verri 1998).

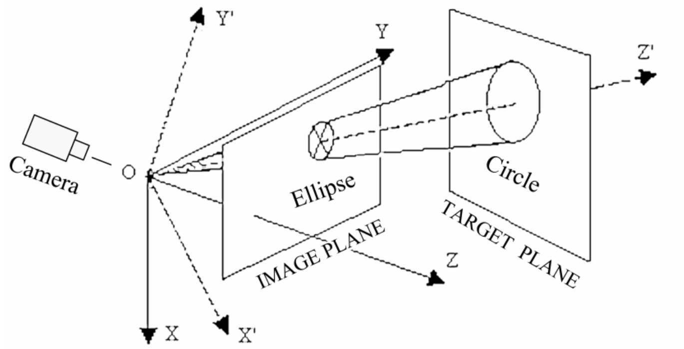


Figure 2. "Pose from circle" algorithm. (*X*,*Y*,*Z*) denotes the coordinate system of the image plane whereas (*X'*,*Y'*,*Z'*) denotes the coordinate system of the target plane. The outcome of the algorithm is the parameters of the transformation between two coordinate systems

For the realization of the algorithm, barcode like tags with circular marks are used. These tags can be printed using regular black-and-white printers. This is one of the main benefits of the system, as the tags are low-cost and low-maintenance, and require no power input. Currently, we use 12 by 12 cm tags. The identification is accomplished by the codes printed around the circular mark (reference circle). Unlike the TRIP system that uses ternary coding, VIOLAS uses binary coding, where the tags are divided into 16 equal sectors (Fig. 3) resembling pie slices. The presence or absence of the black mark on the sector denotes the 1 or 0 coding respectively. The pattern of "0111" code sequence defines the start bits, and is never repeated elsewhere in the rest of the data string.

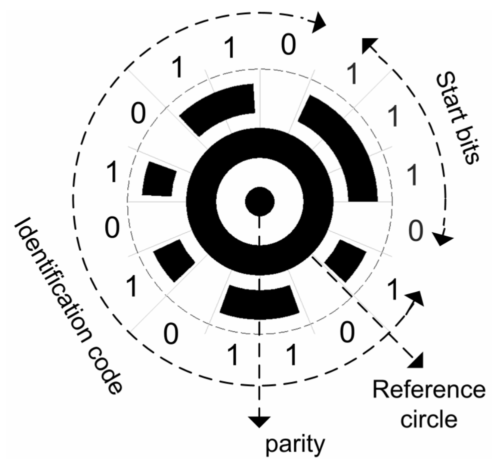


Figure 3. Tag structure is illustrated with a sample tag coded with 0111-011010101101 data string (even-parity = 1). Identification number corresponds to 1709 in decimals

The identification number is encoded in the remaining 12 sectors. Finally, an even-parity bit is added at the center of the reference circle for the verification of the decoded data string in the end of the identification phase. This coding structure enables the definition of 2031 distinct tagged objects.

The TRIP system divides the location sensing procedure into two phases. First is the "target recognition" phase, where the tags are detected, parameters of the reference ellipse (projection of the reference circle on the camera image) are extracted, and the identification numbers are decoded. Second is the "pose extraction", where the location of the tags are computed from the outputs of the first phase (Lopez de Ipina et al. 2002).

VIOLAS enhances the original TRIP method by integrating two additional algorithms (Fig. 4). The original system was implemented on images captured by digital cameras that provide uncompressed, high quality data. However, working on raw images is not applicable in distributed environments such as buildings. As mentioned in section 2.1, netcams are used as sensor devices for this reason, where the images can be transported in wide areas through HTTP. Like other digital video devices, netcams are designed to convey images as fast as possible and therefore apply compression to images prior to transmission. This generates smoothed input images and causes tag image regions to lose sharpness. To compensate this, VIOLAS applies an "adaptive sharpening algorithm" (Battiato et al. 2003) on the input image prior to "target recognition" (Fig. 4).

In addition to camera artifacts, an increase in the distance of tags to camera reduces the pixel resolution of the tag images and makes the identification codes harder to decipher, even though the tags are detected and reference ellipses are extracted properly. To solve the problem, "edge-adaptive zooming" (Battiato et al. 2000) is applied locally to spurious tags from which the code could not be deciphered or validated. Edge-adaptive zooming, as opposed to its counterparts such as bilinear and cubic interpolation, enhances the discontinuities and sharp luminance variations in the tag images. This procedure is repeated until the "target recognition" succeeds or the zoomed image region loses its details (Fig. 4). The latter case indicates a false alarm or an unidentified tag.

2.3 *Surface Reflectance Estimation*

In addition to location information, surface attributes of the corresponding objects are also important for the model generation. Surface Reflectance Estimation determines the reflectance of the objects that are identified and located in the previous steps.

Key information in the reflectance estimation of diffuse surfaces is the illumination data, which is obtained in the system based on the known tag reflectances. Since the tags are attached directly to the object surfaces, they possess the same illuminance. By comparing the brightness of the object with the brightness of the tag, the reflectance of the object surface can be estimated (Horn 1986).

2.4 *Occupancy Sensing*

Occupancy information is acquired from the temporal image sequences with the calculation of optical flow, i.e. a motion detection method that tracks the apparent motion of brightness patterns (Ballard & Brown 1982).

3 VIOLAS IMPLEMENTATION

The aim of the system is to collect visual data from sensors and extract context information from the environment. The results must be conveyed to the Lighting Control System. Figure 5 illustrates the process to fulfill this goal. Towards the realization of the system, context data generated by the "sensing core" must undergo some additional processes to provide unified, consistent, real-world information.

3.1 *Coordinate Transformation*

The outcome of the "sensing core" regarding the location data is the position and orientation information with respect to the coordinates of the camera from which the processed image is acquired. Coordinate Transformation converts the position and orientation data with respect to camera coordinates to the position and orientation data with respect to the real-world coordinates by using 3-D transformations. The transformations must also take into account the presence of a P/T unit and the camera's position on it (Fig. 6).

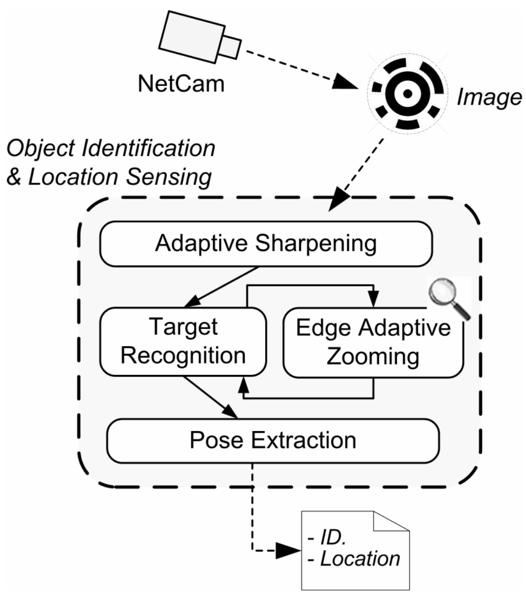


Figure 4. Algorithm flow of Object Identification and Location Sensing in the "sensing core"

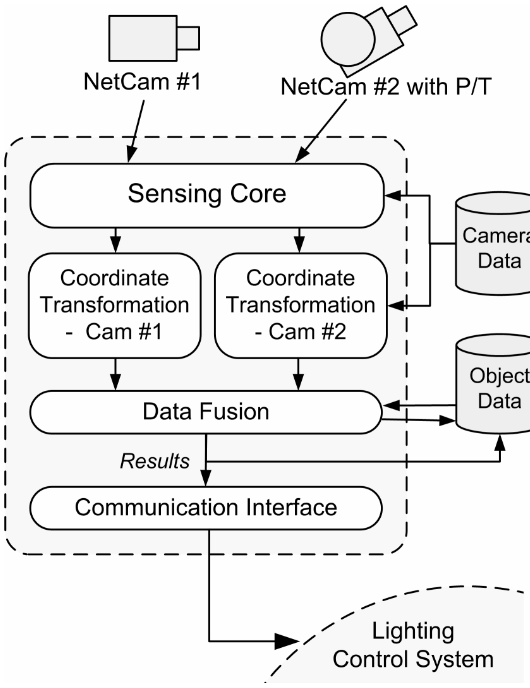


Figure 5. Process flow designed for VIOLAS implementation

3-D transformations map the coordinates of one point in 3-D space to potentially different coordinates defined in a distinct system frame in 3-D space. The camera parameters are stored in a camera database that uniquely identifies the transformations between each of the camera reference frames and the world reference frame (the same database is also used by “sensing core” for data acquisition and camera calibration). For describing the relative positions of the origins of the two reference frames, a 3-D translation vector, T, is used. An orthogonal 3×3 rotation matrix, R, aligns the corresponding axes of the two frames. The orthogonality relations reduce the number of degrees of freedom of R to three angles. In a common notation, the relation between the coordinates of a point P in world and camera frame, P_w and P_c respectively, is

$$P_w = R \cdot (P_c) + T \tag{1}$$

where rotation is defined from room to camera and translation is defined in room coordinates (Fig. 6a).

If the camera is attached on a pan-tilt unit, a sequential transformation must be applied, first from room to P/T, where P/T device is at its original position *(pan = 0°, tilt = 0°)*, and then from P/T to camera, where pan and tilt angles are involved (Fig. 6b). The rotations of these two transformations are given respectively as;

$$R_1 = room \rightarrow P/T, \quad R_2 = P/T \rightarrow camera \tag{2}$$

If we assume that the camera is mounted on top of the pan-tilt unit, there is no additional translation other than the translation of the Pan-Tilt Unit, T. Therefore the equation becomes:

$$P_w = R_1 \cdot R_2 \cdot (P_c) + T \tag{3}$$

However, practically it is impossible to place the camera right at the top of the P/T unit: there is always a shift from the origin of the P/T coordinate system (Fig. 6c). This shift, S, is defined in the P/T unit's coordinate system, and must be transformed into the room specific values, S', in order to be added in the final equation:

$$P_w = R_1 \cdot R_2 \cdot (P_c + S) + T, \quad \text{or} \tag{4}$$

$$P_w = R_1 \cdot R_2 \cdot (P_c) + S' + T \tag{5}$$

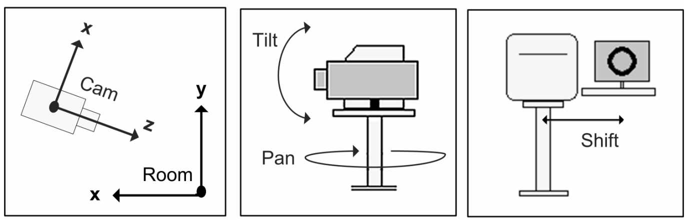
The total rotation ($R_{total} = R_1 \cdot R_2$) and the total translation ($T_{total} = S' + T$) give the final relation between the locations of the objects in world and camera frame.

3.2 *Data Fusion*

The context data acquired from all cameras are combined in the Data Fusion phase. In addition to its fusion task, Data Fusion also constructs the graphical representations of the combined context data. By using these representations, a user interface displays images of the system results for a convenient user interaction. There are two phases of the Data Fusion, namely tag level fusion and object level fusion.

3.2.1 *Tag Level Fusion*

The same tag can be detected with more than one camera, or one camera assigned to multiple instances of the “sensing core” (to be discussed in section 4). This will eventually generate repeated tag records coming from multiple cameras (or “sensing cores”) in the system. Data Fusion combines these records by taking the identification time and uncertainty data into account. Most up-to-date and certain information is selected as the final, unique tag information. Time and uncertainty are assigned by “sensing core” right after the object identification. Uncertainty is generated with respect to the pose data (particularly, the parameters of the reference tag ellipse). This provides information about the accuracy of the location sensing.



(a) (b) (c)

Figure 6. a) Camera parameters are important to define the transformations. b) If P/T unit is involved, pan and tilt angles must also be considered. Most P/T devices are manufactured with built-in potentiometers that provide the angle values. c) Position of the camera on the P/T unit is not negligible

As distance and incidence angle increase (as reference ellipse gets smaller and diverges from a circle), the deviations of the location information from the real values increase as well.

3.2.2 *Object Level Fusion*

The second phase of the Data Fusion is implemented in the object-level. In this phase, the (fused) tag information is transformed to object information.

When a tag-code is created (by a user interface program), the related object information (name, description, dimensions, etc.) is also entered. Additionally, the location of the tag on the object is defined. Therefore, the system is aware of the object information with the identified tag-code and can generate the graphical representation of the object from the object dimensions and tag location.

The system also enables the attachment of multiple tags on a larger object to reduce the occlusion possibility and to increase the line-of-sight between tags and cameras. This requires the second level fusion in order to prevent redundant object records when both tags are identified. As with tag-level fusion, the most up-to-date and certain information is selected as the final, unique object information.

3.3 *Communication Interface*

Output of the Data Fusion, i.e. the final, consistent context data, is transformed into XML-like data packets for convenient data communication, and transferred to the Lighting Control System through the Communication Interface. The communication is established with a TCP/IP socket server implemented in the interface that enables the connection of not only the main system but also any other third party applications that can prospectively download and process the data packets. Communication Interface conveys the context data to the clients in the course of the first connection and afterwards, whenever a change occurs in the environment.

4 SOFTWARE PLATFORM

In order to cover the distinct activities of the location and context sensing in a common software platform and to simultaneously fulfill the requirements of a sensing technology adaptable to the building environment, the VIOLAS platform is designed in a distributed structure, with the components tied together via internet. Communication and data sharing is facilitated by the Distributed Component Object Model (DCOM) protocol that enables software components to communicate directly over a network (DCOM 2004). This design enables efficient resource utilization and permits load balancing and information sharing derived from the parallel operation and remote data access. Distributed structure provides scalability and incremental growth, in addition to enhanced performance resulting from the parallel operation.

Based on the above structure, VIOLAS resides on a distributed platform, divided into server and client tiers (Fig. 7). The server tier is comprised of an application server that achieves the resource management and data integration, a user interface server that performs web-based user interaction, and a database server that handles data management. The application server is the heart of the system. It controls the distributed components including the sensors (resources) and clients (consumers). As mentioned in section 2.1, netcams are used as visual sensors that fit in this structure by conveying video images like as distributed network devices. Clients are the Image Processing Units (IPUs). They wrap the "sensing core" described in section 2 (including Coordinate Transformation described in section 3.1) and interpret input images. IPUs are implemented on different computers scattered across a facility. Results obtained from multiple IPUs are combined in the application server and subsequently transferred to the Lighting Control System. This combination process is performed by Data Fusion and data transfer is accomplished by Communication Interface, both implemented in the application server. Result displays are also generated in the Data Fusion phase as mentioned in section 3.2, and relayed to the users through the user interface server. This server allows for web-based access from every computer on the Internet.

An additional function of the application server is to control the status of the distributed components and dynamically assign active netcams to active IPUs in such a manner that the workload is constantly balanced within the system. This arrangement provides a kind of "self-organizing" capability

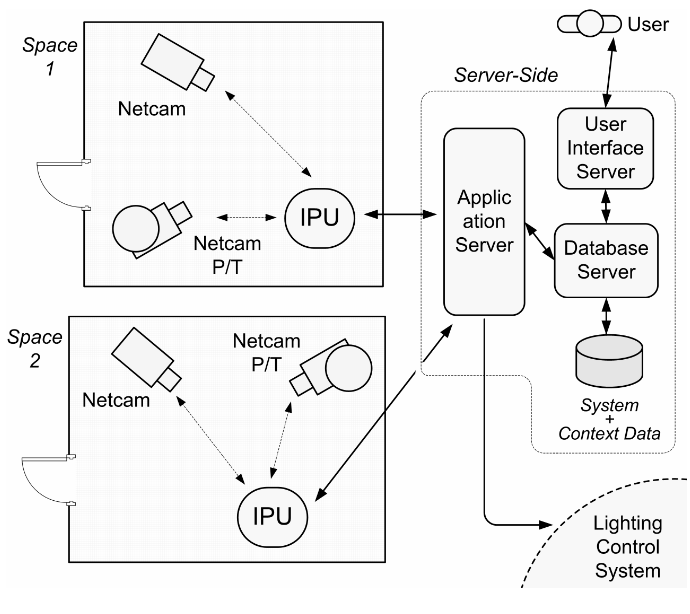


Figure 7. Distributed software platform of VIOLAS. Distinct sensing activities are collaborated under a common self-organized model

by minimizing operator overhead. It also enables the utilization of *a)* multiple cameras by a single IPU in a resource-rich configuration, or *b)* a single camera by multiple IPUs in a customer-rich configuration. A resource-rich configuration can increase the coverage area of the system, whereas a customer-rich configuration can augment the system speed.

5 A DEMONSTRATIVE TEST SPACE

To evaluate the performance of VIOLAS, a demonstrative test was performed. Thereby, the accuracy of object identification and location sensing were observed. Towards this end, a typical office environment (test-bed) was used that involves 26 tagged objects relevant for the Lighting Control System. A 2-D sketch of the test-bed is illustrated in Figure 8. The ground-truth data (actual location information) of the objects were measured for the test (Table 1).

VIOLAS was then instantiated with a single net-cam - P/T pair and the results were recorded. The system achieved a 100% identification performance, extracting all tag-codes and recognizing all objects (Table 2).

To evaluate the test results, “position error” is defined as the distance between the ground-truth position and the sensed position of the tag. “Orientation error” is defined as the angle between a plane's true surface normal and the sensed surface normal.

Table 3 includes, for our test, the resulting position and orientation errors together with the respective camera-tag distances. Table 3 includes also the position errors in relative terms (in percentage of camera-tag distance).

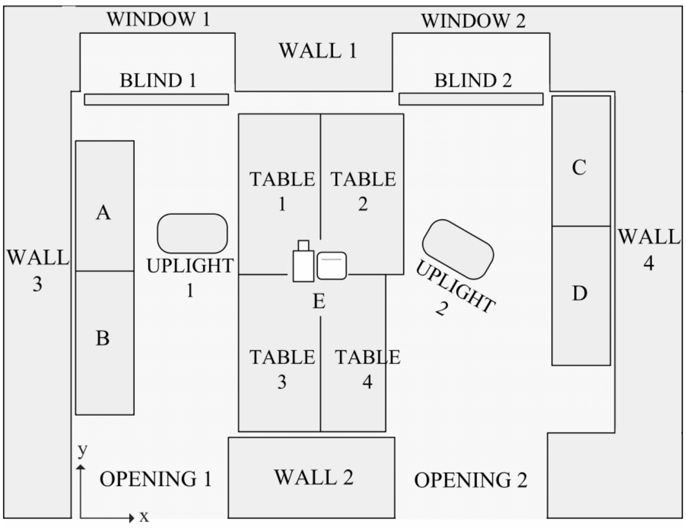


Figure 8. 2-D sketch of the test-bed. ‘A’ refers to Cabinet-3 and Upper-Cabinet-2, ‘B’ refers to Cabinet-4 and Upper-Cabinet-3, ‘C’ refers to Cabinet-1 and Upper-Cabinet-1, ‘D’ refers to Cabinet-2, ‘E’ refers to Camera and P/T unit

Table 1. Ground-truth data of the objects in the test-bed. Orientation is denoted with the normal vectors of the tag planes. This vector represents the orientation of the tag expressed in the room coordinate system

Object Name	Position (m)			Orientation		
	Tx	Ty	Tz	Nx	Ny	Nz
BLIND 1	1.04	4.25	3.38	0	-1	0
BLIND 2	4.03	4.25	3.38	0	-1	0
CABINET 1	5.16	3.1	1.42	-1	0	0
CABINET 2	5.16	1.9	1.42	-1	0	0
CABINET 3	0.44	3.69	1.42	1	0	0
CABINET 4	0.44	2.5	1.42	1	0	0
CEILING	4.1	1.25	4.05	0	0	-1
FLOOR	0.12	0.1	0	0	0	1
TABLE 1	1.95	2.55	0.73	0	0	1
TABLE 2	1.95	0.94	0.73	0	0	1
TABLE 3	3.35	3.94	0.73	0	0	1
TABLE 4	2.75	0.94	0.73	0	0	1
UPLIGHT 1	1.4	2.89	1.75	0	0	-1
UPLIGHT 2	4.24	2.44	1.75	0	0	-1
UPPER CABINET 1	5.16	3.1	2.16	-1	0	0
UPPER CABINET 2	0.44	3.69	2.16	1	0	0
UPPER CABINET 3	0.44	2.5	2.16	1	0	0
WALL 1	2.55	4.33	1.28	0	-1	0
WALL 2	2.55	0.8	1.28	0	1	0
WALL 3	0	1.43	3	1	0	0
WALL 4	5.6	2.8	1.89	-1	0	0
OPENING 1	0.85	0.8	3.5	0	1	0
OPENING 2	4.1	0.8	3.5	0	1	0
WINDOW 1	0.8	5.11	2.6	0	-1	0
WINDOW 2	3.95	5.11	2.6	0	-1	0

Table 2. Sensed location values of the objects in the test-bed recorded by VIOLAS

Object Name	Position (m)			Orientation		
	Tx	Ty	Tz	Nx	Ny	Nz
BLIND 1	0.94	4.13	3.32	0.03	-1	-0.01
BLIND 2	4.01	4.28	3.26	0.07	-0.99	0.08
CABINET 1	5.08	3.22	1.31	-1	0.05	0.05
CABINET 2	5.2	2.04	1.31	-0.99	-0.12	-0.02
CABINET 3	0.5	3.6	1.38	1	0.01	0.02
CABINET 4	0.53	2.43	1.36	1	0.02	0.07
CEILING	4.38	1.25	3.79	-0.21	0.12	-0.97
FLOOR	0.26	0.09	-0.07	-0.01	-0.02	1
TABLE 1	1.97	2.54	0.75	0.01	0.05	1
TABLE 2	2.04	1.02	0.72	-0.01	-0.02	1
TABLE 3	3.27	3.9	0.73	0.03	0	1
TABLE 4	2.82	1.07	0.74	-0.01	-0.02	1
UPLIGHT 1	1.51	2.82	1.67	0.04	0.05	-1
UPLIGHT 2	4.19	2.5	1.64	-0.05	-0.03	-1
UPPER CABINET 1	5.17	3.22	1.77	-0.97	0	0.25
UPPER CABINET 2	0.47	3.59	2.07	1	-0.01	0.03
UPPER CABINET 3	0.49	2.41	2.06	1	0.02	0.03
WALL 1	2.49	4.23	1.24	0.06	-1	0.05
WALL 2	2.65	0.91	1.2	0	1	0.07
WALL 3	0.04	1.22	2.91	1	0.1	0.02
WALL 4	5.6	2.97	1.75	-0.99	-0.02	0.11
OPENING 1	1.02	0.65	3.34	-0.05	1	0.06
OPENING 2	4.33	0.9	3.22	-0.03	1	0
WINDOW 1	0.82	4.96	2.47	0.08	-1	0.03
WINDOW 2	3.9	5.13	2.47	0.02	-1	0.03

As Table 3 shows, position errors increase with camera distance. Table 4 shows the average and maximum position and orientation errors for different camera-tag distance bins. The system possesses an average position error 0.18 m and orientation error of 4.2° on aggregate. The position error percentage has a mean value of 7.3% (Table 4).

The graphical representation of the test bed, as generated by the Data Fusion and displayed by the user interface server, is illustrated in Figure 9.

Table 3. Position and orientation errors of the objects sorted with respect to their camera distances

Object Name	Dist. to Cam (m)	Position Error (m)	Position Error (%)*	Orientation Error (°)
TABLE 1	0.70	0.03	4.3	2.9
UPLIGHT 1	1.21	0.15	12.6	3.7
TABLE 3	1.65	0.09	5.4	1.7
TABLE 4	1.69	0.15	8.8	1.3
WALL 1	1.75	0.12	7.0	4.5
TABLE 2	1.76	0.12	6.9	1.3
UPLIGHT 2	1.78	0.13	7.6	3.3
WALL 2	1.82	0.17	9.3	4
CABINET 4	2.08	0.13	6.2	4.2
UPPER CABINET 3	2.26	0.14	6.4	2.1
CABINET 3	2.32	0.12	5.0	1.3
UPPER CABINET 2	2.49	0.14	5.5	1.8
CABINET 2	2.50	0.18	7.3	7
CABINET 1	2.68	0.18	6.8	4
UPPER CABINET 1	2.83	0.41	14.4	14.5
BLIND 1	3.01	0.17	5.6	1.8
BLIND 2	3.05	0.13	4.1	6.1
WALL 4	3.18	0.22	6.9	6.4
WINDOW 2	3.23	0.14	4.4	2.1
WALL 3	3.28	0.23	7.1	5.8
WINDOW 1	3.3	0.20	6.0	4.9
OPENING 2	3.34	0.38	11.3	1.7
OPENING 1	3.35	0.28	8.3	4.5
CEILING	3.49	0.38	10.9	14
FLOOR	3.69	0.16	4.3	1.3

* Position error in percentage of camera-tag distance.

Table 4. The average and maximum position and orientation errors for different camera-tag distance bins

Error values	Camera-tag distances (m) 0..1	1..2	2..3	3..4	*All Dist.*
Position Error (m)					
AVERAGE	0.03	0.13	0.19	0.23	*0.18*
MAXIMUM	0.03	0.17	0.41	0.38	*0.41*
Position Error (%)					
AVERAGE	4.3	8.2	7.4	6.9	*7.3*
MAXIMUM	4.3	12.6	14.4	11.3	*14.4*
Orientation Error (°)					
AVERAGE	2.9	2.8	5	4.9	*4.2*
MAXIMUM	2.9	4.5	14.5	14	*14.5*

The test was performed with a 800×600 resolution IQeye™ camera possessing 10 mm (36° FOV) lens, f1.6 aperture and 6×6 μm effective pixel size. The P/T unit used in the test is a Bewator Mustang P25 that possesses 0.2° backlash. However, the total maximum error of the device is measured as 0.8° with the addition of quantization error of the digital-analog converter. 0.8° rotation error generates roughly 4.5 cm deviation in 3 meters.

6 CONCLUSION

The object identification and location sensing subsystems of VIOLAS are implanted within the previously mentioned software platform and were fully tested in an office environment as demonstrated in section 5. However, the work on the implementation and integration of reflectance and occupancy sensing subsystems is still in progress. Currently, VIOLAS with a single netcam and pan-tilt unit possesses an effective camera-tag distance range of 4 meters and an effective scanning area range of roughly 50 m^2. It achieves a 100% object identification performance under constant lighting conditions. The identification performance may drop down up to 85% under fluctuating lighting conditions that can reduce the contrast of the images acquired from netcams due to the intrinsic control mechanism of the cameras. The obtained results suggest that vision based sensing, when enhanced computationally and integrated with appropriate hardware, is a promising technology for spatial domains such as facilities and buildings.

Our future studies will focus on increasing the identification performance. Toward this end, software-based methods are being developed to selectively adjust image contrasts to compensate for changing light conditions. Moreover, prior information (and constraints) regarding the nature of the space model as well as heuristic information (and corresponding geometric reasoning) about the attributes of objects in the environment will be used to improve the location sensing performance of VIOLAS.

VIOLAS wraps the assorted sensing solutions under a common self-updating platform representing a scalable and configurable structure. We believe this provides a flexible and adaptive system that is highly suited to the requirements of indoor-environmental control applications in the built environment. The self-updating building model, as generated by VIOLAS, can provide, thus, the core of the prototypical implementation of the simulation-based control strategies in sentient buildings.

ACKNOWLEDGEMENT

The research presented in this paper is supported by a grant from FWF (Austrian Science Foundation), project number P15998-N07. The research team includes, in addition to the authors, G. Suter, K. Brunner, B. Spasojevic, and J. Lechleitner. The authors also wish to thank G. Suter for the helpful contributions.

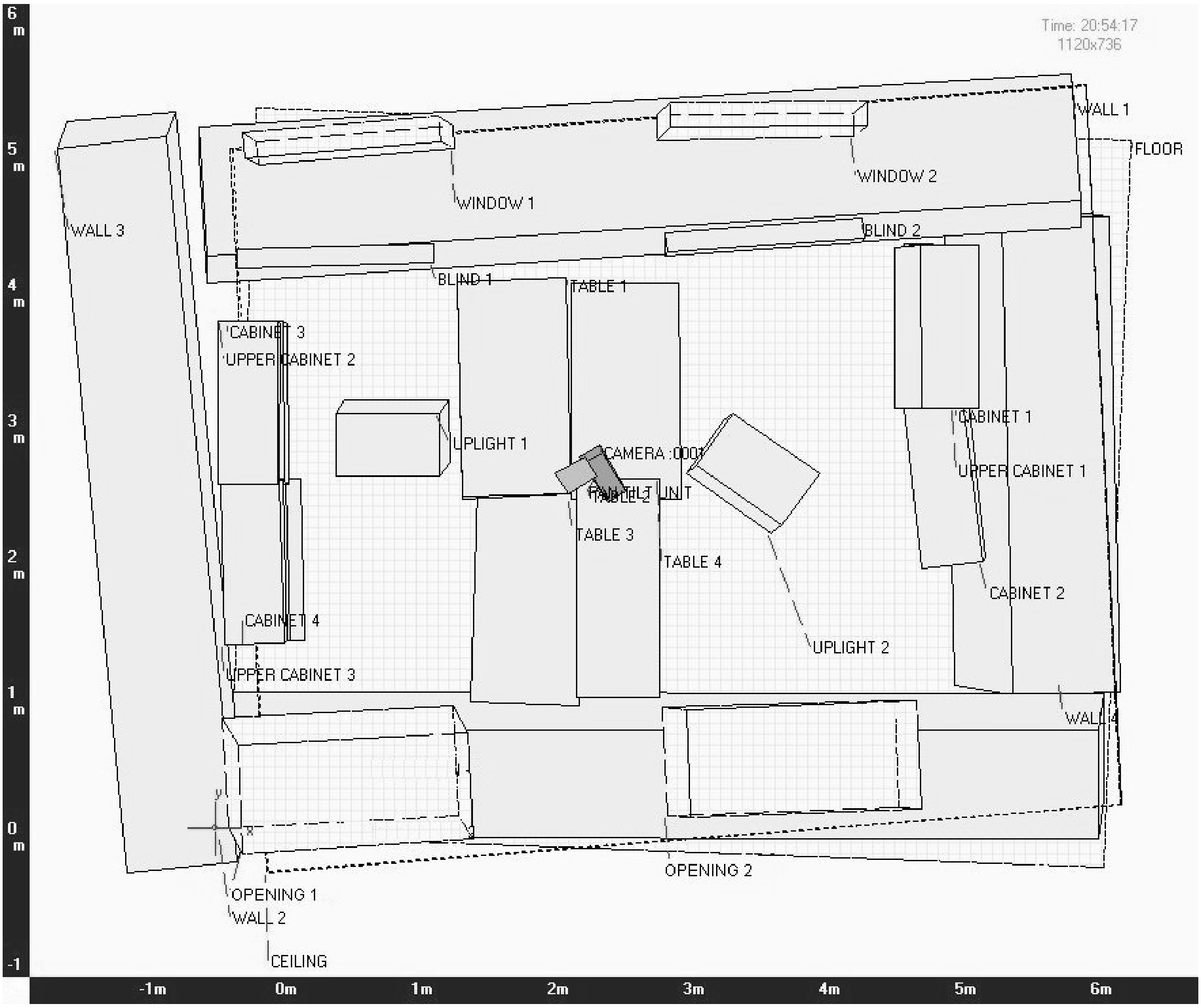


Figure 9. Graphical representation of the testbed generated by VIOLAS, after the execution of the test. The slides are caused by orientation errors that have an average value of ~5°

REFERENCES

Ballard, D.H., Brown, C.M. 1982. *Computer Vision.* New Jersey: Prentice Hall Inc.

Battiato, S., Castorina, A., Guarnera, M. & Vivirito, P. 2003. An adaptive global enhancement pipeline for low cost imaging sensors. *IEEE International conference on consumer electronics, Los Angeles, June 2003*. 398-399.

Battiato, S., Gallo, G. & Stanco, F. 2000. A new edge-adaptive algorithm for zooming of digital images. *IASTED Signal Processing and Communications, Marbella, September 2000*. 144-149.

DCOM. 2004. Microsoft COM technologies - Information and resources for the Component Object Model-based technologies. *http://www.microsoft.com/com/*

Horn, B.K.P. 1986. *Robot Vision*. Massachusetts: The MIT Electrical Engineering and Computer Science Series.

Icoglu, O., Brunner, K., Mahdavi, A. & Suter, G. 2004. A distributed location sensing platform for dynamic building models. In P. Markopoulos (ed.), *Lecture Notes in Computer Science 3295: 124-135, European Symposium on Ambient Intelligence, Eindhoven, November 2004*. New York: Springer.

Lopez de Ipina, D., Mendonca, P.S. & Hopper, A. 2002. Visual sensing and middleware support for sentient computing. *Personal and Ubiquitous Computing*. 6(3): 206-219.

Mahdavi, A. 2001a. Aspects of self-aware buildings. *International Journal of Design Sciences and Technology. Paris*. 9(1): 35-52.

Mahdavi, A. 2001b, Simulation-based control of building systems operation. *Building and Environment*. 36(6): 789-796.

Mahdavi, A. 2004. Self-organizing models for sentient buildings. In A.M. Malkawi & G. Augenbroe (eds), *Advanced Building Simulation*: 159-188. London: Spon Press.

Trucco, E., Verri, A. 1998. *Introductory Techniques for 3-D Computer Vision.* New Jersey: Prentice Hall Inc.