2-Scene Comic Creating System Based on the Distribution of Picture State Transition

Miki Ueno, Naoki Mori, and Keinosuke Matsumoto

Graduate School of Engineering
Osaka Prefecture University
1-1 Gakuencho, Nakaku, Sakai, Osaka 599-8531, Japan
ueno@ss.cs.osakafu-u.ac.jp

Abstract. Understanding picture by computer has become one of the most important topics in computer science. However, there are few researches have been reported about human like picture understanding by computer. The main reason of difficulty is that lots of picture expressions contain more lyric aspect than natural language expressions. Comic is one of the best target of picture understanding researches because pictures in comics express simply and clearly story, therefore we can presume that pictures in comics have strong universality. Picture understanding is defined as understanding situations and estimating transition between current scene and next scene. In this paper, The novel method which generates pictures using prepared picture parts and image objects databases is proposed. We also show the 2-scene comics creating system using user inputs picture and propose the representation of picture state transition.

Keywords: Automatic Drawing Picture, 2-scene Comic Creating System, Picture State Transition, Picture Model, Introducing Semantics to Picture, Comic Engineering.

1 Introduction

Creating products of human intellectual activities such as novels, comics and musics is ultimate purpose of artificial intelligence. The picture is one of the most important representations of such activities, and representation using pictures is a strong way to communicate because pictures in some cases include more information rather than text[1] and are understandable for various people regardless of age and national origin.

In the field of picture research, despite current advances of image recognition[2,3,4] and analyzing brain activities for images[5], there are few research about semantics of picture[6,7]. To solve this problem, we have proposed a novel dialogue system called *Picture Information Shared Conversation Agent*(**Pictgent**)[8] that shares conversational background knowledge by showing prepared pictures with model to user. Though we proposed the picture model to represent semantics of picture in Pictgent, we have not obtained complete one because of difficulty of defining what picture is. In order to solve the problem, we have analyzed the structure of existing comics whether there are certain relation between the story of 4-scene comics and transition of object. However, unexpectedly, we failed to obtain clear constitution of pictures in comics. We found that

S. Omatu et al. (eds.), *Distributed Computing and Artificial Intelligence, 11th International Conference,* Advances in Intelligent Systems and Computing 290,
DOI: 10.1007/978-3-319-07593-8_53,

lots of stories only use objects of human and balloon objects, hence the stories depend on natural language in balloon and it is difficult to create semantics of picture by existing comics. There have been reported another researches[9] of semantic representation of picture, but no research has proposed complete picture model.

In this paper, we tried to propose system which can create comics like pictures flexibly based on objects in the prepared dataset. We focused on 2-scene comics first because this is the minimum set of pictures which contains some kinds of object transition. After user inputing for 1st scene, the proposed method generates 2nd pictures based on 1st picture automatically with utilizing picture objects transition database.

To achieve computer understanding pictures, 3 issues are required as follow:

1. Estimate situations in pictures.
2. Represent feel or emotion of picture numerically.
3. Expect next scene after current pictures.

We propose the method of creating 2-scene comics by user' inputs in section 2. We show the constitution of application in section 3. Finally, in section 4, we present the conclusions of this study.

2 Proposed Method

In this section, we describe the proposed 2-scene comic creating system.

2.1 Representation of Picture in This Paper

In order to represent semantics of pictures for computer, we propose the "picture model" in which we define each of elements in pictures including background images and balloons as certain object.

It is difficult to create reasonable picture model for all pictures. Therefore, as previously mentioned, we focused on 2-scene comics because 2 sequential pictures have enough information rather than single picture. In this method, we should prepare dataset including images of objects, information of objects, types of areas, drawing operator, and transition. Each picture has above information and history, and those are representation of picture in this study.

2.2 Picture Model

In this study, initial picture p_0 is generated by system. p_0 is a blank picture or a background picture and has several areas determined by system. User can operate picture as user input. User operation set in time t is represented by $\boldsymbol{u_t}$, and $\boldsymbol{u_t}$ contains all kinds of user's operation for picture p_t. p_{t+1} is created applying $\boldsymbol{u_t}$ to p_t. We describe this as follows:

$$p_{t+1} = \prod_{u_i \in \boldsymbol{u_t}} u_i p_t \tag{1}$$

where u_i represents each user's operations in time t.

Fig. 1. cup (original image)

Fig. 2. cup (apply $\hat{\phi}_{\text{divide}}$)

Fig. 3. cup (apply $\hat{\phi}_{\text{rotate}}\hat{\phi}_{\text{divide}}$)

The relation among p_t, o_t, $\boldsymbol{u_t}$, $\{O_t\}$ and $\{U_t\}$ is following:

$$\{O_t\} = \{o_t \in p_t\} \tag{2}$$

$$\{U_t\} = \{\boldsymbol{u_1}, \boldsymbol{u_2}, ..., \boldsymbol{u_t}\} \tag{3}$$

$$p_{t+1} = \prod_{u_i \in \boldsymbol{u_t}} u_i p_t = \prod_{u_i \in \boldsymbol{u_t}} u_i \prod_{u_i \in \boldsymbol{u_{t-1}}} u_i p_{t-1} \tag{4}$$

$$... \tag{5}$$

$$= \prod_{j=0}^{t} \prod_{u_i \in \boldsymbol{u_j}} u_i p_0 \tag{6}$$

Definite operation of $u_i \in \boldsymbol{u_t}$ is represented by drawing operator ϕ and operand object in p_t. Each ϕ also has specific augments such as color, angle or other objects. Object o_t is able to represent object transition. Object o_n in n-th scene has Markov model as bellow:

$$o_n = \prod_{i=0}^{N_n^{\max}} \phi_i o_0 \tag{7}$$

where $N^{\max}$ is total number of ϕ to applied to o. $N_n^{\max} \geq n - 1$ because each u_i contains at least one ϕ.

2.3 Base Drawing Operator $\hat{\boldsymbol{\phi}}$

Drawing operator ϕ has the function of transform for pictures. Product of several drawing operators is also drawing operator and the set of base operator is defined as $\{\hat{\phi}\}$. Increasing the number of base operators, the ability of representation of the system also increases more and more because system can create lots of new images utilizing current objects.

2.4 Implementation of Drawing Operator $\boldsymbol{\phi}$

In this paper, the system is written by Java and the operator is implemented by command pattern in design pattern[10]. Base drawing operator corresponds to command

and product of drawing operator corresponds to macro-command. In Java implementation, product of operator is represented by using queue.

For example, ϕ_{crack} which represents cracked transition is composed of base drawing operator $\hat{\phi}_{\text{divide}}$ $\hat{\phi}_{\text{rotate}}$.

ϕ_{crack} is applied to object o from right side, 3 base drawing operators are applied to o as follows:

$$\phi_{\text{crack}} o = \hat{\phi}_{\text{rotate}} \hat{\phi}_{\text{divide}} o \tag{8}$$

Figure 1-3 show the variation of image by applying base drawing operator of ϕ_{crack}.

2.5 Flow of Proposed System

The basic flow of proposed system is described as follows[1]:

1. Select the base image as background for initial states. We define this picture as p_0.
2. System waits for user inputs. User can decide several actions to p_0. In this paper, user action is only setting an object in objects set shown by system GUI. User can set their interested object on one area of p_0 with drawing operator and emotion information. Any number of objects are added to p_0 under condition of existing only one object in each area. When user put an object to p'_0, new areas decided by set object are added to picture.
3. After applying all user inputs $\boldsymbol{u}$ to p_0, p'_0 is obtained. This p'_0 become 1st scene.
4. System tries to change objects in p'_0 based on user decided drawing operator and emotion information to target object. New image of changed objects is obtained by transition DB.
5. Applying all changes to p'_0, p_1 is obtained. p_1 represents 2nd scene.
   2-scene comic : p_1 is final output.
   ***n*-scene comic :** p'_1 becomes p_2. System waits for user inputs again and repeats above flow until p'_{n-1} becomes p_n.

2.6 User Input *u*

As mentioned above, user action is putting some objects on one area of the picture. User decides the object and the area to set shown by system. Set objects in $\{O_{\text{set}}\}$ and picture objects in p_t are different as though set objects become picture objects after user setting, where $\{O_{\text{set}}\}$ represents a group which is applicable to picture. When user sets the object to picture, user also sets the "drawing operator" and "emotion" information to decide transition pattern of setting object. Finally, user can decide following information by one user input.

Object o **in** $\{O_{\text{set}}\}$**.** The object information of user setting object.
Area a**.** Set area of o.

[1] p'_t represents the latest picture after user decides several actions for p_t in eq. 1. The operator applied to p'_t is decided uniquely based on user action and p_{t+1} is obtained after it is executed.

Table 1. Object DB

ID	Object Name	Class Name	Transition:Frequency	Image File Path	Included Area ID
1	cup	container	drink:4,crack:2,drop:1	/fig/cup.png	null
2	cat	animal	bite:5,attach oneself to:3,run away:2	/fig/cat.png	null
3	girl	human	stand:3,sit:1, $\cdots$	/fig/girl.png	2, 3, 4

Some objects have several areas with IDs where user can set a new object.
Class name is important information to use external resources such as Wordnet.

Table 2. Transition DB

ID	Object ID	Area ID	Operator Name	Emotion Name	Drawing Operator
1	*	*	divide	sad	crack(Object ID)
2	*	2	*	*	drop(Object ID)

*: Wildcard
In database, we describe $\phi_{\text{crack}} \rightarrow \text{crack}(x)$.

Operator m**.** Operator type of setting object o.
Emotion e**.** User can set the emotion e of setting object. This information effects transition of o.

Using those information, system decide the transition of each objects in the current picture. All transitions are represented by change of image. Therefore, system have to prepare new images of all objects in picture after transition. Creating new images has the difficulty of huge range of variation. To solve this problem, we prepared only basic image sets and created various images to apply drawing operator to an images. For example, we can create various image pattern of cup such as cracked cup, rotated cup or different colored cup from only one cup image.

2.7 Databases in the Proposed System

Our final purpose is to establish a database which can represent the transition of objects in 2.2. However, to make this, we have to decide the complete distribution model of each object. Since this is a difficult task, we simplified the object transition that all objects have only 1 transition result.

The transition pattern of each object is decided by type of object, area of object, operator for objects and emotion of objects uniquely. We created those patterns based on existing pattern of comics and common sense, e.g. cup is cracked or we use the umbrella in the rain. Although we set only 1 typical pattern in this study, transition patterns will be extended based on distribution of existing data in future work. We created 4 databases (DBs) for managing those information as follows.

Object DB. Object DB manages the information of all objects. In object DB, each object has the attributes of object name, class name, what kinds of transition and the rarity in existed comics and common sense, image file path of image, area IDs. Table 1 shows the example of object DB.

Table 3. AreaName DB

ID	Area Name
1	whole
2	ground
3	hand

Table 4. Area DB

ID	Area ID	Object ID	Start point of area	End point of area
1	1	*	(*, 170)	(*, 190)
2	2	3	(20, 40)	(40, 60)

*: Wildcard

Transition DB. Transition DB manages the distribution of object transition. This database contains user input $\boldsymbol{u}$ as key and drawing operator ϕ as value. User Input $\boldsymbol{u}$ is represented by following equation:

$$\boldsymbol{u} = (\text{object } o, \text{area } a, \text{operatorname } m, \text{emotion } e) \tag{9}$$

Operator name m is name string corresponding to each base drawing operator name. Table 2 shows the example of transition DB.

In current transition DB, only 1 operator is considered for each $\boldsymbol{u}$ although $\boldsymbol{u}$ has the distribution of several operators.

Area Name DB and Area DB. Area Name DB and area DB manage the area information in picture. Though area is not appeared visually in the picture, this is important to control the location of set objects. When user sets the object by mouse operation, user can recognize the area by specific color.

Table 3 shows area name DB which has the relation between area ID and area name. Table 4 shows area database which has the start point and the end point of area. The start and end point of same area name are different among objects.

3 Application

We developed 2-scene comic creating system based on the proposed method. Figure 4 shows the outline of system. Figure 5 shows picture model of 2nd scene.

Followings are explanations of each part of system corresponding to the number in figure 4.

1. Selectable background image. Select button of the base image for 1st scene. The blank picture or prepared picture is able to be selected.
2. 1st scene image. The display area of 1st scene image. After applying user input, p_0 becomes p'_0.
3. Selectable areas. Areas where user can set object in 1st scene.
4. 2nd scene image. The display area of 2nd scene image which is generated by applying user input of 1st scene.
5. Operator & emotion. The display area of operator and emotion for set objects.
6. Selectable object. The display area of $\{O_{\text{set}}\}$ which is set of user selectable objects.
7. Picture model. Picture model XML of 2nd scene.

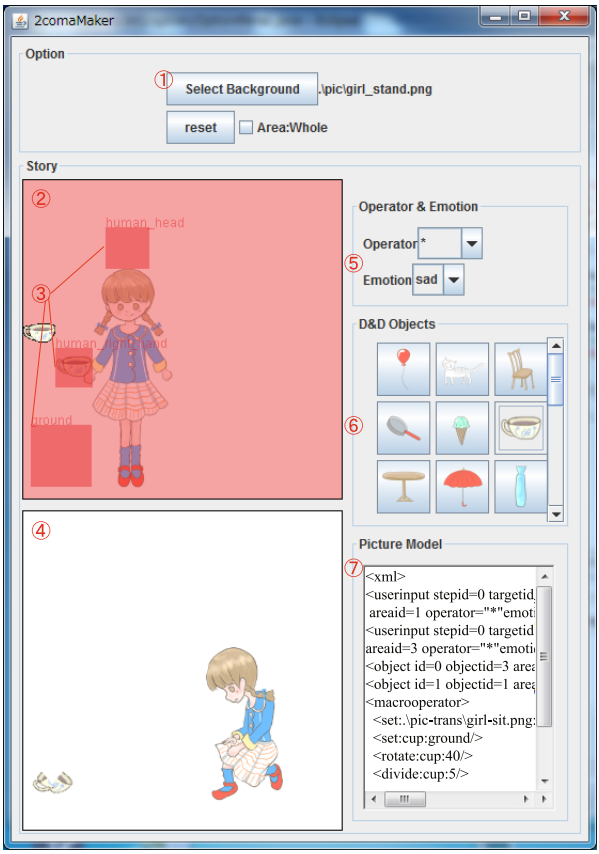

Fig. 4. Outline of 2-scene comic creating system

```
<xml>
<userinput stepid=0 targetid=3 areaid=1 operator="*"emotion="*"/>
<userinput stepid=0 targetid=1 areaid=3 operator="*"emotion="sad"/>
<object id=0 objectid=3 areaid=1 />
<object id=1 objectid=1 areaid=2 />
<macrooperator>
  <set:.\pic-trans\girl-sit.png:whole/>
  <set:cup:ground/>
  <rotate:cup:40/>
  <divide:cup:5/>
  <zoom:cup:50@40/>
</macrooperator>
</object>
</xml>
```

Fig. 5. XML of 2nd scene

Table 5. XML Tag of Picture Model

Tag Name	Explanation
<object>	Information of object. The attributes of this tag are object ID in picture, ID of object DB and area ID.
<macrooperator>	Information of applied drawing operator to the object.
<userinput>	Information of user's inputs. The attributes of this tag are picture step, target object ID, area ID, operator name and emotion name.

4 Conclusion

In this paper, we proposed the method of generating pictures automatically by applying drawing operators to prepared objects.

Furthermore, we developed creating 2-scene comic system utilizing the proposed method. Followings are important future work: (1) What kinds of information object has. (2) How to relate user input and transition. (3) How much drawing operators are required. If we solve those problems completely, we will create conversational system between humans and computers, interactive picture book and drawing application which displays comments about user's drawing.

Acknowledgment. This work was supported by Grant-in-Aid for JSPS Fellows(Grant Number:25·10978).

References

1. Kurlander, D., Skelly, T., Salesin, D.: Comic Chat. In: SIGGRAPH 1996 Conf. Proc., pp. 225–236 (1996)
2. Herot, C.F.: Graphical Input through Machine Recognition of Sketches. Computer Graphics (Proc. SIGGRAPH) 10(2), 97–102 (1976)
3. Eitz, M., Hays, J., Alexa, M.: How Do Humans Sketch Objects? ACM Trans. Graph. (Proc. SIGGRAPH) 31, 44:1–44:10 (2012)
4. Fu, H., Zhou, S., Liu, L., Mitra, N.J.: Animated Construction of Line Drawings. ACM Trans. Graph. (Proc. SIGGRAPH ASIA) 30(6) (2011)
5. Walther, D.B., Chai, B., Caddigan, E., Beck, D.M., Fei-Fei, L.: Simple line drawings suffice for functional MRI decoding of natural scene categories. Proc. National Academy of Sciences 108(23), 9661–9666 (2011)
6. Ito, K., Matsuda, M., Durst, M.J., Hasida, K.: SVG Pictograms with Natural Language Based and Semantic Information. SVG Open (2007)
7. Gross, M.D., Do, E.Y.: Drawing on the Back of an Envelope: A Framework for Interacting with Application Programs by Freehand Drawing. Computers & Graphics 24(6), 835–849 (2000)

8. Ueno, M., Mori, N., Matsumoto, K.: Picture Information Shared Conversation Agent: Pictgent. In: Omatu, S., Paz Santana, J.F., González, S.R., Molina, J.M., Bernardos, A.M., Rodríguez, J.M.C. (eds.) Distributed Computing and Artificial Intelligence. AISC, vol. 151, pp. 91–94. Springer, Heidelberg (2012)
9. Arai, N., Matsuzaki, T.: Can a Robot Get into the University of Tokyo?: The Artificial Intelligence Project at the National Institute of Informatics. Journal of the Japanese Society for Artificial Intelligence 27(5), 463–469 (2012)
10. Gamma, E., Helm, R., Johnson, R., Vlisside, J.: Design Patterns: Elements of Reusable Object- Oriented Software. Addison-Wesley (1995)