Scalable Local Feature Extraction with Orientation Maps and GPU Computing

Naoyuki Ichimura
AIST, Japan
nic@ni.aist.go.jp

Local Invariant Features

The purpose of this research is to improve the scalability of feature extraction in terms of the number of features.

Orientation Maps

The voting values for the histograms of gradient orientations can be computed by applying multiscale Gaussian filters to 2D arrays corresponding to the bins of the histograms of gradient orientations, i.e., orientation maps.

This allows us to calculate the descriptors merely by looking up the locations of convolved orientation maps corresponding to the centers of cells. No exhaustive access to the pixels in cells is required thanks to orientation maps.

Performance Results

The test platform uses a 3.33GHz Intel Xeon X5680 CPU, an NVIDIA GTX580 graphics card, CUDA 4.0 and the Linux. Using orientation maps and the GPU enable us to improve the scalability of feature extraction in terms of the number of features as well as the efficiency of the computation compared to the CPU (25x-40x).

Tour de France

The changes in the computational times as a function of the number of grids on which features are extracted. Note the clear improvement of the scalability of feature extraction using the orientation maps.

Graffiti

Examples of orientation maps corresponding to the eight quantized gradient orientations. Each of the maps contains gradient magnitudes for a quantized orientation obtained from the edge image on the left.

Examples of multiresolutional edge images. Multiresolutional orientation maps are obtained by computing the orientation maps for each edge image.

We can take advantage of GPU computing in calculating the local descriptors using orientation maps:

Computing the voting values by multiscale Gaussian filters can be efficiently implemented by utilizing on-chip memory of a GPU.

The details of the computational times. The unit is microsecond.

Graffiti

The changes in the computational times as a function of the number of grids on which features are extracted. Note the clear improvement of the scalability of feature extraction using the orientation maps.

Calculating Local Descriptors

1. The histograms of gradient orientations are computed in cells arranged in a local region. The voting values for the histograms are the gradient magnitudes of the pixels in the cells.
2. A local descriptor is obtained by concatenating the histograms of the cells.

The purpose of this research is to improve the scalability of feature extraction in terms of the number of features.

The voting values for the histograms of gradient orientations can be computed by applying multiscale Gaussian filters to 2D arrays corresponding to the bins of the histograms of gradient orientations, i.e., orientation maps.

This allows us to calculate the descriptors merely by looking up the locations of convolved orientation maps corresponding to the centers of cells. No exhaustive access to the pixels in cells is required thanks to orientation maps.

The changes in the computational times as a function of the number of grids on which features are extracted. Note the clear improvement of the scalability of feature extraction using the orientation maps.

Calculating Local Descriptors

1. The histograms of gradient orientations are computed in cells arranged in a local region. The voting values for the histograms are the gradient magnitudes of the pixels in the cells.
2. A local descriptor is obtained by concatenating the histograms of the cells.

The purpose of this research is to improve the scalability of feature extraction in terms of the number of features.

The voting values for the histograms of gradient orientations can be computed by applying multiscale Gaussian filters to 2D arrays corresponding to the bins of the histograms of gradient orientations, i.e., orientation maps.

This allows us to calculate the descriptors merely by looking up the locations of convolved orientation maps corresponding to the centers of cells. No exhaustive access to the pixels in cells is required thanks to orientation maps.

The changes in the computational times as a function of the number of grids on which features are extracted. Note the clear improvement of the scalability of feature extraction using the orientation maps.