Feature Harvesting for Tracking-by-Detection
ECCV’06 Demo Proposal

Mustafa Özuysal, Vincent Lepetit, François Fleuret, and Pascal Fua

Computer Vision Laboratory
École Polytechnique Fédérale de Lausanne (EPFL) 1015 Lausanne, Switzerland
{mustafa.oezuysal, vincent.lepetit, francois.fleuret, pascal.fua}@epfl.ch
http://cvlab.epfl.ch

We have developed a fast approach for detecting a single object in 3-D together with the associated camera pose. The method requires a training phase that uses a video sequence to learn the appearance of patches centered on detected feature points. The training is fully automated apart from providing a rough 3-model of the target object and initial pose for the first frame.

Once training is completed, the object can be detected at each frame independently, hence it can initialize automatically and recover from tracking failures. Since a feature based approach is used it is also robust to partial occlusions. When the target is detected in a previous frame, we also use frame to frame matching at run-time to reduce jitter.

The method uses a Randomized-Tree based approach which uses simple binary tests to recognize image patches extracted around feature points. Therefore feature matching can be performed very fast at run-time. After feature matching a RANSAC based robust estimation step calculates the camera pose and if enough inliers are present the object is detected.

Fig. 1. Sample snapshot of our current user interface for object detection application. When the object is detected in the current frame 3-D features are projected on the image with the estimated camera projection matrix.
We want to demonstrate the run-time performance of the algorithm in a live demo. Several objects will be trained prior to the demonstration and the users will be able to test the tracking algorithm with these objects. A simple user interface like the one shown in Figure 1 will allow the users to change some parameters of the algorithm and observe their effects on the run-time performance. Our current implementation runs at 7-9 Hz on a desktop PC at 720x576 frame size and at 13-15 Hz when the frames are subsampled to 360x288.

We will also consider demonstrating the training phase for a simple object from a limited number of viewpoints. The limitation is due to time constraints for a live demo.

Sample videos demonstrating the run-time performance of the algorithm on three different objects can be downloaded from the following URLs: