Finding Distractors In Images

Ohad Fried[1], Eli Shechtman[2], Dan B Goldman[2], Adam Finkelstein[1]
[1]Princeton University. [2]Adobe Research.

Taking pictures is easy, but editing them is not. Professional photographers expend great care and effort to compose aesthetically-pleasing, high-impact imagery. Image editing software like Adobe Photoshop empowers photographers to achieve this impact by manipulating pictures with tremendous control and flexibility – allowing them to carefully post-process good photos and turn them into great photos. However, for most casual photographers this effort is neither possible nor warranted. Last year Facebook reported that people were uploading photos at an average rate of 4,000 images *per second*. The overwhelming majority of these pictures are casual – they effectively chronicle a moment, but without much work on the part of the photographer. Such cases may benefit from semi- or fully-automatic enhancement methods.

Features like "Enhance" in Apple's iPhoto or "Auto Tone" in Photoshop supply one-click image enhancement, but they mainly manipulate *global* properties such as exposure and tone. Likewise, Instagram allows novices to quickly and easily apply eye-catching filters to their images. Although they have some more localized effects like edge darkening, they apply the same recipe to any image. However, local, image-specific enhancements like removing distracting areas are not handled well by automatic methods. There are many examples of such *distractors* – trash on the ground, the backs of tourists visiting a monument, a car driven partially out of frame, etc. Removing distractors demands a time-consuming editing session in which the user manually selects the target area and then applies features like iPhoto's "Retouch Tool" or Photoshop's "Content Aware Fill" to swap that area with pixels copied from elsewhere in the image.

In this work we take the first steps towards semi-automatic distractor removal from images. The main challenge towards achieving this goal is to automatically identify what types of image regions a person might want to remove, and to detect such regions in arbitrary images. To address this challenge we conduct several studies in which people mark distracting regions in a large collection of images, and then we use this dataset to train a model based on image features.

We created two datasets with complementary properties (Figure 1). The first consists of user annotations gathered via Amazon Mechanical Turk. The second includes real-world use cases gathered via a dedicated mobile app. The Mechanical Turk dataset is freely available, including all annotations. Figure 2 shows one of the analyses we performed on the datasets: annotating the different distractors to understand which types of objects are usually unwanted in the final composition.

Armed with the dataset, we calculate various image features and train a distractor classifier. Figure 3 shows some of our classification results. The ability to create distractor maps allows us to semi-automatically remove image distractors, as well as to perform distractor-aware image retargeting.

Our main contributions are: (1) defining a new task called "distractor prediction", (2) collecting a large-scale database with annotations of distractors, (3) training a prediction model that can produce distractor maps for arbitrary images, and (4) using our prediction model to automatically remove distractors from images.

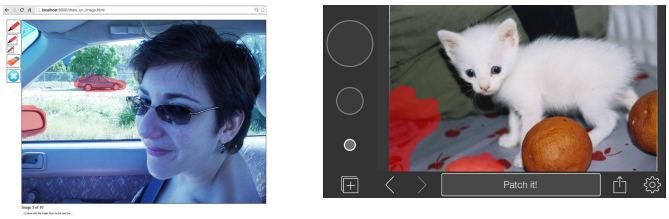

Figure 1: Data collection interfaces. Left: MTurk interface with basic tools for marking and removal. Right: Mobile app using inpainting with a variable brush size, zoom level and undo/redos.

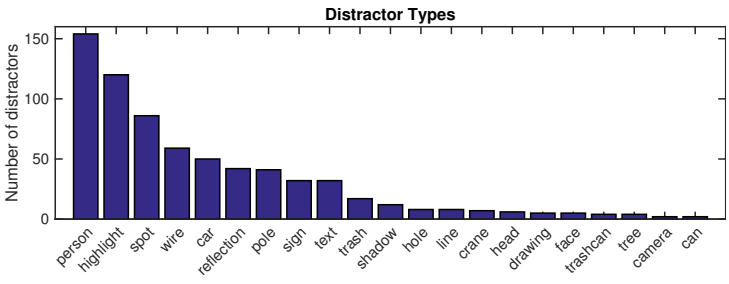

Figure 2: Distractor Types. We manually annotated all distractors to learn which elements people typically remove from photographs (plot shows nameable categories only).

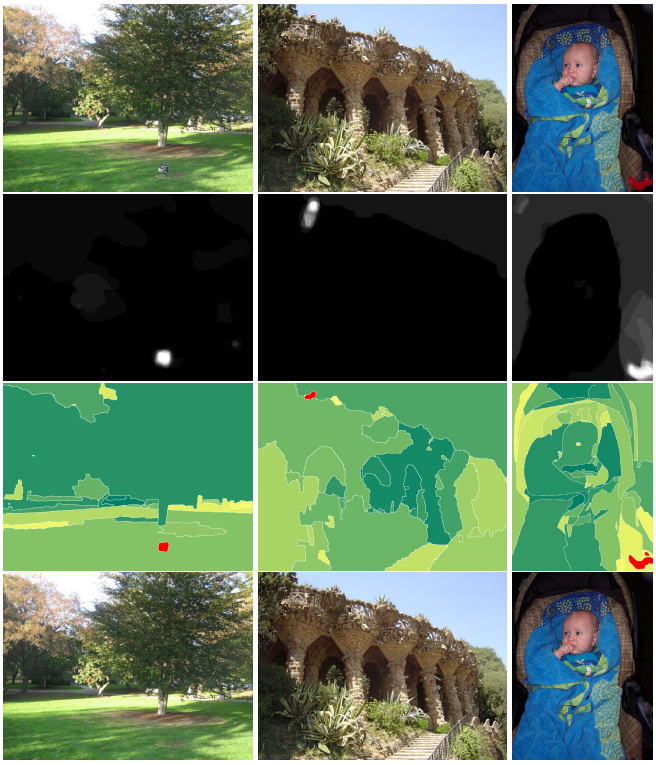

Figure 3: Examples of distractor removal. Each quadruplet shows (from top to bottom): (1) Original image. (2) Normalized average ground-truth annotation. (3) Order of segments as predicted by our algorithm. (4) Distractor removal result. Segment selected for removal are shown in red. Notice how these correlate with the ground-truth annotation. We manage to detect a variety of distracting elements (a sign, a person, etc.)

This is an extended abstract. The full paper is available at the Computer Vision Foundation webpage.