Vision-based 3D Spatio-Temporal Context Generation for Mobile Robot Applications

Jongleul Chung, Hyung Wook Kim, Sanghoon Lee, Il Hong Suh*, Byung Uk Choi* and Myung Kwan Park **

School of Electrical and Computer Engineering, Hanyang University, Seoul, Korea, bellaw@mlab.hanyang.ac.kr
* College of Information and Communications, Hanyang University, Seoul, Korea, {ihsuh, buchoi}@hanyang.ac.kr
** School of IT & Electronic Engineering, Dongyang University, Kyungbook, Korea, mkpark@dyu.ac.kr

ABSTRACT

A vision-based context generation system is proposed, where a point of the object associated with locations of SIFT features of the object model is utilized to describe an object as a representative point. And, SIFT features are also used to identify objects. Primitive spatial data and contexts are obtained based on our point-based description of objects as well as the use of some axiomatic rules. Finally, ST contexts are generated by classifying these primitive-ST contexts. Our proposed method is experimentally shown to be valid.

Keywords: Context, Spatio-temporal Context, Mobile Robot Applications

1. INTRODUCTION

Contextual information is essential to implement robot intelligence with which robots can correctly recognize environments and plan their behaviours to complete missions while adapting to their environments[1-3].

Contextual information must be often found by relatively inexpensive vision sensors. We note that vision-based context understanding requires not only single-object recognition, but also scene interpretation for scenes of many objects.

In this paper, a vision-based ST (Spatio-Temporal) context generation system is proposed. ST Context generation process is shown in Fig.1. SIFT features are used in objects recognition. After recognizing objects, primitive spatial data including location of objects, distances between objects, movement of objects are created. When the primitive spatial data are generated, an approximated center of an object is selected as a representative point for fast and efficient data processing.

The compensated primitive spatial data are then clustered to form primitive-ST contexts. Finally, we can get most of ST contexts by classifying these primitive-ST contexts according to the pre-defined categories. And also we can get some ST contexts by modifying and/or inferencing primitive-ST contexts and known ST contexts.

The organization of this paper is given as follows: Section 2 shows how to create the primitive spatial data from visual data. In Section 3, primitive-ST contexts are generated by clustering primitive spatial data, and ST contexts are also generated by classifying primitive-ST contexts. The proposed approach is evaluated by experiment in Section 4. The conclusion is given in Section 5.

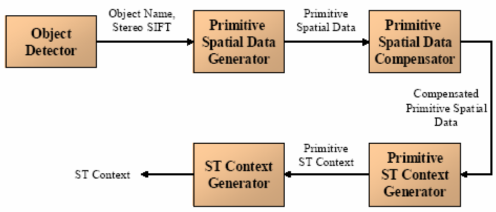


Fig. 1: *Spatio-Temporal(ST) Context Generation Process*

2. VISION-BASED PRIMITIVE SPATIAL DATA GENETATION

2.1 Generation of Primitive Spatial Data

In order to recognize objects, a model-based approach is used. We will define a 3D OFM (Object Feature Model) as the reference model. In our 3D OFM, images of object, their corresponding 3D SIFT keypoints, and representative point are stored. By simple matching of 3D keypoints between OFM and input image, we can identify objects and obtain positions of objects with respect to robot. After identifying objects, primitive spatial data are generated. Primitive spatial data describe geometric relations of recognized objects and are composed of location, distance, and movement data. Primitive spatial data are created based on the representative point of each object.

Location Data Location data implies the location of an object at each time step. Location data for an object ob is defined as R(ob) = (t, obj, **R**(ob)), where t is a time when the location data is created, obj is the name of object ob, and **R**(ob) is a position vector of object ob with respect to the robot coordinate.

Distance Data Distance data represents the distance between two objects at each time step. Distance data for objects ob1 and ob2 is defined as D(ob1, ob2) = (t, obj1, obj2, **D**(ob1, ob2)), where t is a time when the distance data is created, obj1 and obj2 represent names of object ob1 and ob2, respectively. And, **D**(ob1, ob2) is a position vector from object ob1 to object ob2 with respect to the robot coordinate.

Movement Data Movement data shows whether an object ob is moving or not at each time step. Movement data for an object ob is defined as M(ob) = (t, obj, **M**(ob)), where t is a time when the movement data is created, and obj represents the name of object ob. **M**(ob) in Fig.2 is the vector of movement which can be simply obtained by the vector difference R_t(ob) and $R_{(t-1)}$(ob).
\

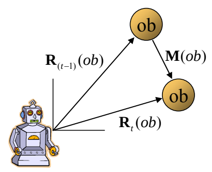


Fig. 2: *Movement Data for Object ob*

2.2 Rule-based Compensation of Primitive Spatial Data

In the object recognition, there may be some position errors caused by the mismatches of 3D SIFT keypoints. Owing to such uncertainties, primitive spatial data as shown in Fig.3(a) can be created. In this example, object A overlaps with object B and ground. And, object B and C are hanging in the air. This is a situation which cannot be practically realized. To cope with this kind of problem, we use two axiomatic rules as in ontology inference[4].

- Solid objects cannot penetrate other solid objects.
- Objects cannot float in the air by themselves.

Positions of objects violating these rules are rearranged. Specifically, to relocate objects, we need to select the reference objects. For example, in Fig.3(a), objects A, B, and C violate above rules. There are several ways of relocation according to the choice of a reference object. To deal with this choice problem, following three guidelines are used;

1. Select non-movable object as a reference object (for example, refrigerator)
2. Select the object having matched SIFT keypoints as many as possible since many matched SIFT keypoints implies that the object is correctly identified with high probability
3. Select the object with the largest volume

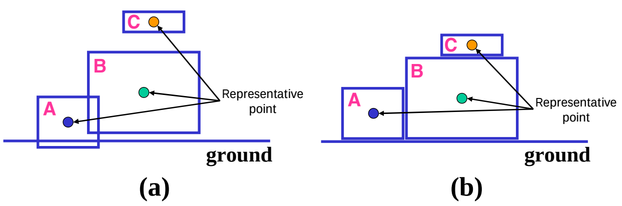


Fig. 3: *Before and After Compensation*

Fig.3(b) shows the result of object rearrangement using guidelines 1) and 3). After this process, the internal data of R(A), R(B), and R(C) are updated.

3. GENERATION OF PRIMITIVE-ST AND ST CONTEXT

3.1 Generation of Primitive-ST Context

Primitive-ST context is the classified data created by clustering primitive spatial data such as location, distance, and movement data. Thus, primitive-ST context is composed of three data (location, distance, movement). In the following definitions of primitive-ST data, *T* is the trustability which means how reliable the primitive-ST data are. Table 1 shows definitions used in generation of primitive-ST data.

Table 1: *Definitions for Primitive-ST Data*

Definitions	Description
t_s	start time
t_e	end time
$t_d = (t_e - t_s)$	time interval of interest
t_{freq}	sampling time
t_i	time acquiring i-th data
$N_{max} = t_d / t_{freq}$	maximum number of available data
e_R	threshold of position error
e_D	tolerance of distance error
T	trustability

Primitive Location Data Primitive location data of object ob(pR(ob)) shows that the object ob stays at the same location for a given period of time td. And pR(ob) is defined as pR(ob) = (id, t_s, t_e, obj, **R**(ob), T).
The primitive location data can be generated as follows;
For any time $t_i \in \{0 < t_s \leq t_i \leq t_e\}$

- Step 1: If $||\mathbf{R}(ob) - \mathbf{R}_{t_i}(ob)|| < e_R$ then $count = count + 1$
- Step 2: $T = count / N_{max}$
- Step 3: Generate $pR(ob)$

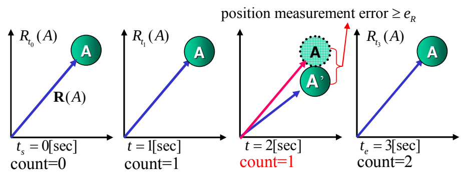


Fig. 4: *Example of Primitive Location Data Generation*

In Fig.4, the position of object A is **R**(A) at t = 0[sec]. Object A does not move at t = 1[sec]. Thus count is increased by one. At t = 2[sec], the position of object A is detected to be changed. Position measurement error is greater than a given threshold e_R. Thus, count for A will not be changed. And, a new R(A′) is created with count=1. In this case, object A does not move actually. At t = 3[sec], object A is still located at its initial position and thus count for A is increased. Here, td is 3[sec] and t_{freq} is 1[sec]. The trustability *T* for R(A) and R(A′) can be calculated as $count/N_{max} = 2/3 = 0.67$ and $count/N_{max} = 1/3 = 0.33$. As a result, following primitive location data are generated.

pR_{t03} (A) = (id, 0, 3, A, **R**(A), 0.67)
pR_{t03} (A') = (id, 0, 3, A′, **R**(A′), 0.33)

Primitive Distance Data Primitive distance data of object ob1 and ob2 (pD(ob1, ob2)) shows whether the distance between object ob1 and ob2 is changed for a give period of time t_d. And pD(ob1, ob2), is defined as pD(ob1, ob2) = (i_d, t_s, t_e, obj1, obj2, **D**(ob1, ob2), T).
The primitive distance data can be obtained by the following algorithm; For any time ti $\in \{0 < t_s \leq t_i \leq t_e\}$

- Step 1: if $\|\mathbf{D}(ob1, ob2)\| - \|\mathbf{D}_{ti}(ob1, ob2)\| < e_D$,
  then count = count + 1
- Step 2: $T = count/N_{max}$
- Step 3: Generate pD(ob1, ob2)

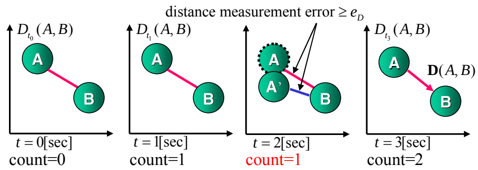


Fig. 5: *Example of Primitive Distance Data Generation*

In Fig.5, the distance between object A and B is not changed until t = 1[sec], which makes count be increased by one. At t = 2[sec], a new distance between two objects is detected to be changed. Distance measurement error is greater than a given threshold e_D. Thus, count for object A and B will not be changed. And, a new D(A′, B) is created with count=1. In this case, object A does not move actually. At t = 3[sec], two objects are still located at their position and thus, count for D(A, B) is increased. Thus, following primitive distance data can be generated as in that of primitive location data.

pD_{t03} (A, B) = (id, 0, 3, A, B, **D**(A, B), 0.67)
pD_{t03} (A', B) = (id, 0, 3, A′, B, **D**(A′, B), 0.33)

It is remarked that pR and pD with trustability less than 0.5 will be discarded.

Primitive Movement Data Primitive movement data of object ob(pM(ob)) shows that the object ob moves from the initial location for a given period of time t_d. And pM(ob) is defined as pM(ob) = (id, t_s, t_e, obj, **M**(ob), T).
The primitive movement data can be generated as follows; For any time $t_i \in \{0 < t_s \leq t_i \leq t_e\}$

- Step 1: if $\|\mathbf{R}_{ti}(ob) - \mathbf{R}_{ti-1}(ob)\| \geq e_R$,
  then count = count + 1
- Step 2: $T = count/N_{max}$
- Step 3: Generate pM(ob)

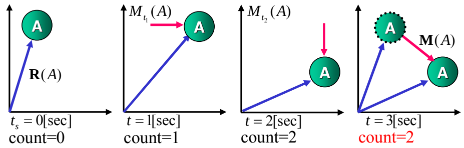


Fig. 6: *Example of Primitive Movement Data Generation*

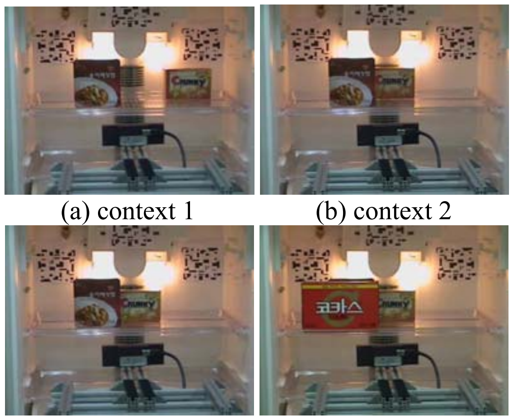
In Fig.6, object A moves successively at t = 1[sec] and t = 2[sec], so the value of count is changed twice. At t = 3[sec], object A is located at its previous position and count is not changed. Thus, primitive movement data can be generated as M_{t03} (A) = (id, 0, 3, A, **M**(A), 0.67).

3.2 ST Context Generation

ST contexts are data consisting of one or more Primitive-ST contexts with the same property. In this paper, 19 ST contexts are defined. Among them, some of ST contexts are shown in Table 2 together with corresponding context generation method, and their logical expressions.

4. EXPERIMENTAL RESULT

The ST context generation experiment was performed for objects in a refrigerator. For sequentially changing context in the refrigerator as shown in Fig.7, mobile robot is to generate primitive spatial data, primitive-ST contexts, and ST contexts. At $t = t_1$, there are two objects (obj2 and obj3) apart from each other in a refrigerator(obj1). At $t = t_2$ and t = t3, obj3 moves left(context 2) and forward(context 3), respectively. Finally, a large object(obj4) appears in front of obj2 and occludes it.



(a) context 1 (b) context 2
(c) context 3 (d) context 4

Fig. 7: *Sequentially Changing Contexts*

Fig.8 shows the display of our context generation system. In the left upper corner, images captured by stereo camera attached on top of the mobile robot are displayed continuously. In the right upper corner, the location of objects, where the location is obtained by our proposed point-based approach and compensated by our proposed axiomatic rules, can be also observed. At the bottom of Fig.8, logical expressions for ST contexts are displayed. Primitive spatial data, primitive-ST contexts, and ST contexts for this experiment are omitted for simplicity. And logical expressions are listed in Table 3. Note that the proposed system can extract the relationship between objects which are not directly visible by inferencing relationship among the primitive spatial data and ST-contexts created at previous times. In this experiment, although obj2 is occluded by obj4, contexts such as front(obj4, obj2) and front(obj2, obj3) were generated.

Table 2: *ST Contexts Generated in The Experiment*

Context	Generation method	Logical Expression
C_1	select all primitive location data for the period of time of interest	object-fixed
C_2	select all primitive distance data for the period of time of interest	fixed-distance
C_5	select primitive distance data which have short distance for specific period of time	object-near
C_7	generate inclusive spatial relation of fixed objects from location data for specific period of time	inside
C_8	select primitive movement data which are not included in C_3 for specific period of time	temp-moving
C_9	select distance data of objects which become near for specific period of time	move-near
C_{14}	select location data at current time	visible
C_{15}	generate left-and-right spatial relation of object from location data at current time	left
C_{16}	generate left-and-right spatial relation of object from location data at current time	front
C_{17}	select distance data of objects which is near robot at current time	near

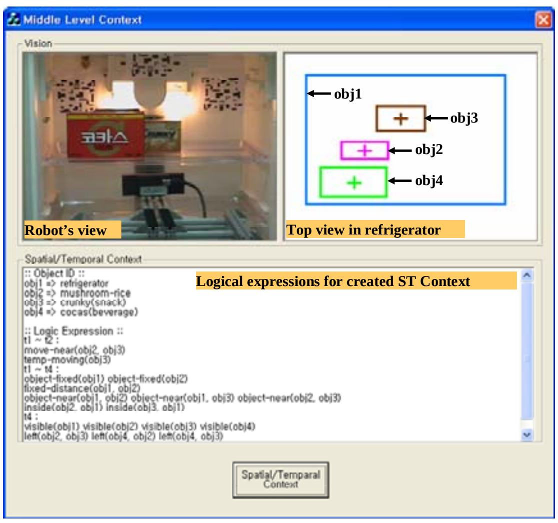


Fig. 8: *Experimental Environment: Inside a Refrigerator*

5. CONCLUSION

A vision-based context generation system was proposed, where SIFT features were employed to recognize known objects and a scheme of representative point-based object representation was addressed. After recognizing objects, primitive spatial data which include location of objects, distances between objects, and movement of objects were created. When the primitive spatial data were generated, a representative point implying an approximated center of an object was used for fast context generation. Position errors in primitive spatial data were compensated by using some axiomatic rules. The compensated primitive spatial data were clustered to form primitive-ST contexts. Finally, ST contexts were generated by classifying and/or inferencing these primitive-ST contexts. The validity of the proposed method was shown by experiment.

Table 3: *Logical Expression for Experiment*

time	Logical expression for contexts in refrigerator
$t_1 \sim t_2$	move-near(obj2, obj3), temp-moving(obj3)
$t_1 \sim t_4$	object-fixed(obj1), object-fixed(obj2), fixed-distance(obj1, obj2), object-near(obj1, obj2), object-near(obj1, obj3), object-near(obj2, obj3), inside(obj2, obj1), inside(obj3, obj1)
t_4	visible(obj1), visible(obj2), visible(obj3), visible(obj4), left(obj2, obj3), left(obj4, obj2), left(obj4, obj3), front(obj2, obj3), front(obj4, obj2), front(obj4, obj3), near(obj1), near(obj2), near(obj3), near(obj4)

6. ACKNOWLEDGEMENT

This work is supported by the Intelligent Robotics Development Program, one of the 21st Century Frontier R&D Programs funded by the Korea Ministry of Commerce, Industry and Energy. And, all correspondences of this work should be addressed to I. H. Suh.

7. REFERENCES

[1] B. Neumann, R. MAoller, "On Scene Interpretation with Description Logics." *FBI-B-257/04(Technical Report)*, Fachbereich Informatik, UniversitAat at Hamburg, 2004

[2] A. Torralba, K. Murphy, W. Freeman, M. Rubin, "Context-based vision system for place and object recognition." International Conference on Computer Vision, p273-280, 2003

[3] J. Pacheco, M. T. Escrig, F. Toledo, "Qualitative spatial reasoning on three-dimensional orientation point objects." *Proceedings of the International WorkShop on Qualitative Reasoning*, 2002

[4] E. Bozsak, M. Ehrig, S. Handschuh, A. Hotho, A. Maedce, B. Motik, D. Oberle, C. Schmitz, S. Staab, L. Stojanovic, "KAON-Towards a large scale semantic web." *Lecture notes in computer science* No.2455, p304-313, 2002