Improving local features by dithering-based image sampling

Christos Varytimidis, Konstantinos Rapantzikos,
Yannis Avrithis and Stefanos Kollias

{chrisvar,rap,iavr}@image.ntua.gr,
stefanos@cs.ntua.gr

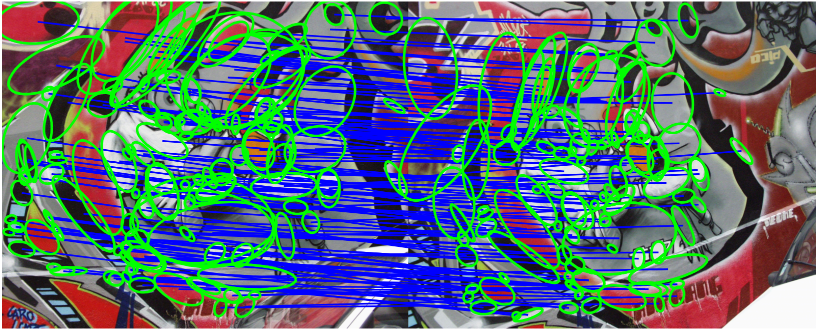
National Technical University of Athens

ACCV 2014

Outline

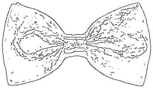
Local features

- Sparse image representation
- High distinctiveness when combined with local descriptors
- Exploited by many computer vision applications
  (stereo matching, object detection, image retrieval, *etc.*)

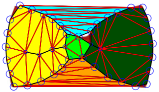


WαSH feature detector

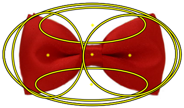
input



edges



samples & triangulation

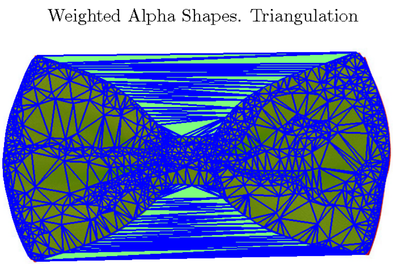


features

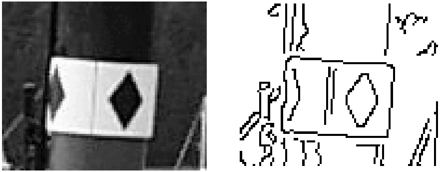
- *Weighted α-shapes* detector starts from sampled image edges (binary) [Varytimidis *et al.* '12]
- Uniform sampling along edges
- Intuitively, image edges are interpretable and repeatable
  - Nevertheless, automatically extracted binary edges can be noisy

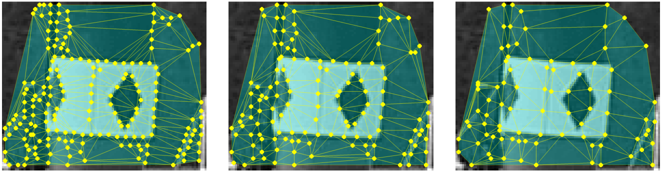
WαSH feature detector

- Triangulation of samples
- Hierarchy of triangles and edges (*filtration*) based on size
- α-shapes are a generalization of the *Convex Hull*
- Each instance of the filtration corresponds to an α value
- α-shapes are nested subsets of the triangulation
- Connected components of the α-shapes are candidate image features



Uniform sampling along binary image edges





$s_1 \quad \leq \quad s_2 \quad \leq \quad s_3$

- Binary edges can be noisy
- Fixed step s along the edge
  - Need for fine-tuning

Proposed Image sampling

Novel image sampling that:

- fires mainly on object boundaries
- is parameter free
- sampling density is based on local image properties

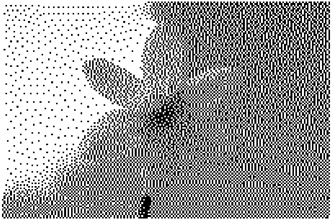
Combined with WαSH, local features:

- capture regions with different levels of detail
- better follow object boundaries
- ????????????

Image dithering

- Dithering uses error–diffusion to minimize quantization error
- Results are visually similar to the original
- Binary images can be interpreted as sampled points
- Functions other than image intensity may also be used

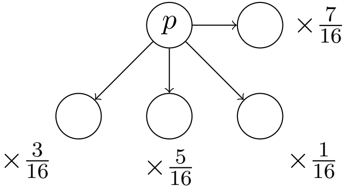
grayscale



dithered

Error diffusion algorithm

- Floyd–Steinberg algorithm [Floyd and Steinberg '76]
  - Fast – only one pass over the image pixels
  - Visually appealing results
  - Easy to implement

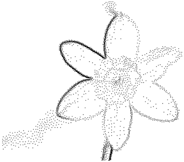


Gradient-based error diffusion

- $G = \|\nabla g(\sigma) * I\|$, gradient strength
- $\hat{G}(x, y)$, normalized to $[0, 1]$
- $s(x, y) = \hat{G}(x, y)^\gamma, \gamma > 0$
- error diffusion step

input

$\hat{G}$

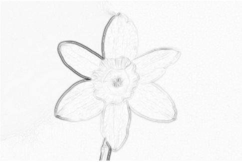


sampling

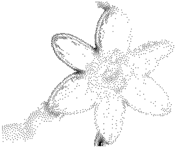
Hessian-based error diffusion

- $\lambda_{\max}(x, y)$, largest eigenvalue at (x, y) of Hessian
- $\hat{\lambda}_{\max}(x, y)$, normalized to $[0, 1]$
- $s(x, y) = \hat{\lambda}_{\max}(x, y)^{\gamma}$
- error diffusion step

input



$\hat{\lambda}_{\max}$



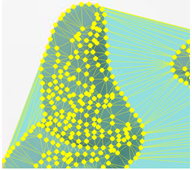
sampling

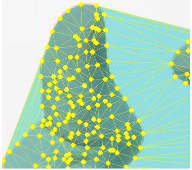
Improved local features

Uniform sampling (WαSH)

input

binary edges

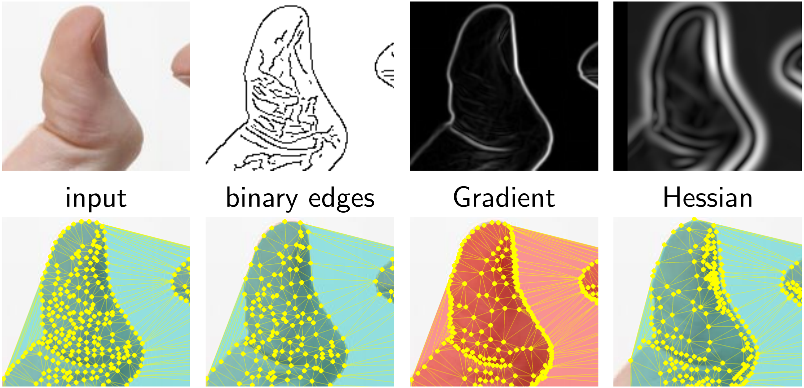




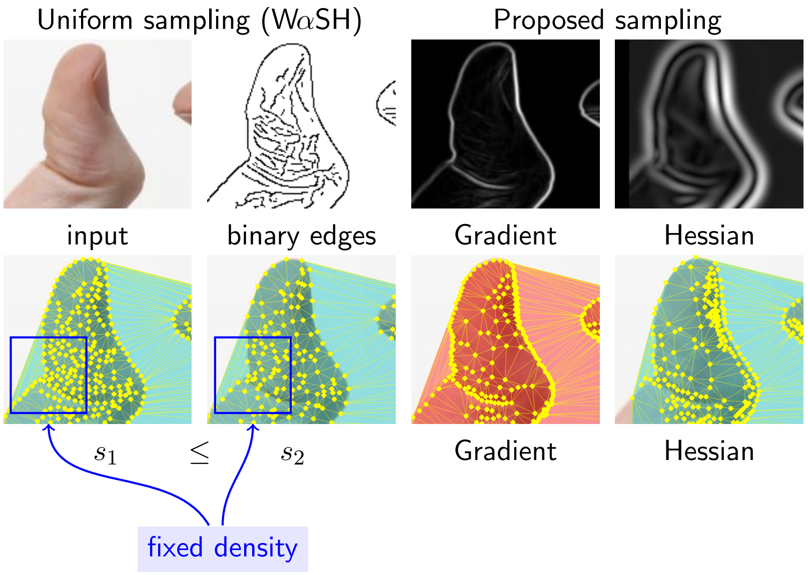
$s_1 \leq s_2$

Improved local features

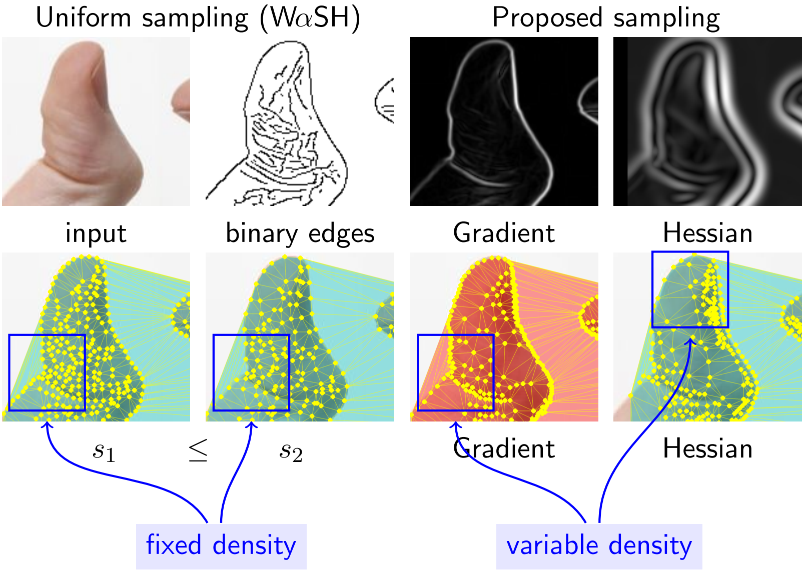
Uniform sampling (WαSH) Proposed sampling



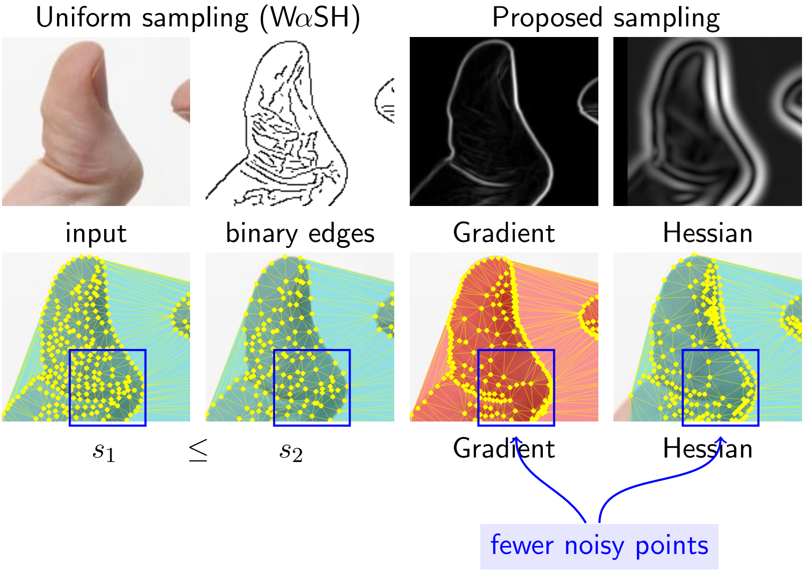
Improved local features



Improved local features



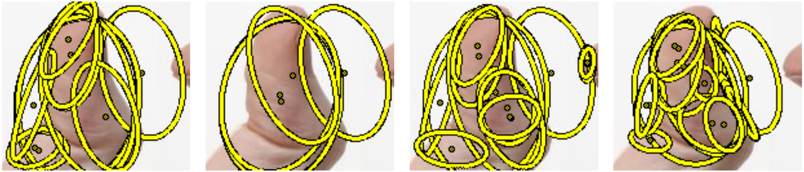
Improved local features



Improved local features

Uniform sampling (WαSH) Proposed sampling

input binary edges Gradient Hessian



$s_1 \leq s_2$ Gradient Hessian

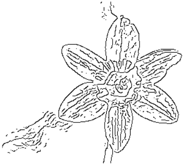
Examples

- Input image

- Object consists of well–defined parts
- Object parts are textured
- ??????

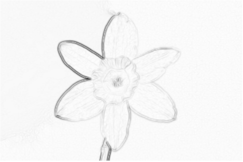
Examples

- Image function to sample



edges

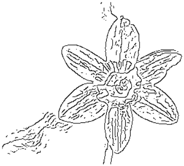
gradient



Hessian

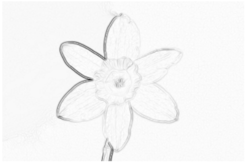
Examples

- Image function to sample



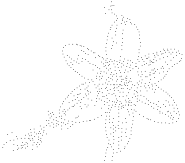
edges

gradient

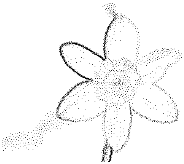


Hessian

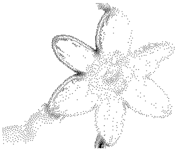
- Sample points



uniform



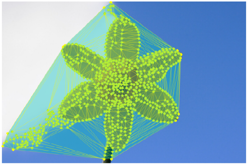
gradient



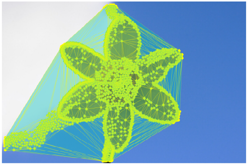
Hessian

Examples

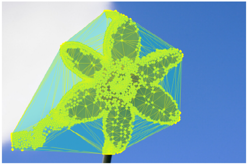
- Sample points and triangulation



uniform



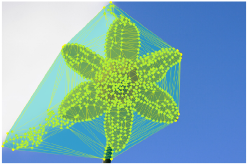
gradient



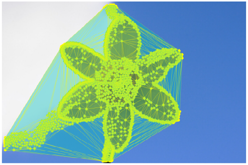
Hessian

Examples

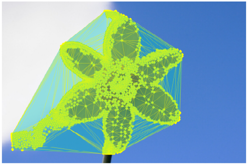
- Sample points and triangulation



uniform

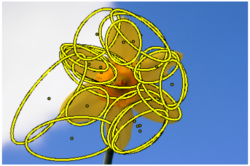


gradient

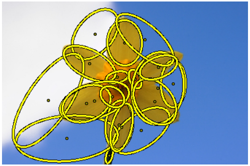


Hessian

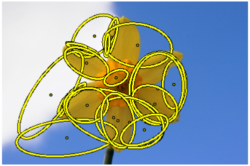
- WαSH detected features



uniform



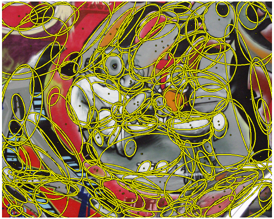
gradient



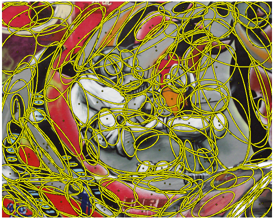
Hessian

Evaluation – Repeatability, matching score

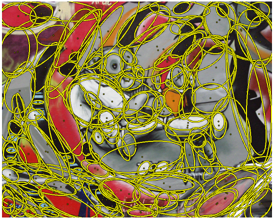
- Metrics and dataset from [Mikolajczyk *et al.* '05]
- VLBenchmarks evaluation framework [Lenc *et al.* '11]



WαSH (uniform)

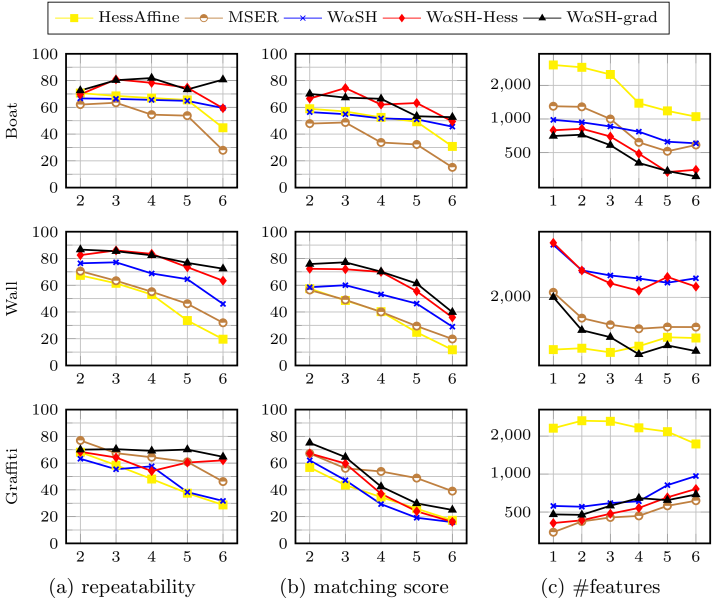


WαSH (gradient)

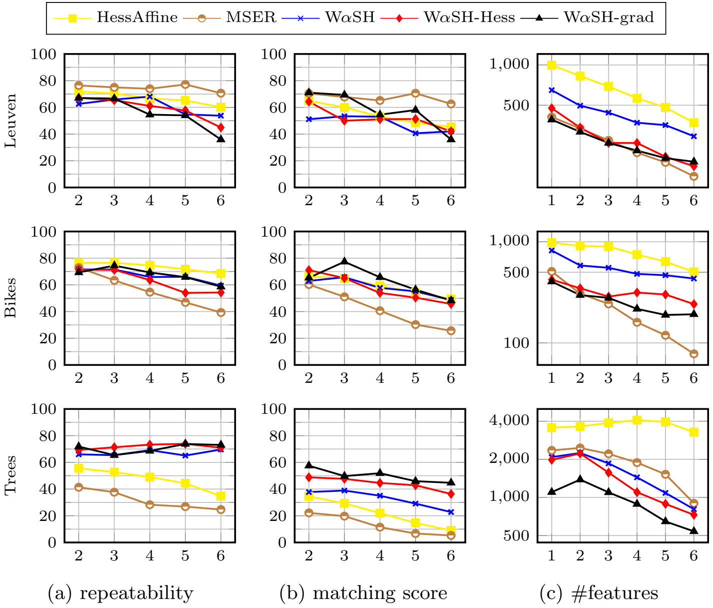


WαSH (Hessian)

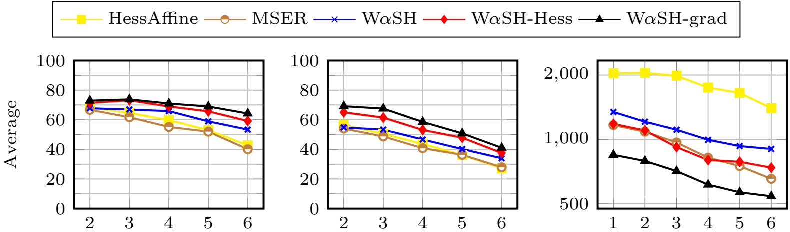
Evaluation – Repeatability, matching score



Evaluation – Repeatability, matching score



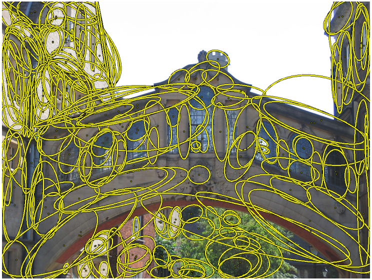
Evaluation – Repeatability, matching score



(a) repeatability (b) matching score (c) #features

Evaluation – Large scale image retrieval

- *Oxford 5K* dataset [Philbin *et al.* '07]
- SIFT descriptor for all detectors (except SURF)
- *approximate k-means* for clustering
- *fast spatial matching* for results verification



Evaluation – Large scale image retrieval

detector	features ($\times 10^6$)	Bag-of-Words (mAP)			ReRanking (mAP)		
		50K	100K	200K	50K	100K	200K
HessAff	29.02	0.483	0.539	0.573	0.518	0.577	0.607
MSER	13.33	0.487	0.534	0.565	0.519	0.569	0.595
SIFT	11.13	0.422	0.465	0.495	0.441	0.486	0.517
SURF	**6.84**	0.465	0.526	0.574	0.509	0.573	0.603
WαSH	7.19	0.529	0.569	0.590	0.537	0.569	0.585
WαSH, grad	7.63	**0.531**	**0.580**	**0.605**	**0.543**	**0.578**	**0.609**
WαSH, Hess	7.29	0.518	0.553	0.582	0.511	0.557	0.584

Thank you!