Fast Detection of Curved Edges at Low SNR

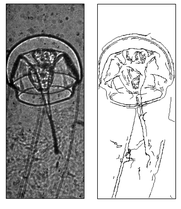
Nati Ofir Meirav Galun Boaz Nadler Ronen Basri

מכון ויצמן למדע
WEIZMANN INSTITUTE OF SCIENCE

IEEE 2016 Conf. on Computer Vision and Pattern Recognition
CVPR 2016

Objective:

- Efficient (near linear time) detection of faint curved edges in noisy images



Motivation:

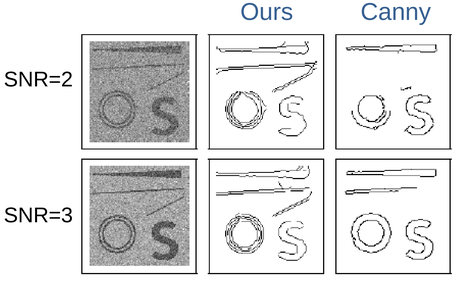
- Longer curves are perceived in lower SNR than short ones

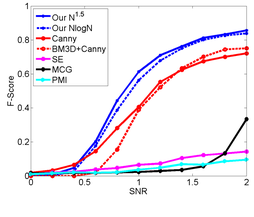
SNR = 1 | SNR = 2 | Clean Pattern

- Long is good: noise can be averaged out by smoothing along the curve (while maintaining contrast across the curve) by using a *matched filter*
- But where are those curves and what are their shapes? curves can appear in any of an exponential number of shapes

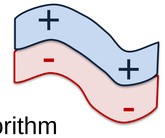
Our solution:

- An efficient hierarchical algorithm to examine an exponential number of candidate curved edges
- Use statistically rigorous adaptive threshold to detect edges at very low SNRs



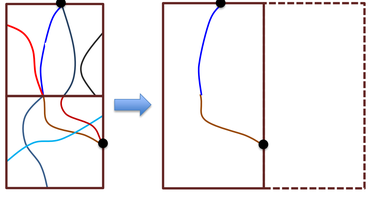


Approach:

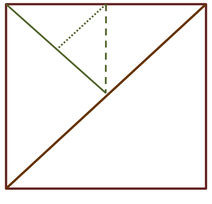


- Examine each potential edge curve using its "custom tailored" matched filter
- Do this efficiently using a dynamic programming-like algorithm on a hierarchical, binary-split tree of the image (keep best curve for each two points on the boundary of a tile)

Rectangle/square split



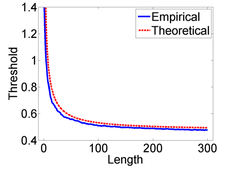
Triangle-split tree



Limits of detectability:

$T(L) = \sigma\sqrt{\frac{2\ln K_L}{wL}}$, where $K_L = 6N(2^{\beta L})$

(number of considered curves, K_L, grows exponentially with L)



$T(L \to \infty) = \Omega(\frac{\sigma}{\sqrt{w}})$

(Notation: N-image size; L, w-filter size; σ-noise level, β-constant)

Computational complexity:

- Stringent: $O(N^{1.5})$ - examine all contact points in the interface between tiles
- Greedy: $O(N \log N)$ - contact points are sorted by score; only curves through highest scoring points are examined
- Runtime:0.9 (0.6) secs on 129×129.