Proceedings of the 2017 IEEE
International Conference on Robotics and Biomimetics
December 5-8, 2017, Macau SAR, China

A Supervisory Hierarchical Control Approach for Text to 2D Scene Generation*

Yu Cheng, Zhiyong Sun, Sheng Bi, Congjian Li, and Ning Xi

Abstract— **Photo-realistic image generation conditioned on text is an important problem and would have a wide range of applications. The gap between the high level abstract semantic information and the detailed low level image elements and operations on them hampers utilization of text-to-image systems. Many research works have been proposed to bridge the gap. However, most existing approaches do not allow operations on objects during the image synthesis. To generate photo-realistic images with more flexibility, in this work, we propose a two level hierarchical control system framework for image synthesis. The high level system acts as task planner and cope with abstract concepts extracted from input text. While the lower level system plays the role of action planner, which takes command from task planner and generates an action sequence for lower level implementation. The proposed approach allows control of the location and dimension of each object appeared in the text input, which helps to generate more complex scene. In addition, the proposed method allows to synthesize images with higher resolution. Experimental results have been provided to validate the proposed method.**

Fig. 1. Generated scene for "A dog is at the right side of the car. The car is on the beach". The relative pairwise spatial relations can be tuned during image synthesis using language commands.

I. INTRODUCTION

Photo-realistic image generation conditioned on text is an important problem and would have a wide range of applications, including photo-editing, computer-aided design, etc. However, the gap between the high level abstract semantic information and the detailed low level image elements hampers utilization of text-to-image systems.

In order for the system of text-to-image generation to be useful, synthesized images should have at least discernible image resolution, and support control on elements that comprise of target images, as shown in Figure 1.

Compelling image syntheses towards this goal have been demonstrated using Conditional Random Field (CRF) [1], Variational Autoencoders [2], GAN based networks [3], etc. However, existing approaches fail to generate images with sufficient resolution, and don't allow for control of object size, object position, and relative spatial relations during the synthesis process.

To generate photo-realistic images with more complex scenes and higher resolution, it would be beneficial to incorporate control methods into the image synthesis models. Our proposed models employ a two-level supervisory control hierarchy as the high level controller to regulate the image synthesis process. Objects that compose the image conditioned on input text can be controlled via real-time language commands from a human observer. This encodes human experience and allows online tuning of the image to better fulfill human aesthetic values and common sense.

The main contributions are as follows: (1) a novel method for image synthesis that allows control of object location and dimension based on spatial relation and dimensional constraints specified in the text descriptions (2) In particular, it supports online tuning of the objects extracted from other images during the image synthesis, yielding photo-realistic images with more complex scene.

II. RELATED WORK

Text to Image. Image synthesis conditioned on text has been researched along two directions: object level and pixel level. Object level image synthesis refers to organizing semantic elements to generate images conditioned on text descriptions. While the later means to map semantic text to most likely object pixel distributions learned from training data.

For object level image synthesis, Zitnick et al. [1] build a conditional random field (CRF) model to synthesize clipart images conditioned on predicate tuples extracted from text input. A predicate tuple contains two nouns and a relation. Chang et al. [4] employ Bayesian probability to generate 3D scenes conditioned on text containing objects

*This work is partially supported by the National Science Foundation under Grant CNS-1320561 and IIS-1208390, the U.S. Army Research Laboratory, and the U. S. Army Research Office under the Grant W911NF-11-D-0001, W911NF-09-1-0321, W911NF-10-1-0358, W911NF-14-1-0327 and W911NF-16-1-0572.

Yu Cheng, Zhiyong Sun, Sheng Bi, Congjian Li, and Ning Xi are with the Department of Industrial and Manufacturing Systems Engineering, The University of Hong Kong, Hong Kong (Email: xining@hku.hk)

Yu Cheng is with the Department of Electrical and Computer Engineering, Michigan State University, East Lansing, MI, 48823, USA (Email: chengyu9@msu.edu)

978-1-5386-3742-5/17/$31.00 © 2017 IEEE

and spatial relations that could be expressed implicitly. Text descriptions are mapped to operations on objects. Once the image synthesis is complete, instructors are not able to modify the image anymore, even if the generated images do not reach the expectation of the instructors. In comparison with their methods, the proposed approach in this paper allows continued tuning on generated images, which helps to generate more realistic scenes.

More recent works focus on pixel level image synthesis. Reed et al. [3] train generative adversarial networks (GANs) to generate 64×64 images for flowers and birds with respect to text descriptions. Their follow-up work was able to generate 128×128 images via additional annotations on locations of object part [5]. Mansimov et al. [2] synthesize images from text descriptions using a variational recurrent autoencoder with attention, which is similar to DRAW [6]. Han et al. [7] propose stacked GANs to generate 256×256 images for birds and flowers conditioned on text descriptions, which has higher resolution than previous works. Compared with pixel level methods, the proposed approach is able to generate complex scenes with higher resolution. In addition, pixel level text to image synthesis approaches suffers the same problem of uncontrollability of semantic objects.

Natural Language based Robot Control. Using natural language to control robot behaviors makes it easier for people to employ robots, especially for novice users. To some extent, text to 2D scene synthesis can also be seen as a natural language based robot control problem. One key step of natural language based robot control is to translate the natural language commands into formal and unambiguous representations such that robots are able to understand and implement the instructions.

Based on how the language commands are processed into executable action plans, existed methods can be divided into two categories. Methods of the first category translates linguistic commands into executable action plans based on a set of predefined rules distilled from peoples experience. Lauria et al. [8] use hand-coded grammars to map navigational commands into predefined action templates. MacMahon et al. [9] define a set of rules to process linguistic input into predicate-argument structure. Rybski et al. [10] propose a method to search keywords in the instructions that are in the form of conditional branch structure and then transform them into conditional structures similar to that of programming language. Cantrell et al. [11] propose a similar method. The difference is they map instructions with a set of templates instead of keyword searching. Kress-Gazit et al. [12][13] use linear temporal logic (LTL) formulas to represent language commands and then compute an automaton based on the LTL expressions to serve as the high level controller. Cheng et al. [14] translate language commands into target states and use automata as task planner to supervise system state transitions conditioned on linguistic input. The other type of natural language based robot control is by using data-driven approaches to learn the implicit rules from data instead of using hand-designed explicit ones. The probability based methods differ in their probabilistic models (e.g., linear model [15], hidden Markov model (HMM) [16], Markov logic network (MLN) [17], conditional random field (CRF) [18], etc.), formal representations (e.g., predicate-argument structure [19], lambda calculus [20], graphical representation [21], etc.), designed features for training the models and so on.

Most of existing approaches use plain structure to build the behavior model conditioned on input language commands. In comparison, our proposed work employs a hierarchical architecture that can handle commands with different logic depth. It helps to bridge high level semantic concepts to lower level robot operations.

The rest of the paper is organized as follows. Section III briefly introduces the theory of supervisory control. Section IV illustrates the proposed approach in detail. In Section V, the proposed method has been applied for image synthesis with experimental results. Finally, Section VI concludes the paper.

III. IMAGE SYNTHESIS APPROACH

In this section, we will illustrate the proposed framework and approach in detail. Figure 2 shows an overview of the proposed image synthesis system framework. The main components of the system are the task planner and action planner. The roles of these components are in the following.

The task planner takes a set of target states from the natural language processing module as its input and organize the states into Dependency Relation Matrix (DRM) for subtask planning. Locations of some objects are dependent on the spatial relations with other objects in text descriptions. The locations of dependent objects can be determined only if the locations of objects they depend on have already been decided. The task planner generates an ordered sequence of subtasks based on the spatial relation dependency between objects.

The action planner is realized by using finite state machines. In fact, task planner and action planner form a hierarchical supervisory control structure [22][23] to cope with commands of different logic depth. The action planner contains a set of concatenation rules for lower level actions. Each action is represented as a finite state machine. As an action planner, it generates action plans with respect to the given task represented as the goal states (the output from task planner) and temporal states (extracted from temporal sensory information).

A. High Level System Modeling

The high level state set is comprised of parameterized pairwise spatial relations. In this work we consider six pairwise spatial relations: *on, under, left, right, front, and back*. It should be noted that the proposed approach are not limited to these 6 relations. Parameterization of states depends on the text specifications. The high level event set is comprised of parameterized instances of $operation\_on(object)$, where $object$ is a symbol variable. The set of marker state contains only the target states. Compared with fixed state transitions in classical supervisory control applications, we propose

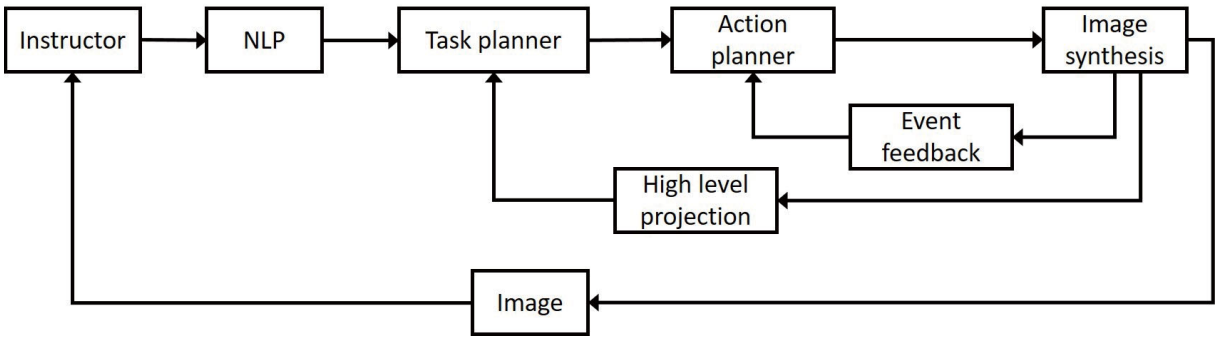

Fig. 2. Proposed image synthesis system framework. The system framework has three loops. The innermost loop is the action level, which monitors and controls the behaviors of lower level. The intermediate loop is the task level, the status of image synthesis system is projected into states of task planner through high level projection. Completion of current subtask will stimulate the task planner to issue next subtask. The outermost loop is perceptive feedback, through which the instructor adjusts the synthesized scene according to individual aesthetic value and common sense.

a method to flexibly plan state transitions based on text descriptions.

The text descriptions are translated into a set of target states and then organized into Dependency Relation Matrix (DRM) matrix. A DRM is represented as:

$$\begin{array}{c} \\ O_1 \\ O_2 \\ \vdots \\ O_n \end{array} \begin{array}{c} \begin{array}{cccc} O_1 & O_2 & \cdots & O_n \end{array} \\ \begin{bmatrix} [DRM]_{11} & [DRM]_{12} & \cdots & [DRM]_{1n} \\ [DRM]_{21} & [DRM]_{22} & \cdots & [DRM]_{2n} \\ \vdots & \ddots & \ddots & \vdots \\ [DRM]_{n1} & [DRM]_{n2} & \cdots & [DRM]_{nn} \end{bmatrix} \end{array}$$

It is a $n \times n$ matrix, where n is the number of items appeared in the text. The column index of DRM denotes the object index from the top to the bottom. The row index represents object index with the same order from the left to the right. Each element in the DRM captures the dependency relation between the column object over the row object. The value of an element in the DRM reflects the relative priority between sets of operations on two items. The element at the intersection of the i_{th} row and j_{th} column of the matrix can be defined as:

$$[DRM]_{ij} = \begin{cases} 0, \text{relation with the object itself,} \\ 1, i^{th} \text{ object and } j^{th} \text{ object are not} \\ \text{dependent,} \\ 2(-2), i^{th} \text{ object supports (is dependent on)} \\ j^{th} \text{ object.} \end{cases} \tag{1}$$

The raw output of Natural Language Processing (NLP) contains only the relations and involved items appeared in the commands. Each relation is regarded as a subtask. To figure out the order of subtask sequence, we derive a goal DRM (GDRM) that contains all the dependency relations between each pair of items using following rules:

$$\begin{aligned} &[DRM]_{ij} = relation \\ &\Rightarrow \begin{cases} [DRM]_{ji} = -relation \quad \text{if } relation \neq 1, \\ [DRM]_{ji} = relation \quad \text{if } relation = 1 \\ \text{and } [DRM]_{ji} = 1, \\ [DRM]_{ij} = -[DRM]_{ji} \quad \text{if } relation = 1 \\ \text{and } [DRM]_{ji} \neq 1, \end{cases} \end{aligned} \tag{2}$$

where $relation \in R$, $R = \{-2, 1, 0, 1, 2\}$ depends on the relation definition.

$$\begin{aligned} &\begin{cases} [DRM]_{ij} = relation \\ [DRM]_{jk} = relation \end{cases} \\ &\Rightarrow [DRM]_{ik} = relation \end{aligned} \tag{3}$$

At the same time, sensory information is processed into DRM representation in a similar manner, denoted as temporal DRM (TDRM). Before the robot executes the task, GDRM and initial TDRM are used to derive a feasible sequence of subtasks. Subtract GDRM by initial TDRM, we have the error DRM, denoted as DRM_e. Its elements represent the difference between the initial configuration and goal configuration.

$$DRM_e = GDRM - TDRM \tag{4}$$

A vector can be obtained by summing up all the relation values for each item corresponding to other items, as represented in Equation 5. The smaller of the value, the higher the priority of the subtask. The order of the subtasks can be determined through sorting the elements of the vector in an ascending way.

$$\begin{aligned} &f(DRM_e) = \\ &\left[\sum_{j=1}^{n}(a_{1j}), \quad \sum_{j=1}^{n}(a_{2j}), \quad \cdots, \quad \sum_{j=1}^{n}(a_{nj})\right]^T \end{aligned} \tag{5}$$

In this proposed work, each subtask means an ordered set of manipulations of objects conditioned on text. Ordering objects to be painted on the image discovers implicit priority among these elements, which is beneficial for synthesizing complex scenes with compact descriptions as the way humans usually communicate with each other.

B. Lower Level System Modeling

Lower level action planner takes commands from high level task planner and plans atomic operations of image synthesis. The event set of lower level action planner includes ten predefined atomic actions, as shown in Table I. The first seven actions are used for scene generation, while the later three actions are used for image tuning. The state set of action planner can be determined based on the event set. An automaton model can be obtained through parallel operation on each physically allowable state transitions [24]. However, the synthesized model may contain undesired behaviors that violate system constraints. Trimming the synthesized model with respect to constraints on both the system and image generation, we can obtain the action planner model represented as automata. Figure 3 shows the model for initial scene synthesis.

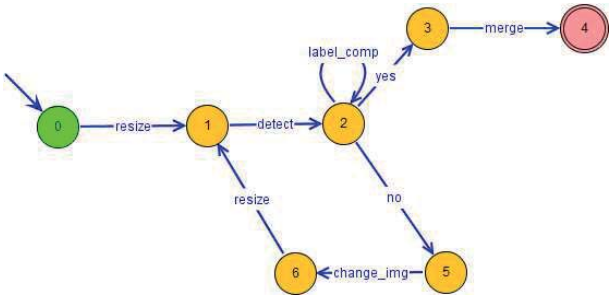

Fig. 3. Synthesized action planner model for initial scene synthesis based on target image composition and system constraints.

IV. EXPERIMENTAL RESULTS

A. Image Synthesis System

Text Description Parsing. During the parsing of text we identify the scene type, desired object types, and their relative spatial relations. The text instructions are firstly processed syntactically using Stanford parser [25]. The scene type and object type are determined by matching the words in the utterance against known scene types and object types in the database. Spatial relations are determined in a similar way.

Image Segmentation. The ability to manipulate semantic objects synthesized on images is based on object segmentation. The work presented by Zheng et al. [26] is used, which utilizes deep learning techniques and probabilistic graphical models for semantic image segmentation. It is formulated as pixel-level labelling tasks. The output of the segmentation is a 2D matrix that has the same dimension as the input image. Each value of the matrix denotes the object category of the pixel at the same location of the input image.

Location Instantiation Inference. Symbolic spatial relations carry semantic meanings of abstract concepts. Its limited information is not sufficient for instantiation of initial placement of objects. To provide reasonable coordinates of initial placement on the image, we train the prior statistical distributions of pairwise objects under each relation. The position of initial placement is calculated using Bayesian theorem.

B. Image Synthesis Illustration

Take the example as shown in Figure 1, the text description of the desired image is: *A dog is at the right side of the car. The car is on the beach*. The text description is firstly translated into state representation by Natural Language Processing (NLP) module:

$$Right(dog, car), On(car, beach)$$

In this example, each state can be reagarded as a subtask. The states are organized into Dependency Relation Matrix (DRM) for task planning. Since the location of the dog is dependent on that of the car, operations on the car has higher priority than operations on the dog. DRM based framework has the advantage of planning a reasonable task sequence under the situation when the text descriptions have mixed orders. Simply following the order of instructions as specified by the operator may lead to failures of tasks.

If the image segmentation fails to detect anything in the background image, the to-be-added object can be placed according to designer's preference. In this work, we choose to place the object at the lower left corner, as shown in subfigure (2) of the first image synthesis example in Figure 4.

After adding each object into the scene, the instructor can tune the location and size of the newly added item to make the scene looks more realistic, as shown in subfigures (2), (3), and (4) of the same image synthesis example. After the tuning process, or if no extra operations are needed after the addition, the next subtask in the ordered task sequence will be executed. This operation manner on each subtask will be implemented iteratively until the task is completed. The outcome corresponding to the input text is shown as subfigure (6) of the same image sequence. Figure 4 displays more examples of generated scenes, including novel scenes that comprised of common objects doing unusual things.

C. Discussion

We observe that the failures to generate desired scenes are due to two major reasons: parsing error and segmentation error. We use a deterministic approach to map text directly to semantic states needed for later use, which hampers the system from dealing with more diverse utterances.

In addition, the unreliability of image segmentation can also cause failures of image synthesis. Sometimes image segmentation algorithm falsely segment the image, resulting in incomplete object. Also, the image segmentation algorithm assigns wrong category labels every now and then.

V. CONCLUSION

In this work, we propose a novel approach for text to 2D scene synthesis. The proposed method firstly translates input text descriptions of the target image into a set of states. Each state is represented as a spatial relation with two items. Then the set of states are organized into Dependency Relation Matrix (DRM) for task planning. This helps to avoid task failure due to simply following the orders of text instructions. Action planner takes each subtask in state representation as

TABLE I

LIST OF BASIC ACTIONS FOR IMAGE SYNTHESIS.

Primitive Actions	Description
Resize	Adjust the size of the image
Detect	Detect object category
Label_compare	Compare category labels between objects
Merge	Add object into current scene
Yes	Confirm the previous operation
No	Deny the previous operation
Change_img	Change another image from the image base
Zoom in	Enlarge the size of an object
Zoom out	Reduce the size of an object
Move	Change the position of an object

input and generate an ordered action sequence for lower level system to implement.

Compared with existing image synthesis methods, the proposed approach allows control over locations and dimensions of added items. The extra control helps to compensate for spatial priors if trained with insufficient data. Also, it helps to generate more complex scenes with text descriptions. Furthermore, the proposed framework forms a two-level hierarchy to cope with abstract high-level semantic concepts and detailed lower-level robot operations separately, and will help to analyze the image synthesis system model to improve its performance. Additionally, the developed image synthesis system can be used to collect data for future language parser and spatial priors training.

REFERENCES

[1] C. L. Zitnick, D. Parikh, and L. Vanderwende, "Learning the visual interpretation of sentences," in *Proc. of IEEE International Conference on Computer Vision (ICCV)*, 2013, pp. 1681–1688.

[2] E. Mansimov, E. Parisotto, J. L. Ba, and R. Salakhutdinov, "Generating images from captions with attention," *arXiv preprint arXiv:1511.02793*, 2015.

[3] S. Reed, Z. Akata, X. Yan, L. Logeswaran, B. Schiele, and H. Lee, "Generative adversarial text to image synthesis," *arXiv preprint arXiv:1605.05396*, 2016.

[4] A. X. Chang, M. Savva, and C. D. Manning, "Learning spatial knowledge for text to 3d scene generation." in *EMNLP*, 2014, pp. 2028–2038.

[5] S. E. Reed, Z. Akata, S. Mohan, S. Tenka, B. Schiele, and H. Lee, "Learning what and where to draw," in *Proc. of Advances in Neural Information Processing Systems (NIPS)*, 2016, pp. 217–225.

[6] K. Gregor, I. Danihelka, A. Graves, D. J. Rezende, and D. Wierstra, "Draw: A recurrent neural network for image generation," *arXiv preprint arXiv:1502.04623*, 2015.

[7] H. Zhang, T. Xu, H. Li, S. Zhang, X. Huang, X. Wang, and D. Metaxas, "Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks," *arXiv preprint arXiv:1612.03242*, 2016.

[8] S. Lauria, G. Bugmann, T. Kyriacou, J. Bos, and E. Klein, "Personal robot training via natural-language instructions," *IEEE Intelligent systems*, vol. 16, no. 3, pp. 38–45, 2001.

[9] M. MacMahon, B. Stankiewicz, and B. Kuipers, "Walk the talk: Connecting language, knowledge, and action in route instructions," *Def*, vol. 2, no. 6, p. 4, 2006.

[10] P. E. Rybski, J. Stolarz, K. Yoon, and M. Veloso, "Using dialog and human observations to dictate tasks to a learning robot assistant," *Intelligent Service Robotics*, vol. 1, no. 2, pp. 159–167, 2008.

[11] R. Cantrell, K. Talamadupula, P. Schermerhorn, J. Benton, S. Kambhampati, and M. Scheutz, "Tell me when and why to do it! run-time planner model updates via natural language instruction," in *Proc. of ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, 2012, pp. 471–478.

[12] H. Kress-Gazit, G. E. Fainekos, and G. J. Pappas, "Temporal-logic-based reactive mission and motion planning," *IEEE Transactions on Robotics*, vol. 25, no. 6, pp. 1370–1381, 2009.

[13] C. Lignos, V. Raman, C. Finucane, M. Marcus, and H. Kress-Gazit, "Provably correct reactive control from natural language," *Autonomous Robots*, vol. 38, no. 1, pp. 89–105, 2015.

[14] Y. Cheng, Y. Jia, R. Fang, L. She, N. Xi, and J. Chai, "Modelling and analysis of natural language controlled robotic systems," *IFAC Proceedings Volumes*, vol. 47, no. 3, pp. 11 767–11 772, 2014.

[15] N. Shimizu and A. R. Haas, "Learning to follow navigational route instructions." in *IJCAI*, vol. 9, 2009, pp. 1488–1493.

[16] W. Takano, I. Kusajima, and Y. Nakamura, "Generating action descriptions from statistically integrated representations of human motions and sentences," *Neural Networks*, vol. 80, pp. 1–8, 2016.

[17] G. Lisca, D. Nyga, F. Bálint-Benczédi, H. Langer, and M. Beetz, "Towards robots conducting chemical experiments," in *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2015, pp. 5202–5208.

[18] D. K. Misra, J. Sung, K. Lee, and A. Saxena, "Tell me dave: Context-sensitive grounding of natural language to manipulation instructions," *The International Journal of Robotics Research*, vol. 35, pp. 281–300, 2015.

[19] L. She, Y. Cheng, J. Y. Chai, Y. Jia, S. Yang, and N. Xi, "Teaching robots new actions through natural language instructions," in *Proc. of IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, 2014, pp. 868–873.

[20] C. Matuszek, E. Herbst, L. Zettlemoyer, and D. Fox, "Learning to parse natural language commands to a robot control system," in *Experimental Robotics*. Springer, 2013, pp. 403–415.

[21] T. Kollar, S. Tellex, M. R. Walter, A. Huang, A. Bachrach, S. Hemachandra, E. Brunskill, A. Banerjee, D. Roy, S. Teller, *et al.*, "Generalized grounding graphs: A probabilistic framework for understanding grounded language," *Journal of Artificial Intelligence Research*, 2013.

[22] P. J. Ramadge and W. M. Wonham, "The control of discrete event systems," *Proceedings of the IEEE*, vol. 77, no. 1, pp. 81–98, 1989.

[23] P. Hubbard and P. E. Caines, "Dynamical consistency in hierarchical supervisory control," *IEEE Transactions on Automatic Control*, vol. 47, no. 1, pp. 37–52, 2002.

[24] C. G. Cassandras and S. Lafortune, *Introduction to discrete event systems*. Springer Science & Business Media, 2009.

[25] D. Chen and C. Manning, "A fast and accurate dependency parser using neural networks," in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 2014, pp. 740–750.

[26] S. Zheng, S. Jayasumana, B. Romera-Paredes, V. Vineet, Z. Su, D. Du, C. Huang, and P. H. Torr, "Conditional random fields as recurrent neural networks," in *Proc. of IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 1529–1537.

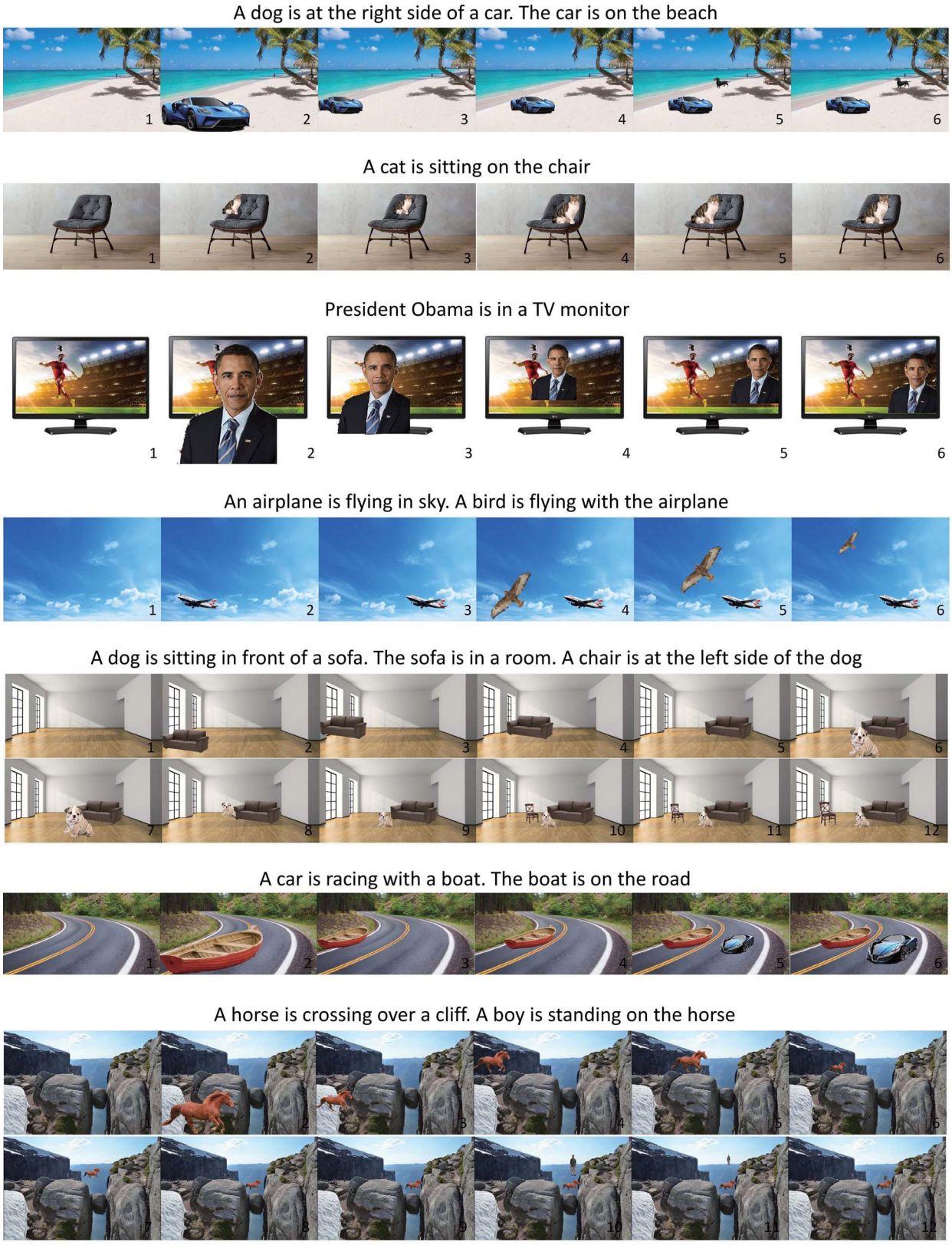

Fig. 4. More examples of generated scenes using the proposed image synthesis approach. Novel scenes that are highly unlikely to happen in real life can also be synthesized.