Taking Pictures with Diminished Reality

Fernando Akio de Araujo Yamada
Computer Science Department
Universidade Federal de Juiz de Fora
Email: akio@ice.ufjf.br

Luciano Walenty Xavier Cejnog
Computer Science Department
Universidade Federal de Juiz de Fora
Email: luciano.wxc@gmail.com

Rodrigo Luis de Souza da Silva
Computer Science Department
Universidade Federal de Juiz de Fora
Email: rodrigoluis@ice.ufjf.br

***Abstract*—This work presents an approach for object removal in photos without a post-processing step, that is, at the moment the picture is taken, based on a marker guided selection of a region of the image and the application of an inpainting algorithm over this selection, in order to give a realistic look. Our results show that the proposal is feasible and promising, being capable of producing realistic pictures on successful cases, although its clear limitations.**

I. Introduction

Diminished Reality is the opposite of Augmented Reality, aiming to remove aspects of the reality around the observer. Object removal is a common diminished reality task, allowing a user to perceive that a desired object is not visually there anymore.

In this paper, we propose an application that allows the user to take a picture, selecting an area with a marker and removing the content behind the selected region without any post-processing, filling the region realistically. Such application can be useful when one desires to take a picture with a digital camera and remove something immediately, like an undesirable person on a picture, or a stain on someone's clothes, for example. This technique can also be used on optical head-mounted displays, which seem will soon become more popular.

Two main techniques are used, first the identification and tracking of a selection marker, and second the process of filling in the selected region, through an image completion method. Both are further detailed in next sections.

A. Related Works

Several types of methods for tracking hands or objects exists in the literature. These may include bare-hand tracking, like the system presented by Stenger et al. [1], that yields good results, at the cost of a non-interactive system. A real-time approach was made by Schlattman et al. [2], that requires a setup of three cameras. This setup is, however, unfeasible in our application. The region selection step of our work relates to Mistry et al.'s [3], [4], since it is based on a camera tracking some color markers.

As for the image completion step, it consists in filling a missing region of an image in a way an observer does not perceive the change. It is a well explored problem and the literature has some good approaches. Bertalmio's inpainting [5] was the first solution for image completion problem. This approach used the concept of continuity of the isophote lines and a diffusion process to fill in missing areas on an image. This approach yielded good results on line completion, being feasible for applications like restoration of old pictures with small corrupted lines. Telea et al [6] propose a technique based on a fast marching method, which achieves similar results in a much smaller time. The main limitation of both methods is their inability to deal with texture propagation, which becomes evident in larger regions.

Criminisi et al. [7] propose an algorithm to image completion using a patch-based technique, which uses texture synthesis and inpainting to enhance the quality of the output. Wexler et al [8] applied some concepts of patch-based methods to videos. It is an important step towards diminished reality, even though is was not their goal, attempting to hide an object in a video stream. The downside, on the other hand, is the time constraint, turning impossible to run in real-time. Barnes et al. [9] proposed the PatchMatch, also based on patch techniques, as the name suggests. One of its possible applications is image completion, which uses an approximated nearest-neighbor field to estimate the correspondences between the missing area and patches on the rest of the image. Herling et al. [10] propose an improvement on the PatchMatch, to achieve real-time capabilities on image completion on video. Although a good approach, it is focused on a recorded video, being a different problem than ours. We use the PatchMatch, due to its good capabilities as an image completion algorithm.

II. Proposed method

An intuitive way to point out something is using our fingers, our system is based on this concept. When one wishes to take a picture and there is something undesirable on the scene, he can point it out, select its surroundings and save the picture without this region in a realistic way. To make things simpler, the selection of the region is made using a colorful object, such a way it highlights from the background. This is a simpler and faster approach than, for example, hand detection techniques, and the marker object can be customizable.

The proposed system is then based on two main techniques: marker tracking and filling the selected region using an image completion algorithm. For the rest of this paper, we consider a picture as a frame extracted from a video camera stream.

A. Object Selection

The selection is made using any colorful object, as previously stated, whose color is known in the HSV spectrum. This color can be obtained in an optional calibration step, where the user selects a region containing the marker and it automatically determines its HSV values. HSV color system is suitable, since the Hue channel is less sensitive to brightness change, a

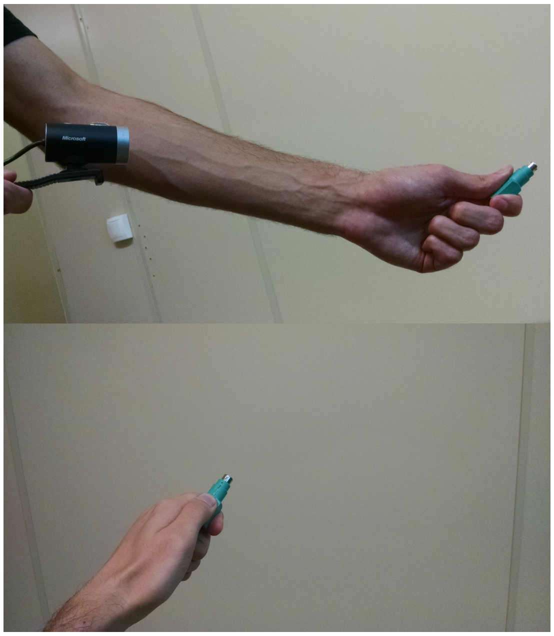

Fig. 1: The setup of the camera and the marker.

common problem in computer vision applications. This way, every frame is converted to HSV and is filtered, removing pixels outside of a threshold interval t. The resulting pixels are binarized, that is, pixels whose color is close to the object are considered correct pixels and have value 1, otherwise, their value is 0. This procedure is equivalent to apply a band-pass filter.

The binary image can still have noisy points, due to camera refresh rate, which makes the selection process unstable. In order to reduce the noise on the process, and therefore, improve the stability of the detected object, a morphology process is applied, dilating twice and then eroding the image.

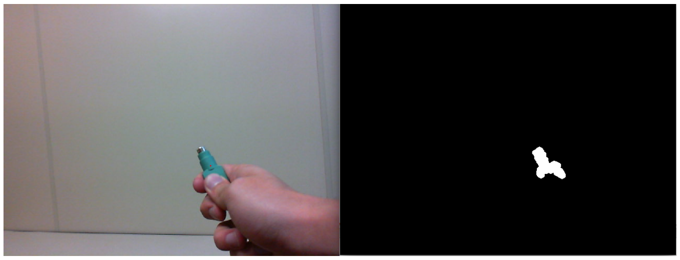

Fig. 2: Original frame and binary frame.

The selection point is set as the centroid of the correct pixels, so it is subject of variations of number of correct pixels, i.e., when the number is too small, its position will be too unstable. To reduce this instability, the selection point at a given frame c_i is a weighting of the previous point c_{i-1} and the current centroid of correct pixels.

A start button and a finish button control when the selection begins and ends, respectively. Since the method is based on tracking a marker, moving the camera when the marker is seen will cause an apparent movement of the marker, thus producing a rough selection area. This way, the user must be careful when using the system.

B. PatchMatch

The inpainting technique used on this work is based on PatchMatch [9] structural image editing technique. This technique is based on the estimation of an approximate nearest neighbor field, which contains a coarse estimative of correspondence between pixels of two sets. The improvement over previous techniques is the change of the search space: instead of pixels, the method searches for offsets that applied to pixels lead to good correspondences.

The Nearest-Neighbor estimation initializes the offsets randomly and looks in its neighborhood for offset values which minimizes the difference between the pixel and its correspondent patch, propagating good matches. This is combined to a random search step, which looks into random patches on a fixed radius for good offsets, aiming to make guesses that, if good, can be propagated on the following iterations.

When the algorithm is used to image completion the search domain is the original image, and it aims to minimize a distance functional containing a completeness term, which ensures that information from the input is not lost on the process, and a coherence term, which penalizes the creation of artifacts. We use the algorithm to fill the selected region with patches from the rest of the image. Since the PatchMatch tends to create a smooth filling, the result is usually realistic.

The minimization of this functional is given by an Expectation-Maximization algorithm, and in order to avoid local minima, the process starts in a lower resolution and is gradually upsampled until the dimensions of the image are reached. The computational complexity of the process is proportional to the size of the selected region and to the number of scales used.

III. Results

In our experiments we used a green marker, shown in Figure 1 and Figure 2, since there was not any other green object in the scene. The HSV interval threshold used for this object is: Hue: $[51, 91]$, Saturation: $[103, 183]$, Value: $[100, 180]$. The size of the morphology mask used in the dilation and erosion step is 7×7, using a cross shaped structuring element.

We performed the removal of objects and imperfections on five different pictures. Over a linoleum floor, an asphalted ground, a blue and a white table and a flat floor. From these, only the white table presented flaw cases, which was a harder scenario, containing many objects.

The first case, on Figure 3, the mug was selected to be removed. The inpaiting was acceptable, tricking an unaware observer, but some imperfections on the joints of the wooden floor are visible. The edges of the selected region are clearly non-smooth. This is due to false-positives in the selection step, possibly caused by the shirt of the stuffed tiger. The second example, also on Figure 3, the keyring was selected. The result is very good, as expected from the PatchMatch over an irregular texture. Again the edges of the selected region were non-smooth, this time affected by the leaves.

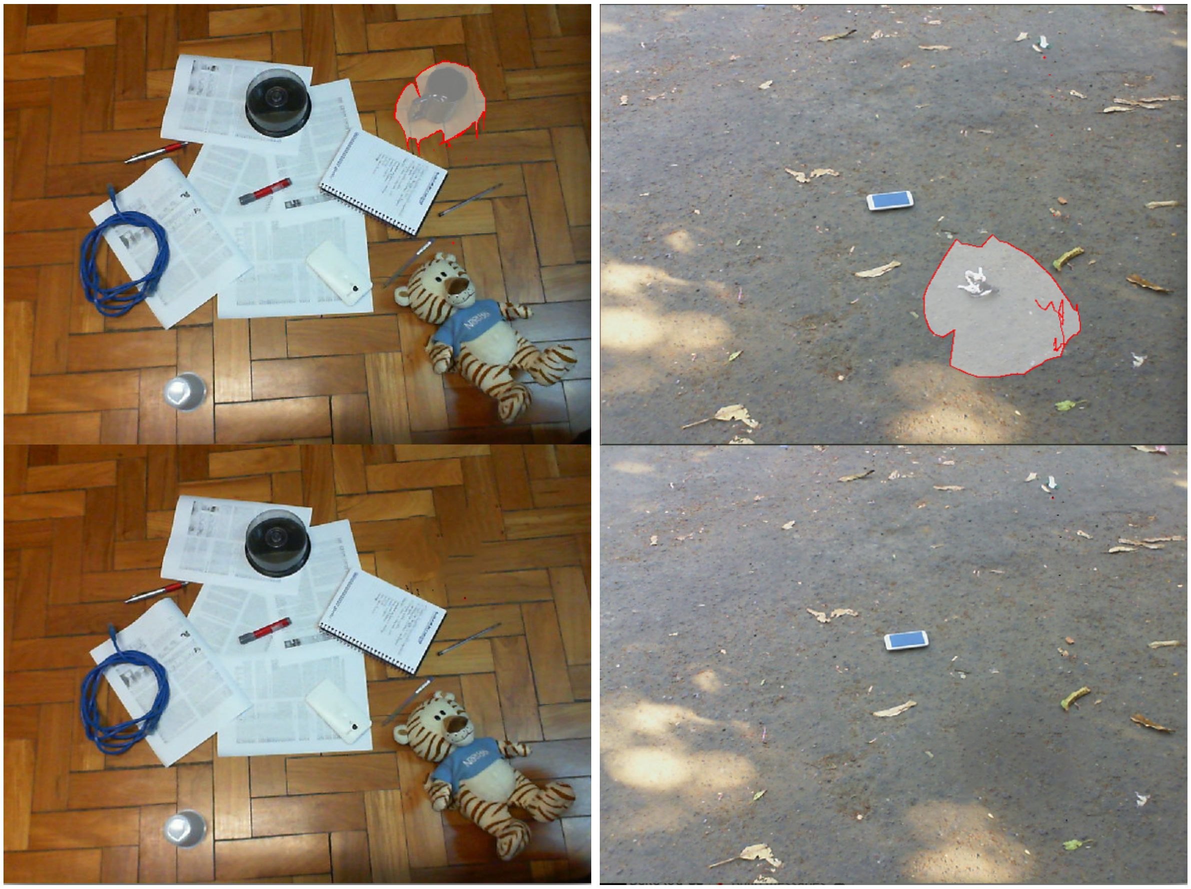

Fig. 3: Selection and removal of the mug and the keys.

On Figure 4, imperfections on the table and the floor were hidden, using our method. Since there were not any other object whose color is similar to our marker, the selected regions were smoother than on Figure 3. The inpainting here was also very good.

Figure 5 shows two flaw cases. The first, the pen in the center was selected, but the PatchMatch filled the region in with only part of the bottom pen. In the second, the DVD case was selected to be removed, note how imprecise the selection was, again because of the blue shirt of the stuffed tiger. The PatchMatch successfully copied the net cable. We consider this a flaw case, since in our purpose not every object copy is capable of making the picture realistic.

All the above experiments were made using a *Microsoft LifeCam Cinema*. The time spent during the inpainting step were about $20 \sim 25$ seconds, using an AMD A8-2500M notebook, with 6GB DDR3 1600MHz. This time constraint limits the real-time usage of the method. The whole project was implemented using the C++ language.

IV. Conclusion and Future Works

The proposed application is capable of, at the moment one takes a picture, removing an area selected with the use of a marker, and filling this region in a realistic way, through the use of an inpainting algorithm. Results show that the inpainted regions are good, but the method still have difficulties in filling in some regions, specially if the area of the selected region is big, with respect to the dimensions of the picture. Another issue noted is the time to compute the inpainting. Although feasible, it benefits from faster inpainting methods that produces good results, like some state of the art algorithms.

The main flaw of the proposed application is the need of a colorful marker. Although computationally cheap to track, it may become an issue, due to the need of a unique color restriction. Another issue is the need to hold still the camera, in order to produce smoother selection regions. Results show the effect of having an object of similar color: the selection point becomes too unstable, and the edges of the selected region are not smooth. A finger detection technique would be much more suitable to the proposed application, at the cost of an extra processing time. Some gestures could also be used to control the beginning and the end of the selected region, introducing some natural interface concepts.

The natural extension of this application would be an implementation to mobile devices, since nowadays portable gadgets with a digital camera are a reality. The only restriction would be the computational cost of the inpainting algorithm, but since mobile devices are becoming faster, it soon will not be a problem anymore.

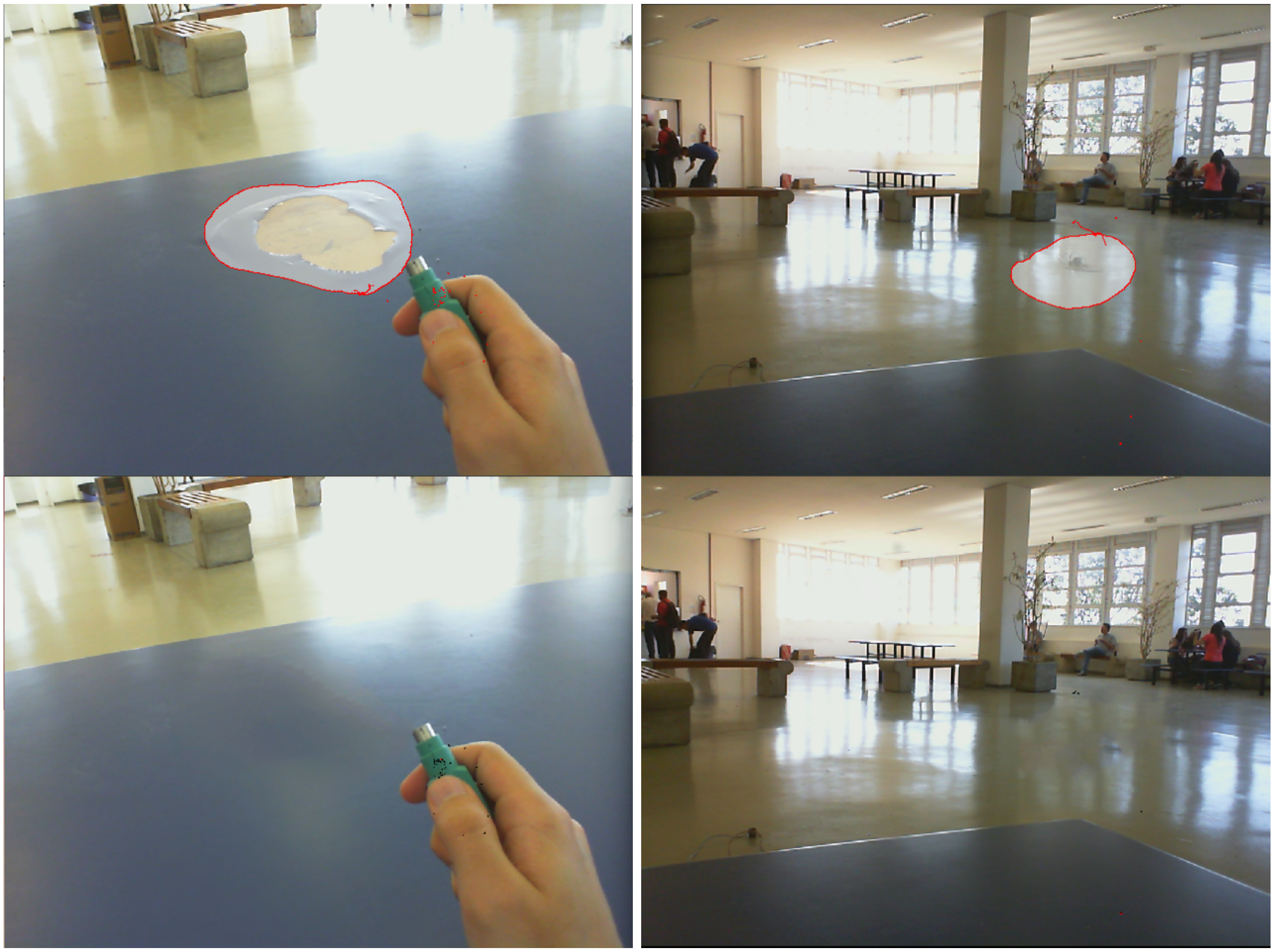

Fig. 4: Removal of imperfections on a table and the floor.

Acknowledgment

The authors would like to thank FAPEMIG and CAPES for the financial support.

References

[1] B. Stenger, A. Thayananthan, P. H. Torr, and R. Cipolla, "Model-based hand tracking using a hierarchical bayesian filter," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 28, no. 9, pp. 1372–1384, 2006.

[2] M. Schlattman and R. Klein, "Simultaneous 4 gestures 6 dof real-time two-hand tracking without any markers," in *Proceedings of the 2007 ACM symposium on Virtual reality software and technology*. ACM, 2007, pp. 39–42.

[3] P. Mistry and P. Maes, "Sixthsense: a wearable gestural interface," in *ACM SIGGRAPH ASIA 2009 Sketches*. ACM, 2009, p. 11.

[4] P. Mistry, P. Maes, and L. Chang, "Wuw-wear ur world: a wearable gestural interface," in *CHI'09 extended abstracts on Human factors in computing systems*. ACM, 2009, pp. 4111–4116.

[5] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester, "Image inpainting," in *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*. ACM Press/Addison-Wesley Publishing Co., 2000, pp. 417–424.

[6] A. Telea, "An image inpainting technique based on the fast marching method," *Journal of graphics tools*, vol. 9, no. 1, pp. 23–34, 2004.

[7] A. Criminisi, P. Pérez, and K. Toyama, "Region filling and object removal by exemplar-based image inpainting," *Image Processing, IEEE Transactions on*, vol. 13, no. 9, pp. 1200–1212, 2004.

[8] Y. Wexler, E. Shechtman, and M. Irani, "Space-time completion of video," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 29, no. 3, pp. 463–476, 2007.

[9] C. Barnes, E. Shechtman, A. Finkelstein, and D. Goldman, "Patchmatch: A randomized correspondence algorithm for structural image editing," *ACM Transactions on Graphics-TOG*, vol. 28, no. 3, p. 24, 2009.

[10] J. Herling and W. Broll, "Advanced self-contained object removal for realizing real-time diminished reality in unconstrained environments," in *Mixed and Augmented Reality (ISMAR), 2010 9th IEEE International Symposium on*. IEEE, 2010, pp. 207–212.

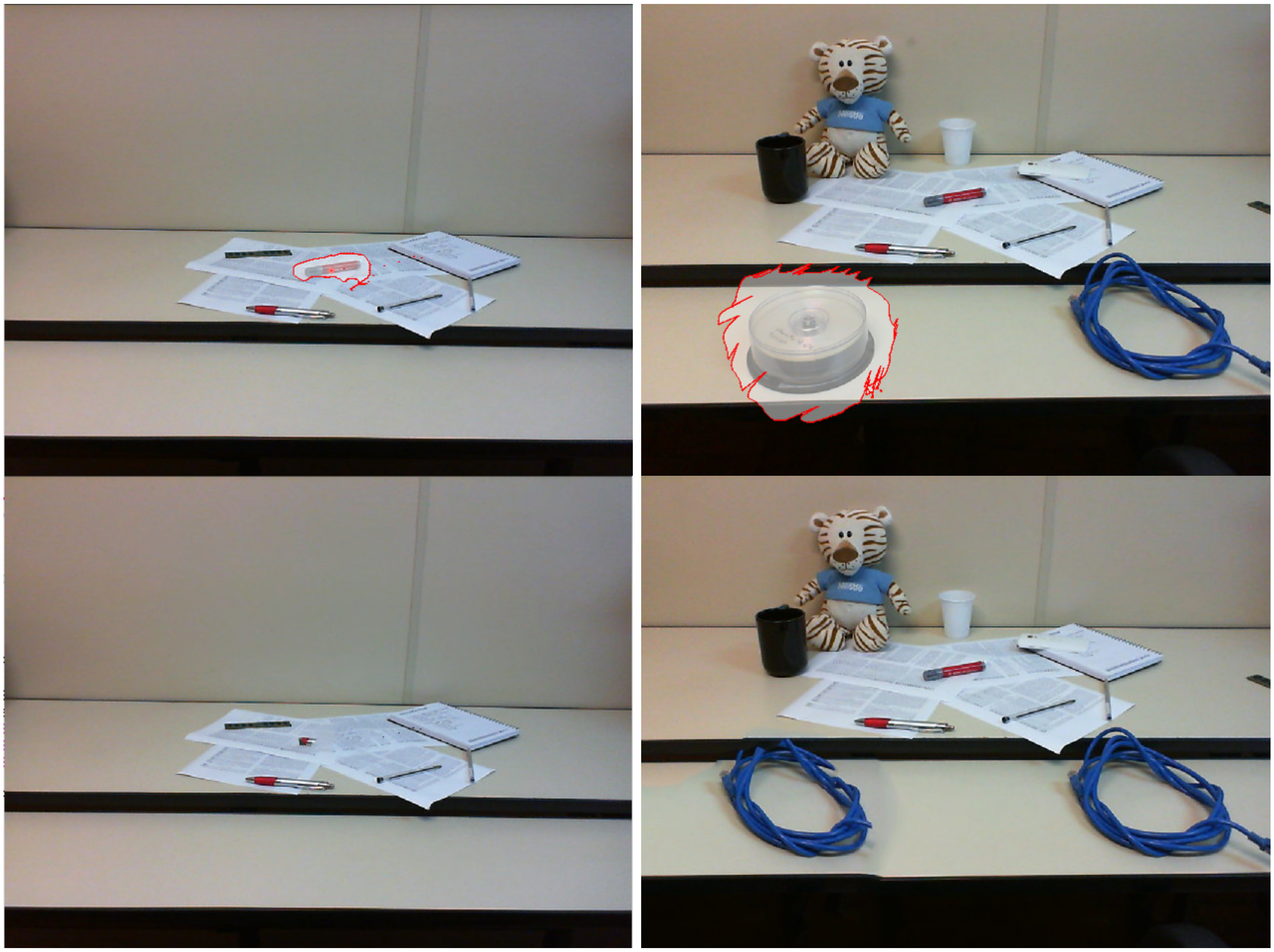

Fig. 5: Selection and removal of the pen and the DVD case. Both are flaw cases of our method.