



AR Paint: A Fusion System of a Paint Tool and AR

Jihye Hong, J. Ryu, Hyun Yang, Suwon Lee, Jinki Jung

► **To cite this version:**

Jihye Hong, J. Ryu, Hyun Yang, Suwon Lee, Jinki Jung. AR Paint: A Fusion System of a Paint Tool and AR. Gerhard Goos; Juris Hartmanis; Jan van Leeuwen. 11th International Conference on Entertainment Computing (ICEC), Sep 2012, Bremen, Germany. Springer, Lecture Notes in Computer Science, LNCS-7522, pp.122-129, 2012, Entertainment Computing - ICEC 2012.

HAL Id: hal-01556155

<https://hal.inria.fr/hal-01556155>

Submitted on 4 Jul 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

AR Paint: A fusion system of a paint tool and AR

Suwon Lee, Jinki Jung, Jihye Hong, JB Ryu, Hyun S. Yang

Dept. of Computer Science, KAIST, Republic of Korea
{swlee, jk, jihyerish, jbmagic, hsyang}@paradise.kaist.ac.kr

Abstract. In this paper, we present AR Paint, a fusion system of a paint tool and Augmented Reality (AR) that can augment in-place hand-drawn paintings onto an object. The system allows users to draw anything using the paint tool and then augment it onto an object whenever they want while providing a new experience and enjoyment during the process. It also has potential value if used in industry or for medical purposes. The hand drawing and visualization for AR is done through the screen of a mobile device, and all processes of the system are performed in real time.

Keywords: Augmented Reality, AR, AR Paint

1 Introduction

Have you ever desired to draw something directly onto a famous painting? When you visit an art gallery, you may wonder how a painting would look if something you drew could be added to the painting. For example, you can imagine drawing Mona Lisa's eyebrows onto the painting Mona Lisa by Leonardo da Vinci. This impossibility has motivated us, and we made it possible using Augmented Reality.

Augmented Reality (AR) is a type of human-computer interaction that enhances our perception and helps us to see, hear, and feel our environments in new and enriching ways while providing local virtuality [1, 2]. With AR technology, we can gain a better understanding of education, find guidance through processes, be instructed in the assembly of objects, and have new experiences that we otherwise could not.

In this paper, we present a fusion system of a paint tool and AR termed AR Paint. The system allows users to draw anything using the paint tool and augment it onto an object in real time while providing a new experience and enjoyment during the process. The drawing and visualization for AR is done through the screen of a mobile device, and all processes of the system are performed in real time. The system also has the potential to be used in industry or for medical purposes. For example, surgeons may need to mark specific blood vessel for tracking while performing surgery. In such a situation, AR Paint can be the best choice when it supports full 3D object tracking with drawing in 3D space. In the AR Paint system, however, we currently consider only 2D surface tracking while drawing on a surface.

The rest of this paper is organized as follows. In Section 2, we first present the AR Paint system with its scenario and framework. We then show a demonstration with a

performance evaluation in Section 3. Finally, we conclude with a brief summary and discussion in Section 4.

2 AR Paint

In this section we present the AR Paint system from a scenario based on end users' viewpoints to detailed accounts of its processes. The system can take many different forms depending on its application, but in this paper, we describe our system via an AR Paint application for an art gallery for a clear understanding of its capabilities.

2.1 Scenario

In the Louvre Museum in Paris, a man is standing in front of Mona Lisa, Leonardo da Vinci's most famous painting. While looking at the painting, he wonders how different the painting would look if Mona Lisa was given eyebrows. He runs the AR Paint application on his smart phone and then he points its built-in camera at the painting. The boundary of the painting is marked and tracked in every frame. Once he touches a screen of the smart phone, the current frame is fixed and the GUI for the paint tool pops up. He starts drawing the eyebrows on the touch screen using his fingertip. Once he finishes this task, the eyebrows he drew are visualized and tracked in every frame.

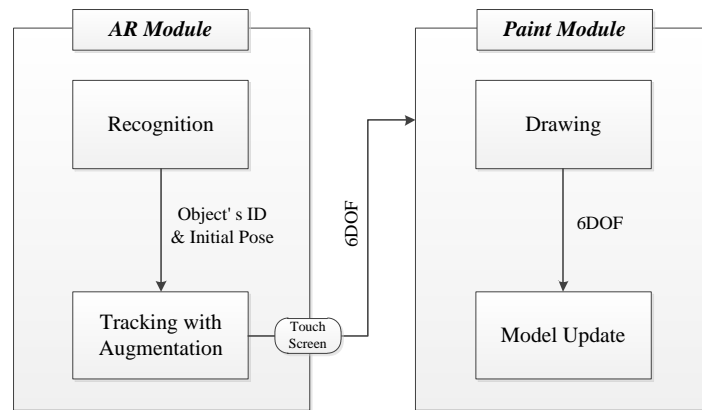


Fig. 1. The framework of AR Paint

2.2 Framework

For the AR Paint system, only a mobile device equipped with a camera and a supporting touch-screen input are necessary, both of which are available on the latest smart phones. AR Paint system is composed of two main modules: the AR module and the Paint module, as shown in Fig. 1.

In the AR module, recognition and tracking are performed using training data that is learned from recognizable objects. The recognizable objects and the distribution of the training data depend on the application. In our application, the recognizable objects are limited to 2D paintings and the distribution of the training data is done in advance when the application is installed on the mobile device.

The AR module switches to the Paint module when the user touches the screen while tracking an object. When switching, the three-dimensional position of the object is remembered for use later. Users can add something to the painting in the current fixed frame using the paint tool. The three-dimensional position is used to remove any distortion from the user's drawing and update the model which will be augmented in the AR module.

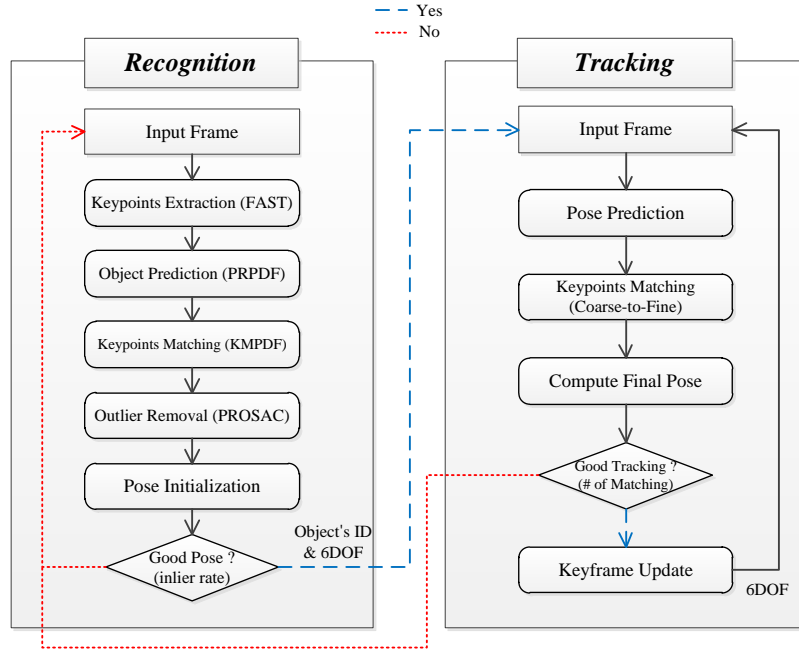


Fig. 2. Recognition and Tracking process in the AR module

2.3 AR Module (Recognition and Tracking)

Fig. 2 shows the entire process of the AR module, which is described in this section. The recognition process is initially performed for every frame captured by the camera. For real-time object recognition with pose estimation, we use a Generic Random Forest (GRF) [3]. The GRF is designed to be capable of simultaneous object recognition and keypoint matching through one pass over the original Random Forest [3, 4]. As shown in Fig. 3, a GRF consists of N_T random trees, and each tree conducts D binary tests comparing the intensities of two pixel locations of the input image patch surrounding a keypoint. Every leaf node of each tree stores $n + 1$ probability

distributions (1 distribution is for object recognition, and each of the n distributions is for the corresponding object's keypoint matching). Therefore, for an image patch surrounding a keypoint, the GRF returns two results ('where' is the keypoint from among n objects, and 'what' is the keypoint from among the k keypoints of the corresponding object). For every frame, several keypoints are extracted via a FAST corner detector [5]. Every keypoint is passed through the GRF and the candidate object is predicted with matching pairs. Outliers are eliminated by the computation of a homography using PROSAC [6], and only matching pairs with reprojection error amounts of less than 10 pixels become inliers. If the number of inliers is smaller than a threshold, the candidate object is rejected for recognition; otherwise, the initial six-degree-of-freedom (6 DOF) pose of the candidate object is calculated [7].

Once an object is recognized, its ID and 3D pose are estimated and the tracking process is started. During the tracking process, a new pose of the object in a new frame is predicted using the previous 3D pose and the current camera motions. For the predicted new pose, coarse-to-fine matching is conducted to calculate the final pose and the opportunity to update the keyframe is given only in the case of good tracking. All of the aforementioned tracking processes are performed in the same manner as [8]. While an object is tracked, the boundary of the object is marked with a red line whose inner area becomes the canvas in the Paint module. Moreover, if there are some drawings of the object, they have to be augmented onto the object. In order to complete these tasks simply, we put one virtual layer per object and refer to the virtual layer as a model. The model is initialized by a transparent canvas with a red boundary considering corresponding to the object's shape and size. As a result, we can easily facilitate both augmentation and model update (Fig. 5).

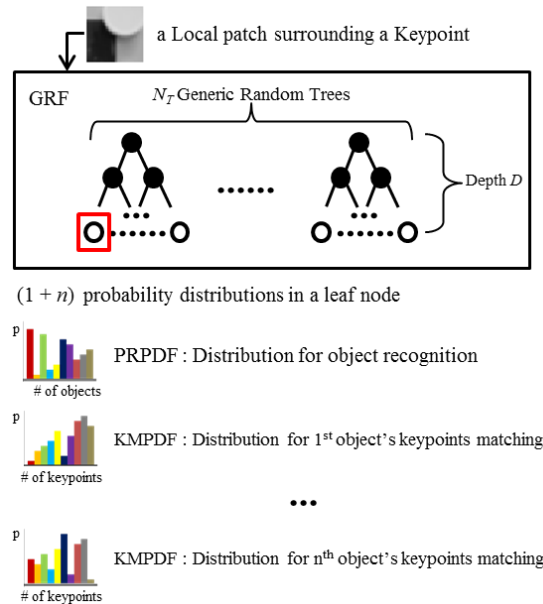


Fig. 3. Generic Random Forest



Fig. 4. Ready for drawing (fixed frame with the GUI of the paint tool)

2.4 Paint Module (Drawing and Model Update)

When the system is tracking, the current frame has an augmented model on the object as its image. By touching the screen during the tracking process, the current frame is fixed and the GUI for paint tool is displayed on the screen, as shown in Fig. 4. This state is ready for drawing and the inner-area of the model's boundary becomes a canvas for drawing. Users can draw something on the touch screen using their fingertip or a touch pen. In our application, currently, drawing is limited to 2D space, as our recognizable objects are 2D paintings and because the drawing is added onto their surfaces.

Once the drawing task is finished, the system starts the tracking process again and the users can see their drawing become augmented. This means the model for the object is updated immediately. When users draw something onto an object, the 3D position of the object differs depending on the viewpoint of the users. Therefore, we need to remove distortion in the drawing before applying it to a model. As shown in Fig. 5, the drawing is inversely warped to remove its distortion using 6 DOF pose of the object and the model is then updated according to the result.

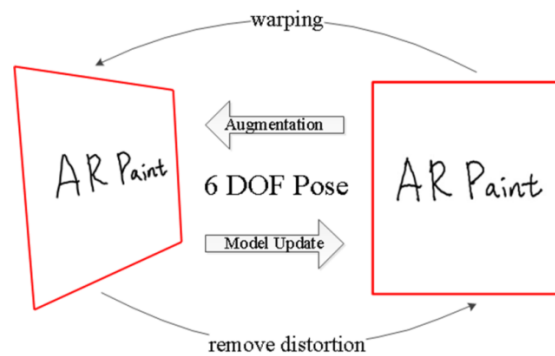


Fig. 5. Augmentation and Model Update

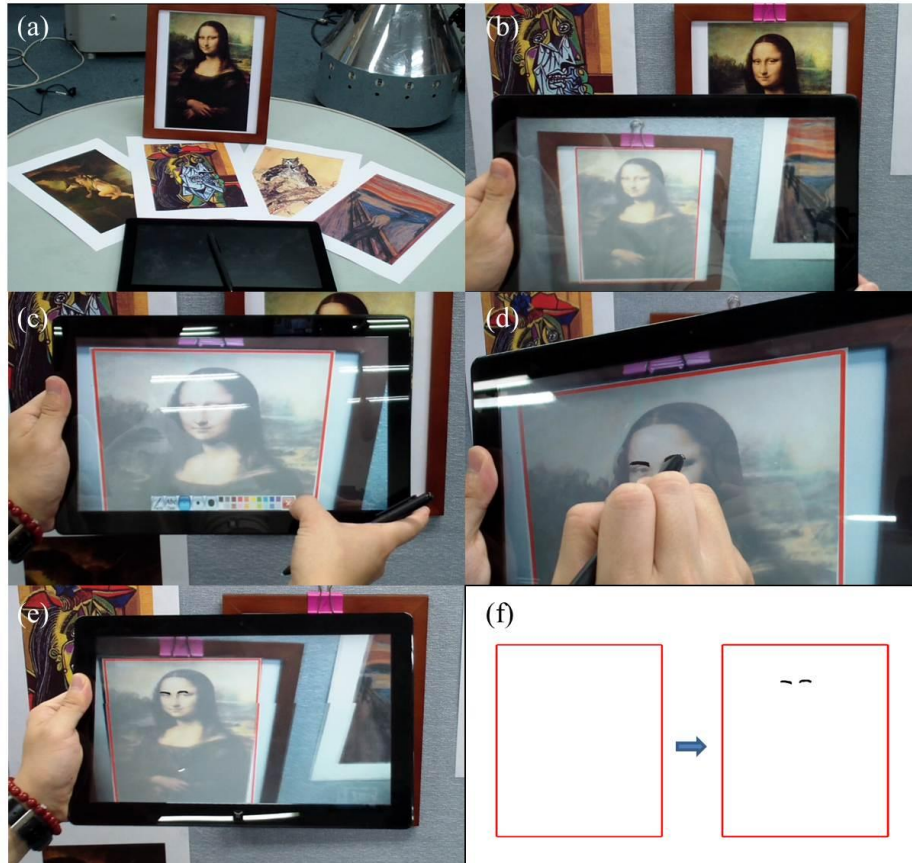


Fig. 6. Demonstration of AR Paint System :
 (a) Environments: Tablet PC and target objects (b) Tracking with model augmentation
 (c) GUI for drawing (d) Drawing on a touch screen
 (e) Updated model in next tracking (f) Model update

3 Demonstration

To test our system, we implemented the AR Paint application on a tablet PC with a 1.4GHz dual-core processor and 1GB of RAM. For target objects, we printed out 100 paintings and trained them using a GRF which consisted of 40 trees of 12 depths, as mentioned in Section 2.3. At runtime, the application works as described in Section 2.1. Fig. 6 shows the demonstration of the application. All processes of the application were performed in real time.

For an evaluation of the performance, we first measured the recognition rate and recognition time as the number of training objects increased from 20 to 100. We randomly synthesized 100 new images per object, and they were used to measure the recognition performances. The average recognition rate and recognition time are

shown in Table 1. We then measured the tracking time using a test movie with 500 frames. The test movie includes various events with tilts, scale variations, rotation, and fast movement. The averaged tracking time over all of the frames was 5.2ms. With the recognition time, it achieves more than 30 fps considering the total time requirements. This demonstrates that our system is fast enough in terms of its frame-rate performance.

# of training objects	20	40	60	80	100
Recognition Rate (%)	99.0	98.8	98.6	98.3	98.1
Recognition Time (ms)	18.9	19.2	19.6	20.2	21.0

Table 1. Recognition Rate and Recognition Time

4 Conclusion

In this paper, we presented AR Paint, a system which is able to augment in-place hand-drawn paintings onto an object. With the AR Paint system, users can immediately add their own drawing to a well-known painting, and the system provides a new experience and enjoyment for users as they use it. In the proposed AR Paint system, currently only 2D drawings on 2D surfaces are allowed. However, by allowing 3D drawings on 3D objects, it will become feasible to use this system for various purposes. As a future work, we first intend to find needs the AR Paint system can fulfill by conducting user studies. We also have plans to expand the system to support 3D drawings on 3D objects.

5 Acknowledgement

This work was supported by the IT R&D program of MKE/MCST/IITA, [Development of learner-participational and interactive 3D Virtual learning contents technology].

This research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education, Science and Technology (2011-0018291 / 2012-0005793)

References

1. Azuma, R., Bailiot, Y., Behringer, R., Feiner, S., Julier, S., MacIntyre, B.: Recent Advances in Augmented Reality. *IEEE Computer Graphics and Applications* 21(6), 34–47 (2001)
2. Krevelen, D.W.F., van Poelman, R.: A survey of augmented reality technologies, applications and limitations. *Int. J. of Virtual Reality* 9(2), 1–20 (2010)
3. Cho, K., Yoo, J., Yang, H.: Markerless Visual Tracking for Augmented Books. *Joint Virtual Reality Conference of EGVE-ICAT-EuroVR* 13–20 (2009)

4. Lepetit, V., Fua, P.: Keypoint recognition using randomized trees. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 28(9):1465–1479 (2006)
5. Rosten, E., Tom. D: Machine Learning for High-Speed Corner Detection. In Proc. 9th European Conference on Computer Vision 430–443 (2006)
6. Chum, O., Matas. J: Matching with PROSAC-progressive sample consensus. In Proc. IEEE Conference on Computer Vision and Pattern Recognition 220–226 (2005)
7. Schweighofer, G., Pinz, A.: Robust Pose Estimation from a Planar Target. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 28(12), 2024–2030 (2006)
8. Cho, K., Jung, J., Lee, S., Lim, S., Yang, H: Real-time recognition and tracking for augmented reality books. *Computer Animation and Virtual Worlds* 22(6):529–541 (2011)