Adaptive Manifolds for Real-Time High-Dimensional Filtering

Milestones and Advances in Image Analysis

Markus Schwinn

November 27, 2012

Motivation

$$g_i = \frac{\sum_{p_j \in S} \phi(\hat{p}_i - \hat{p}_j) \cdot f_j}{\sum_{p_j \in S} \phi(\hat{p}_i - \hat{p}_j)}$$

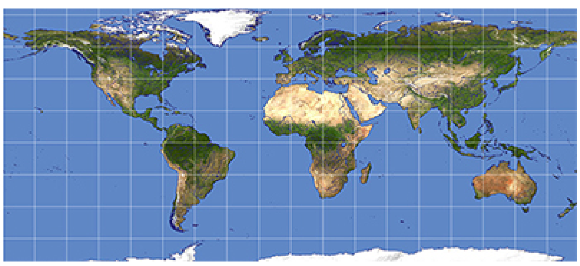
- "framework" for some high-dimensional filter
- ϕ is a Gaussian kernel
- possible filters for $2D$ colour images:
  - convolution $\rightarrow \hat{p} \in \mathbb{R}^2$
  - bilateral $\rightarrow \hat{p} \in \mathbb{R}^5$
  - non-local mean $\rightarrow \hat{p} \in \mathbb{R}^{3n^2+2}$
    n depends on window size
- powerful... *but very slow*

Questions to Solve

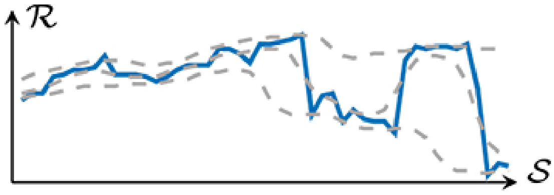
1. What are adaptive manifolds?
2. How to construct them?
3. What's the use of?

Manifolds Are Not Unknown!

- practical use of manifolds
  $\rightarrow$ projecting world onto a map



- approximation of input signal in high-dimensional space
- approximately linear w.r.t local neighbourhood



- For $2D$ colour image
  $\rightarrow$ dealing with 5-dimensional space
- Construction of one point on manifold
  $\rightarrow P(S_x, S_y, R_S, G_S, B_S)$
  S denotes point in spatial domain

1. low-pass filtering input signal
   $\rightarrow$ generates first manifold η_1

2. compute colour deviation of the pixels depending on manifold and original image

   more technical:
   largest eigenvector v_1 of

   $$(f_1 - \eta_1) \cdot (f_1 - \eta_1)^T$$

   $\rightarrow v_1$ describes variation of colour values

3. cluster pixels in two subsets.

   → depending on "main colour"
   defining *above* and *below* w.r.t first manifold

   more technical:

$$sign = v_1^T (f_i - \eta_{1i})$$

$$\mathcal{C}_+ \leftarrow p_i \text{ if } sign \geqslant 0$$
$$\mathcal{C}_- \leftarrow p_i \text{ if } sign < 0$$

Adaptive Manifolds
Markus Schwinn
Motivation
Adaptive Manifolds
Creating Adaptive Manifolds
The Algorithm
Runtime
Applications
Summary
References

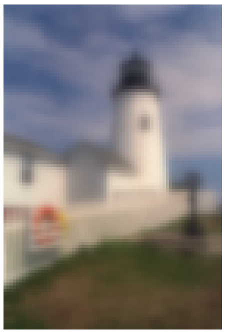
4. compute for each cluster manifolds η_+ and η_-
   higher weighting for pixels, *not represented well* in η_1

5. repeat up from Step 2
   until number of manifolds is reached

Adaptive Manifolds - Example

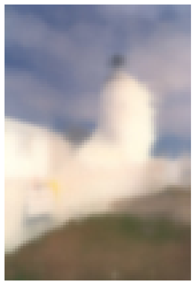
original image

$\rightsquigarrow$

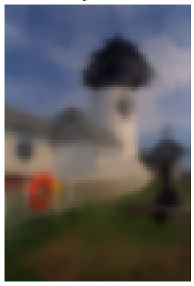


η_1

$\rightsquigarrow$



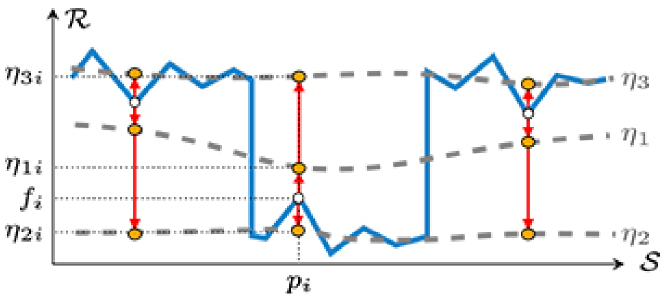
η_+



η_-

The Algorithm

- *Splatting*

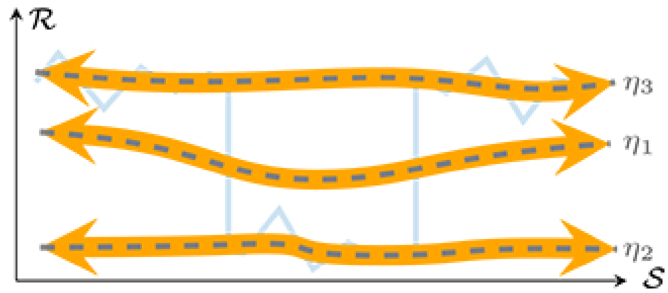


projects colour for each position onto each manifold Gaussian weighted with

$$\Psi_{splat}(\hat{\eta}_{ki}) = \phi(\eta_{ki} - f_i) f_i$$

ϕ is a Gaussian kernel

- *Blurring*



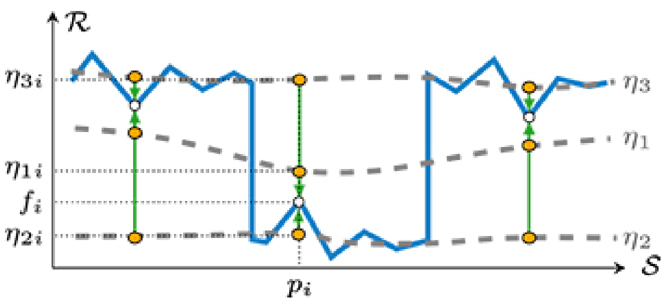
blurs over all manifolds

$$\Psi_{splat}(\hat{\eta}_{ki}) \rightsquigarrow \Psi_{blur}(\hat{\eta}_{ki})$$

changes information between sample points η_{ki}

Adaptive Manifolds
Markus Schwinn
Motivation
Adaptive Manifolds
Creating Adaptive Manifolds
The Algorithm
Runtime
Applications
Summary
References

- *Slicing*



compute filter response
interpolates by blurred values over all adapted manifolds

$$g_i = \frac{\sum_{p_j \in S} \phi(\hat{p}_i - \hat{p}_j) \cdot f_j}{\sum_{p_j \in S} \phi(\hat{p}_i - \hat{p}_j)}$$

$$\rightsquigarrow g_i = \frac{\sum_{k=1}^{K} \phi(\hat{p}_i - \hat{p}_j) \cdot \Psi_{blur}(\hat{\eta}_{ki})}{\sum_{k=1}^{K} \phi(\hat{p}_i - \hat{p}_j)}$$

$$\rightsquigarrow g_i = \frac{\sum_{k=1}^{K} \phi(\eta_{ki} - f_i) \cdot \Psi_{blur}(\hat{\eta}_{ki})}{\sum_{k=1}^{K} \phi(\eta_{ki} - f_i)}$$

$$\rightsquigarrow g_i = \frac{\sum_{k=1}^{K} \phi(\eta_{ki} - f_i) \cdot \Psi_{blur}(\hat{\eta}_{ki})}{\sum_{k=1}^{K} \phi(\eta_{ki} - f_i)\Psi_{blur}(\phi(\eta_{ki} - f_i))}$$

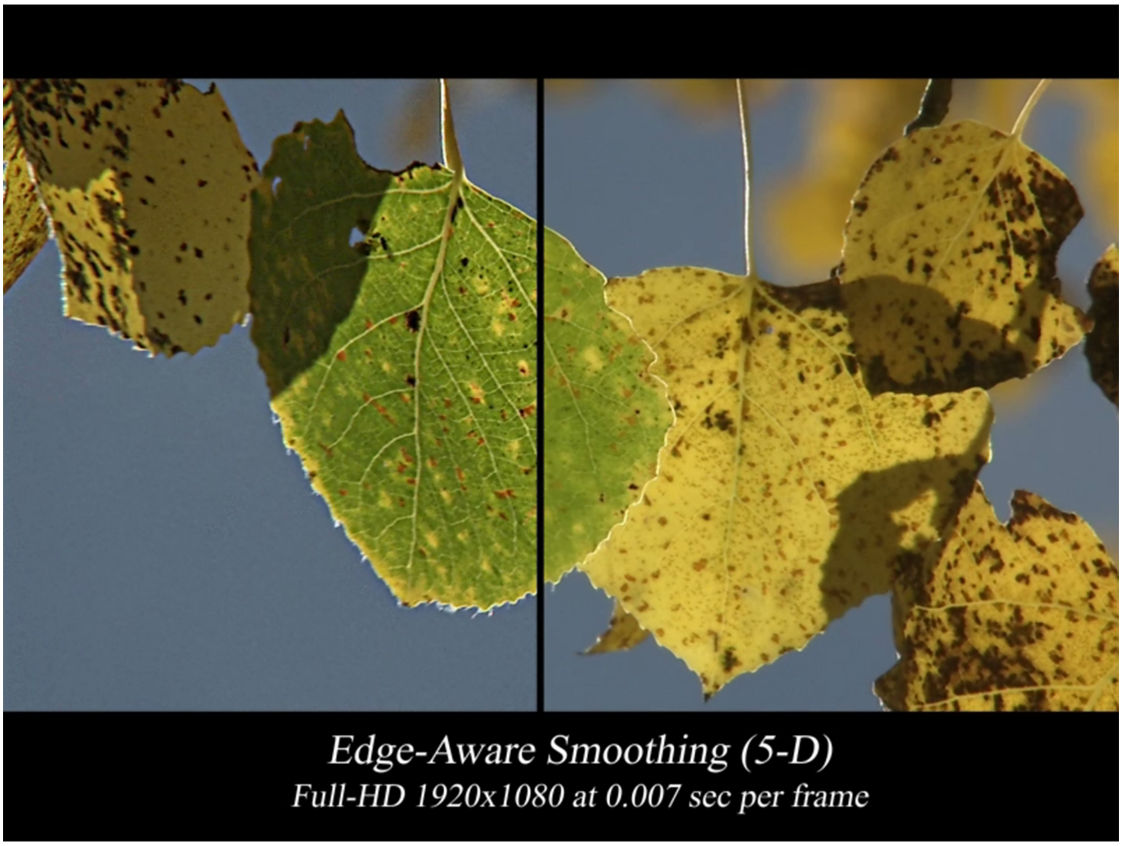
$$\Rightarrow g_i = \frac{\sum_{k=1}^{K} \phi(\eta_{ki} - f_i) \cdot \Psi_{blur}(\hat{\eta}_{ki})}{\sum_{k=1}^{K} \phi(\eta_{ki} - f_i)\Psi_{blur}(\phi(\eta_{ki} - f_i))}$$

Adaptive Manifolds
Markus Schwinn
Motivation
Adaptive Manifolds
Creating Adaptive Manifolds
The Algorithm
Runtime
Applications
Summary
References

- independent of number of pixels and dimension
- no general mechanism, sensitive to the problem
- but: depends of standard deviation of spatial- and range domain

- Runtime
  - clustering $\rightarrow O(dN \log K)$
  - computing manifolds $\rightarrow O(dNK)$
  - performing filter $\rightarrow O(dNK + dNK)$
  - *in total: $O(dNK)$ with $K = const \Rightarrow O(dN)$*

$\Rightarrow$ high performance for runtime and good storage allocation



Adaptive Manifolds
Markus Schwinn
Motivation
Adaptive Manifolds
Creating Adaptive Manifolds
The Algorithm
Runtime
Applications
Summary
References

- add additional channels for more information
- adding a infrared channel
  $\rightarrow$ improving result

noisy image

infrared image

Adaptive Manifolds

Markus Schwinn

Motivation

Adaptive Manifolds

Creating Adaptive Manifolds

The Algorithm

Runtime

Applications

Summary

References

- denoised results

without IR channel

with IR channel

- advantages
  - adaptable for a "general framework"
  - runtime linear in number of pixels and dimension
  - euclidean and also geodesic filters adaptable
- drawbacks
  - sensitive to number of manifolds
  - choose of Gaussian kernels (standard deviation)

Adaptive Manifolds
Markus Schwinn
Motivation
Adaptive Manifolds
Creating Adaptive Manifolds
The Algorithm
Runtime
Applications
Summary
References

- literature

Gastal, E. S. L. and M. M. Oliveira: *Adaptive Manifolds for Real-Time High-Dimensional Filtering*. ACM TOG, 31(4):33:1–33:13, 2012. Proceedings of SIGGRAPH 2012.

- figures

1 http://it-material.de/IT-online5/wp-content/uploads/2009/09/sv003.jpg
2 http://earth.imagico.de/maps/earth_large.jpg

Thank you for your Attention