Interactive Pinpoint Image Object Removal

Frank Nielsen
Sony Computer Science Laboratories, Inc.
Frank.Nielsen@acm.org

Richard Nock
UAG France
Richard.Nock@martinique.univ-ag.fr

Abstract

We present a novel interactive system and its user interface for removing objects in digital pictures. Our system consists of two components: (i) (partially supervised/automatic) image segmentation [2], and (ii) (guided) texture synthesis [3].

1. User Interface and Algorithms

The user starts by loading a picture and moves the mouse over the image (Fig. 1-3a). Whenever the mouse moves, the segmented area implicitly selected by the cursor position is highlighted (Fig. 1-3b). Computing segmentations with or without bias is done in almost linear-time [2]. Once an area is removed by the user, a fast hole-filling texture synthesis procedure is triggered [3] (Fig. 1c, 2d and 3c).

Our system has three operation modes:

(1) For removing the implicitly selected area Fig 1.(1b), the user clicks the mouse left button. The system computes a bounding box $B$ around the removed area (after a few image morphology dilatations of the segmented area), expand it by a factor $\lambda$ and initialize texture synthesis using the remaining pixels in $(1 + \lambda)B$ (Fig 1.(1c)).

(2) Sometimes, we better fill the removed area by other similar parts of the image. We scribble by moving the mouse pushing the right button to indicate the portion of the image to use to initialize the texture synthesis (Fig 1.(2c) ). We define the selected pixels by taking the bounding box of the stroke or a fat stroke.

(3) Automatic segmentation does not often guarantee the perfect object recognition. In case of failure, the user presses the SHIFT key and the mouse left (inside) and right (outside) clicks to indicate prior cues (Fig 1.3b)). The segmentation algorithm [2] then efficiently uses those user-input bias to catch the desired object. Texture synthesis is either initialized from an enlarged bounding box, or a user stroke.

2. Results

We implemented in C++ those image editing procedures. Although our region-growing segmentation algorithm does not provide accurate object boundaries, it is fast and often enough for object removal applications. Bias is input as a few inside/outside points and not by strokes, as in [1]. It is particularly advantageous for objects with many contours, since "intelligent scissors" assume in their UI that objects have a single outer contour. We are currently investigating further extensions to our system: pulling out object mattes from trimaps obtained by our segmentation, trading between texture synthesis and image inpainting depending on the geometric characteristics of the removed region.

References