[Poster] Local Optimization for Natural Feature Tracking Targets

Elias Tappeiner[1], Dieter Schmalstieg[2], Tobias Langlotz[3]

[1,2]Graz University of Technology

[3]University of Otago

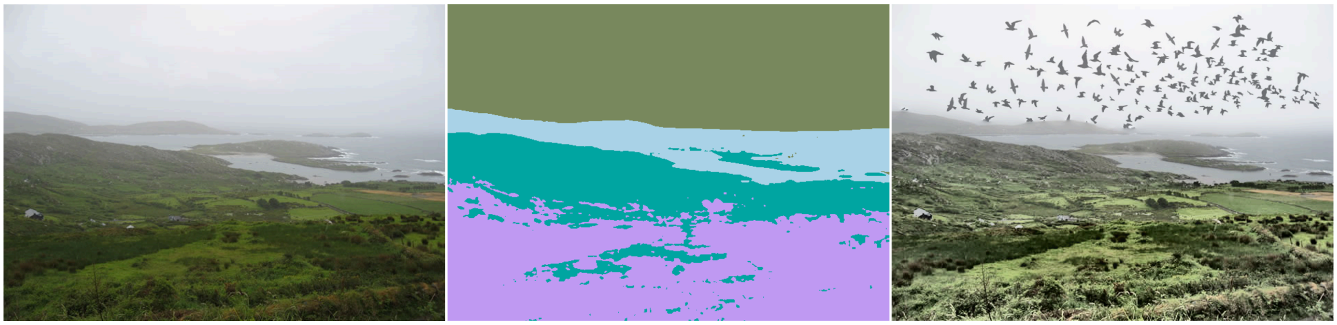

Figure 1: Example images from our target optimization pipeline. (Left) Original target intended for tracking using natural feature-based tracking, showing an unevenly distributed amount of features and a low contrast in some image areas. (Middle) Automatic image segmentation based on recursive Otsu segmentation applied on the input image. (Right) The tracking target after using our optimization pipeline. Local areas of the target are optimized for improved trackability (e.g., normalizing the image intensities or blending with image patterns in areas missing a minimum number of features). Edges between locally optimized images areas are seamlessly blended.

ABSTRACT

In this work, we present an approach for optimizing targets for natural feature-based pose tracking such as used in Augmented Reality applications. Our contribution is an approach fo r locally optimizing a given tracking target instead of applying global optimizations, such as proposed in the literature. The local optimization together with visualized trackability rating leads to a tool to create high quality tracking targets.

Keywords: Pose tracking, Natural features.

Index Terms: H.5.1 [Information Interfaces and Presentation]: Multimedia Information Systems – Artificial, augmented, and virtual realities; I.4.8 [Image Processing and Computer Vision]: Scene Analysis – Tracking

1 INTRODUCTION

Pose tracking based on tracking image features is nowadays widely used in Augmented Reality (AR) research as well as in commercial AR applications. In particular for the latter, preparing targets is still a challenging part when creating AR applications. Designers responsible for creating or selecting tracking targets (e.g. pages in magazines, company logos) often fail to understand the technical requirements for good tracking targets, such as having image features that a) exist in a sufficient number, b) are evenly distributed over the target, and c) have unique feature descriptors.

Existing commercial tracking solutions offer often a rating of the tracking target (expressed with marks). However, this rating is based on the number of image features but does not reveal issues with feature distribution or similar feature descriptors. Even worse, there is no feedback for assisting designers in solving these common problems when designing tracking targets for AR applications. Existing works from literature mostly present improved feature descriptors to tackle tracking issues but only a few works investigated the options for optimizing the tracking target. Gruber et al. demonstrated a system for analyzing tracking targets for their trackability and later also investigated the influence of the several texture characteristics in the trackability [3]. In their work the targets are altered globally which heavily affects the perceived visual quality of the used tracking target. However, local adjustments of critical image areas are only conceptually presented but not investigated in detail.

[1] e.tappeiner@student.tugraz.at
[2] schmalstieg@icg.tugraz.at
[3] tobias.langlotz@otago.a.c.nz

2 APPROACH

Our approach for locally improving tracking targets is based on three main parts: Firstly, an analysis of the tracking target to determine an overall rating of the target. Secondly, different image segmentation techniques are applied to identify image areas that are problematic for tracking. Finally, we optimize the identified image areas for better trackability and blend the visible artifacts at the edges of image areas that could arise from the optimisation.

2.1 Tracking target analysis

The designer starts using our system by loading the image to track from. Our system analyses the image for trackability. For the analysis, we use an approach similar to the one presented by Gruber et al. [3], which creates a large number of novel camera views on the fly. Having a virtual camera with a position sampled on hemispheres with increasing radii creates these views. To gain more views, the parameterized hemisphere is translated relative to the tracking object.

Once created, we compute image features using SIFT and SURF (we used both to be more generic) and compute the pose of the virtual camera after matching features. This strategy of systematically sampling possible views and computing the camera pose helps us to compute an initial rating for the image. The rating expresses how well this image qualifies as a trackable target. The main factors for our rating are: Re-projection error of all matched features as well as inliers (after RANSAC), inlier to outlier ratio, and the L^2 error of the computed pose to the ground truth over all views.

2.2 Segmentation

The analysis of the features within the target provides a quality rating for the whole target. In the next step, we segment the image to identify regions of the image that are critical for the trackability. During empirical tests with different targets we identified several classes of tracking targets that worked best with different segmentation techniques. We consequently integrated a set of segmentation techniques that offer different levels of granularity and run on the original input image or on the saliency map of the original image. This gives the designer the option to treat salient regions differently (e.g. applying only modest changes) compared to non-salient areas. For low granularity, we compute four local image segments using recursive Otsu foreground background segmentation [4] on the input image as well as on the saliency map [2] of the input image (see Figure 1, Middle). For high granularity, we use a superpixel over-segmentation instead of previously Otsu segmentation (Slic superpixel [1]) and apply it again on the input image and the saliency map. We compute these segmentations in parallel and show the designer the four resulting segmentations (high vs. low granularity and original image vs. saliency map based segmentation). We further indicate for each segment if it has a sufficient number of features ('OK'), or in case there are not enough features, if it can be fixed by contrast adjustment ('Needs contrast adjustment') or, in case there is only a uniform segment with no features at all (based on the standard deviation of pixel values in this segment), we recommend 'Needs artificial features'. Based on these recommendations, the designer can choose the most appropriate image segmentation and, based on the highlighted issues in each image segment, plan the next optimization steps. Furthermore, the interface for our approach does not require the designer to adjust parameters but instead shows the result of applying different strategies. The designer can choose the best one to continue to work with.

2.3 Local image optimization

The last step within our optimization pipeline focuses on giving the designer the option to fix the identified issues with local image segments to consequently improve the overall trackability. The designer should get an immediate response according to her selection of the segmentation, so we kept the optimization basic and fast. This makes it possible to use our implementation actively during the design process.

In case the previous analysis revealed no issues with a segment, the designer can continue with the other segments. However, some segments lack a sufficient amount of features due to a lack of contrast or caused by uniform colours (indicated by standard deviation of pixel values in this segment). Gruber et al. showed that a global contrast adjustment raises the texturedness of the feature set and therefore the trackability of the tracking target [3]. The intension of our work is to keep the target as unchanged as possible, so we adjust the contrast only locally to the critical segments. We provide two optimization techniques, basic histogram equalization and *Contrast Limited Adaptive Histogram Equalization* (CLAHE) [5]. Both are applied on the luminance component of the YCbCr color space. They are capable of revealing hidden structures and improve the contrast, leading to a more unique feature points and a better trackability. However, CLAHE limits the amplification by clipping the histogram before the equalization, which reduces the noise and only adjusts the contrast to a suitable level. When compared to basic histogram equalization this leads to generally better results in terms of visual quality. Locally optimizing image segments can lead to artifacts at the edges of image segments after the optimization. We overcome this issue by smoothing the transitions using Laplace blending between the optimised segment and the remaining segments (see Figure 1, Right). Contrast adjustment methods can lead to more feature points in a local segment, but only if there is already some existing structure. For a segment that is lacking sufficient enough information, we use texture blending to add new structures. We created a large texture database containing textures of all kind that are characterised by having a high trackability. By comparing the histograms of the segment and the histogram of textures in the texture database, we find the best fitting texture. Still, we give the designer the option to manually select the texture, as this step is (especially for large segments) essential for the end result (see the birds texture introducing features to the otherwise featureless sky, Figure 1, Right). Segments completely lacking features are common when brand labels/icons ought to be tracked or when large areas of the image depict the sky. In all those cases, the segments have hard transitions. Therefore, we merge this class of segments into the target without Laplace blending after the optimization.

3 Conclusion and Future Work

We presented a system that allows designers having no technical background to semi-automatically and locally optimize tracking targets when building commercial AR applications. Our system automatically analyses the selected image, computes local image areas through image segmentation and improves those segments showing common problems such as a lack of features, uneven distribution of features or a high similarity of image features. Finally, we seamlessly blend the optimized image segments to produce the final image target. This allows us to keep the design idea of the image target used for tracking as well as maintain a high image quality. Future work will target the integration into commonly used productivity tools such as Adobe Illustrator or Photoshop.

References

[1] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Susstrunk. Slic superpixels. EPFL Technical Report no. 149300, 2010.

[2] M. Cheng, G. Zhang, N.J. Mitra, Huang X., and Shi-Min Hu. Global contrast based salient region detection. pp. 409–416, 2011.

[3] L. Gruber, S. Zollmann, D. Wagner, D. Schmalstieg, and T. Hollerer, "Optimization of Target Objects for Natural Feature Tracking," in 2010 20th International Conference on Pattern Recognition, 2010, pp. 3607–3610.

[4] N. Otsu A threshold selection method from gray-level histograms. IEEE Transactions on Systems, Man and Cybernetics no. 9, pp. 62–66, 1979

[5] S. M. Pizer, E. P. Amburn, and J. D. Austin. Adaptive histogram equalization and its variations. Computer Vision, Graphics, and Image Processing no. 39, pages 355–368, 1987.