Nonlinear Diffusion in Computer Vision and Statistical Shape Models, with Applications in Image Analysis of Articulators of Voiced and Signed Speech

PhD Work Presentation

Anastasios Roussos

November 2010

Contents

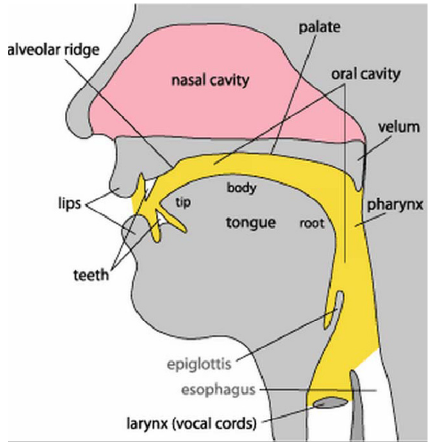
PDEs & Shape Models in Computer Vision

- Partial Differential Equations (PDEs) in Computer Vision (CV) and Image Processing
  - Started in 1980's
  - Popular due to various advantages compared to classic approaches
  - Development of Scale Spaces
  - Nonlinear diffusion for Computer Vision problems
  - Active Contours for Image Segmentation
  - Optical Flow

- Statistical Shape Models
  - Exploitation of prior shape information
  - They are generative and deformable
  - Object tracking and classification: model fitting
  - Active Shape Models, Active Appearance Models

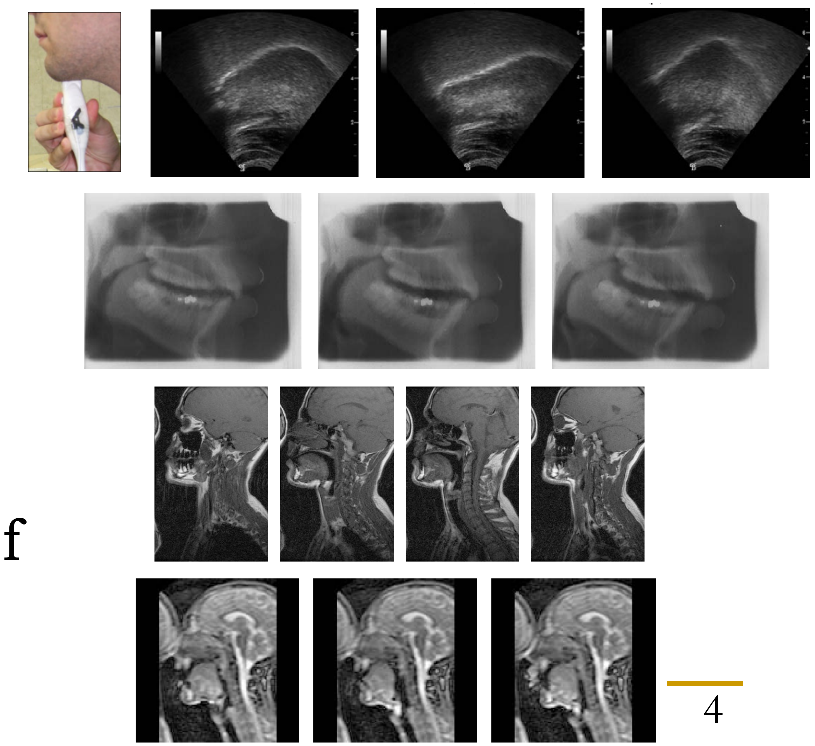
Research on Human Speech Production System

- Sub-problems
  - Articulated Speech Synthesis
  - Audio-visual Speech Inversion

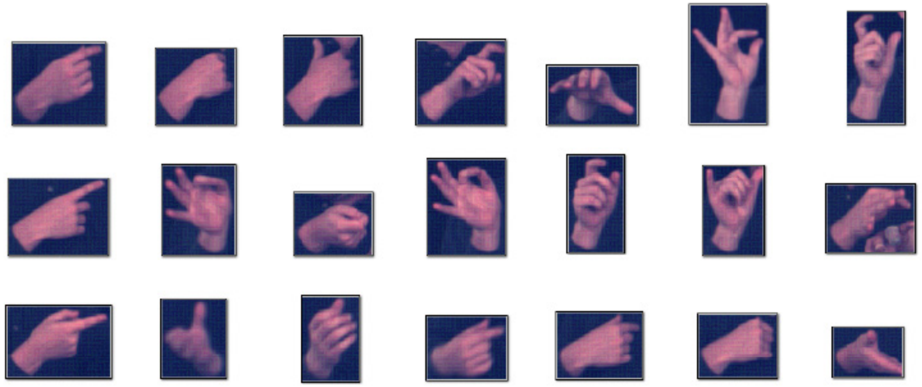


Speech Articulators & Vocal Tract

- Articulatory image data during speech
  - Acquisition techniques
  - Image enhancement using digital post-processing
  - Image analysis for extraction of geometric information



Automatic Sign Language Recognition

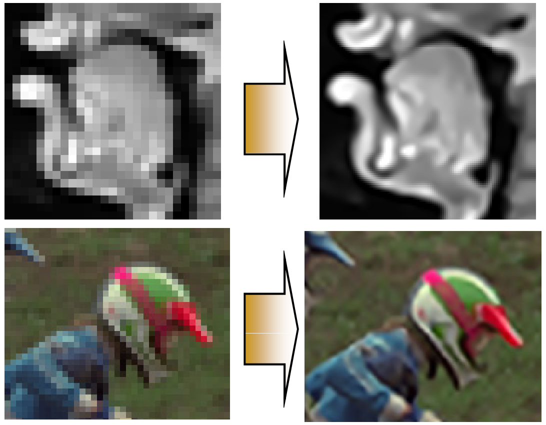


- Sub-problems
  - Localization & tracking of signer's hands and head
  - Extraction of features that reliably describe the hand configurations
- Difficulties
  - Fast hands movement
  - Occlusions
  - High variability on the hand pose and shape

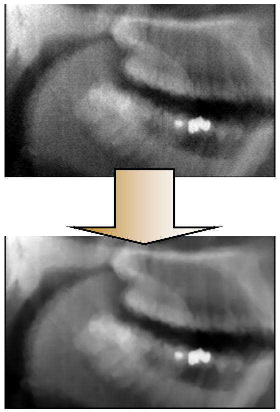
PhD Research Contributions

Nonlinear diffusion methods for image enhancement

Nonlinear diffusion for vector-valued image interpolation

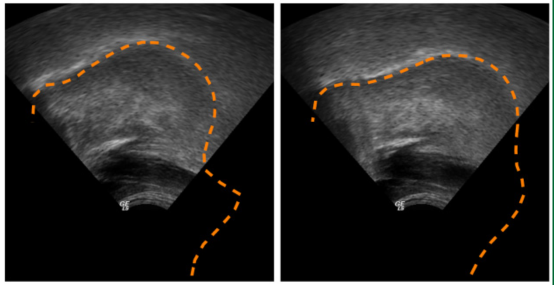


Generalized variational framework for regularized nonlinear diffusion

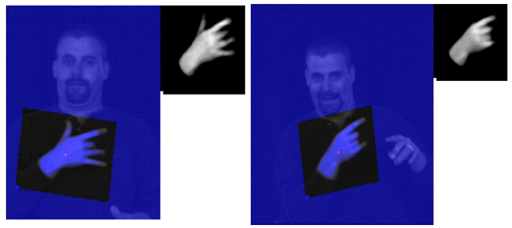


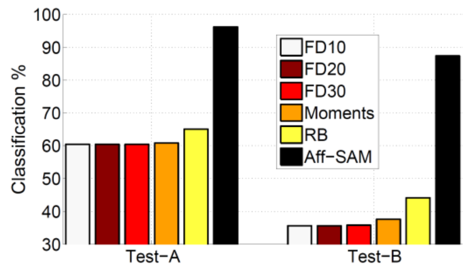
Statistical Shape Models

Tongue tracking with Active Appearance Models



Affine-invariant Shape-Appearance modeling





Speech Production System Research

Sign Language Recognition

Contents

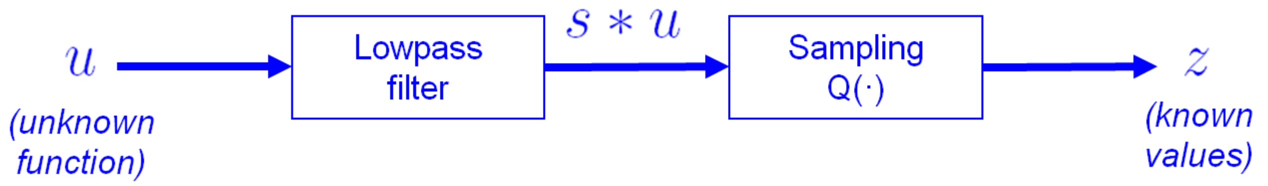
- Introduction
  - PDEs & Shape Models in Computer Vision
  - Applications
  - Research Contributions
- Nonlinear Diffusion for Image Interpolation
- Variational Frameworks for Tensor-based Diffusion
- Tongue Tracking with Active Appearance Models
- Handshape Modeling for Sign Language
- Conclusions

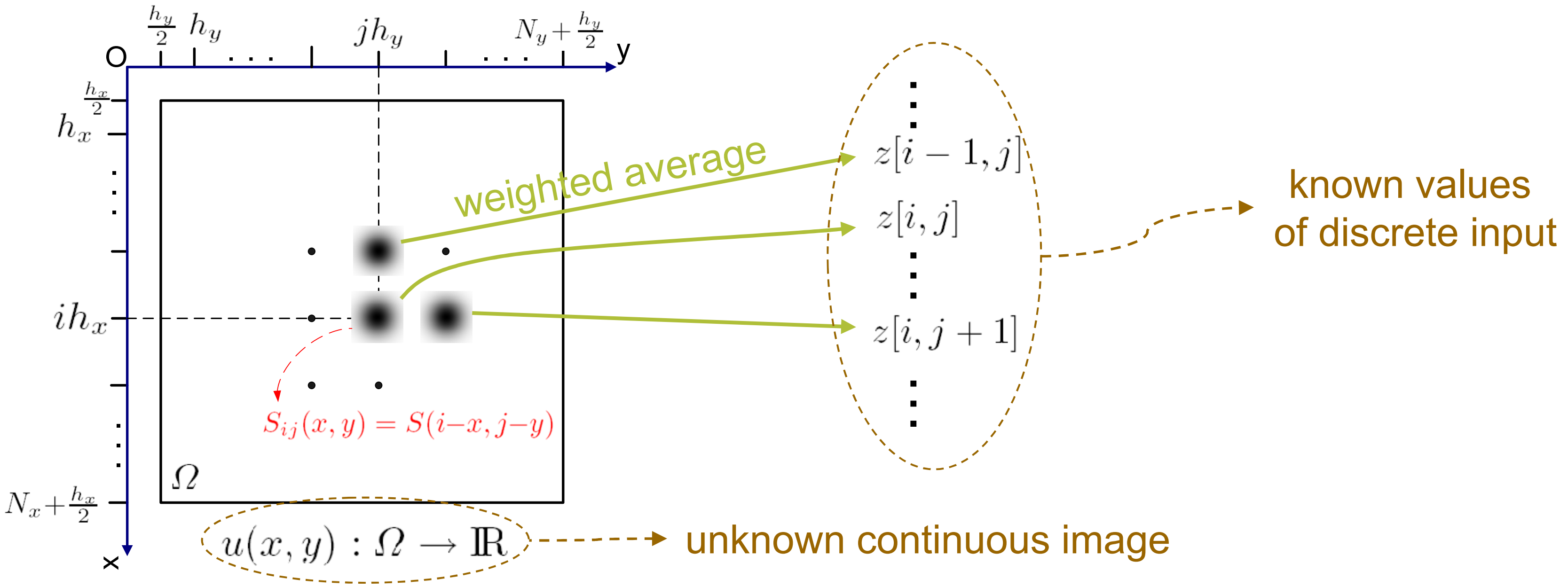
Image Interpolation

- Can be defined as the operation that:
  - takes as input a discrete image and
  - recovers a continuous image or a discrete one of higher resolution
- Fundamental Image Processing problem with various applications:
  - biomedical image processing, aerial & satellite imaging, text recognition and high quality image printing
- Pre-processing step in various Computer Vision problems, such as:
  - Image Segmentation, Feature Detection, Object Recognition and Motion Analysis
- Classes of methods
  - Classic linear methods
  - Adaptive nonlinear methods

Reversibility Condition Approach for Interpolation

- Similar problem formulation to [Malgouyres,Guichard, SIAM J. Num. Anal. '01]
- The solution must satisfy a reversibility condition:





$$(S * u)(i_1 h_x, i_2 h_y) = z[i_1, i_2], \ \ \forall\, (i_1, i_2) \in \{1,..,N_x\} \times \{1,..,N_y\}$$

Nonlinear Diffusion Method for Image Interpolation

[Roussos,Maragos SSVM 07], [Roussos,Maragos IJCV 09]

- Novel *Partial Differential Equation (PDE) flow* that:
  - is designed for general vector-valued images (e.g. color)
  - evolves in the subspace $\mathcal{U}_{z,S}$ of functions that satisfy the reversibility condition
  - performs iterative adaptive smoothing, leading to elements of $\mathcal{U}_{z,S}$ with "better" visual quality

Proposed PDE for Image Interpolation (1)

$$\frac{\partial u_m(\boldsymbol{x},t)}{\partial t} = \underbrace{P_{\mathcal{U}_{0,S}}}_{\text{projection operator}} \left\{ \operatorname{div}\left(\underbrace{T\left(J_\rho(\nabla \boldsymbol{u}_\sigma)\right)}_{\text{2x2 diffusion tensor}} \nabla u_m \right) \right\} \; , \; m=1,..,M$$

artificial time

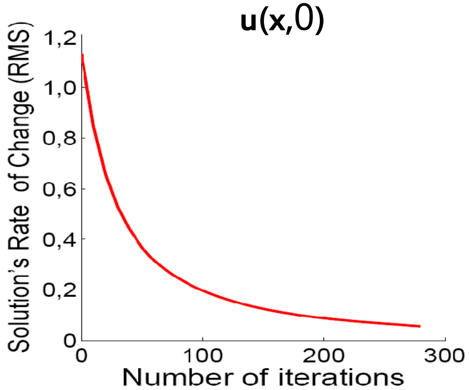
$\boldsymbol{u}(\boldsymbol{x},0)$ = zero-padding high frequencies ($\in \mathcal{U}_{z,S}$)

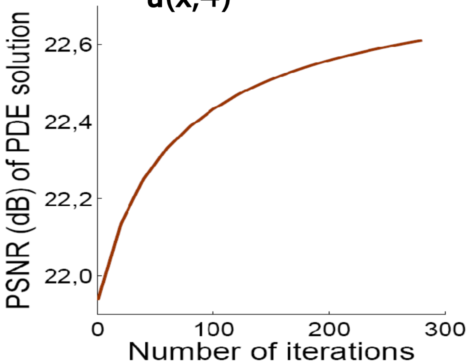
input **z**

u(x,0)

u(x,4)

u(x,56)





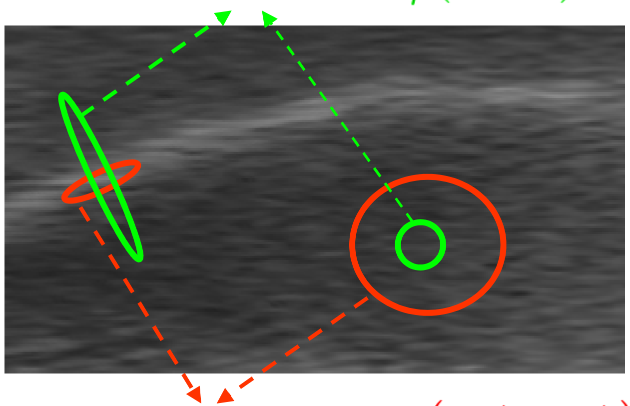
Proposed PDE for Image Interpolation (2)

$$\frac{\partial u_m(\boldsymbol{x},t)}{\partial t} = \underbrace{P_{\mathcal{U}_{0,S}}}_{\text{projection operator}} \left\{ \operatorname{div}\left(\underbrace{T\left(J_\rho(\nabla \boldsymbol{u}_\sigma)\right)}_{\textbf{2x2 diffusion tensor}} \nabla u_m \right) \right\}, \quad m=1,..,M$$

artificial time

$\boldsymbol{u}(\boldsymbol{x},0)$ = zero-padding high frequencies ($\in \boldsymbol{\mathcal{U}}_{z,S}$)

structure tensor $J_\rho(\nabla u_{\boldsymbol{\sigma}})$



diffusion tensor $T\left(J_\rho(\nabla u_{\boldsymbol{\sigma}})\right)$

$$J_\rho(\nabla \boldsymbol{u}_\sigma) = G_\rho * \sum_{m=1}^{M} \nabla(G_\sigma * u_m)\,\left(\nabla(G_\sigma * u_m)\right)^{\mathrm{T}}$$

$$T\left(J_\rho(\nabla \boldsymbol{u}_\sigma)\right) = \left[1+(\mathcal{N}/K)^2\right]^{-\frac{1}{2}} \boldsymbol{w}_- \boldsymbol{w}_-^{\mathrm{T}} + \left[1+(\mathcal{N}/K)^2\right]^{-1} \boldsymbol{w}_+ \boldsymbol{w}_+^{\mathrm{T}}$$

$$\mathcal{N} = \sqrt{\lambda_+ + \lambda_-}$$

Proposed PDE for Image Interpolation (3)

$$\frac{\partial u_m(\boldsymbol{x},t)}{\partial t} = \underbrace{P_{\mathcal{U}_{0,S}}}_{\textbf{projection operator}} \left\{ \operatorname{div}\left(\underbrace{T\left(J_\rho(\nabla \boldsymbol{u}_\sigma)\right)}_{\text{2x2 diffusion tensor}} \nabla u_m \right) \right\} \ , \ m=1,..,M$$

artificial time

$\boldsymbol{u}(\boldsymbol{x},0)$ = zero-padding high frequencies ($\in \boldsymbol{\mathcal{U}}_{z,S}$)

$$u \in \mathcal{U}_{z,S} \iff \sum_{(k_1,k_2)\in\mathbb{Z}^2} \hat{S}\left(\tfrac{2\pi}{\widetilde{N}_x}(n_1+k_1\widetilde{N}_x), \tfrac{2\pi}{\widetilde{N}_y}(n_2+k_2\widetilde{N}_y)\right) \cdot \hat{u}_{n_1+k_1\widetilde{N}_x, n_2+k_2\widetilde{N}_y} = \hat{z}_{n_1,n_2}$$

$$\begin{cases} P_{\mathcal{U}_{0,S}}\{v\} = v(\boldsymbol{x}) - w(\boldsymbol{x}) \ , \\ \hat{w}_{m_1,m_2} = \left\{ \displaystyle\sum_{(k_1,k_2)\in\mathbb{Z}^2} \hat{\phi}\left(\tfrac{2\pi m_1}{\widetilde{N}_x}+k_1 2\pi, \tfrac{2\pi m_2}{\widetilde{N}_y}+k_2 2\pi\right) \cdot \hat{v}_{m_1+k_1\widetilde{N}_x, m_2+k_2\widetilde{N}_y} \right\} \cdot \hat{\phi}\left(\tfrac{2\pi m_1}{\widetilde{N}_x}, \tfrac{2\pi m_2}{\widetilde{N}_y}\right) \\ \hat{\phi}(\omega_1,\omega_2) = \left\{ \displaystyle\sum_{(k_1,k_2)\in\mathbb{Z}^2} \left| \hat{S}\left(\omega_1+k_1 2\pi, \omega_2+k_2 2\pi\right) \right|^2 \right\}^{-\frac{1}{2}} \cdot \overline{\hat{S}(\omega_1,\omega_2)} \end{cases}$$

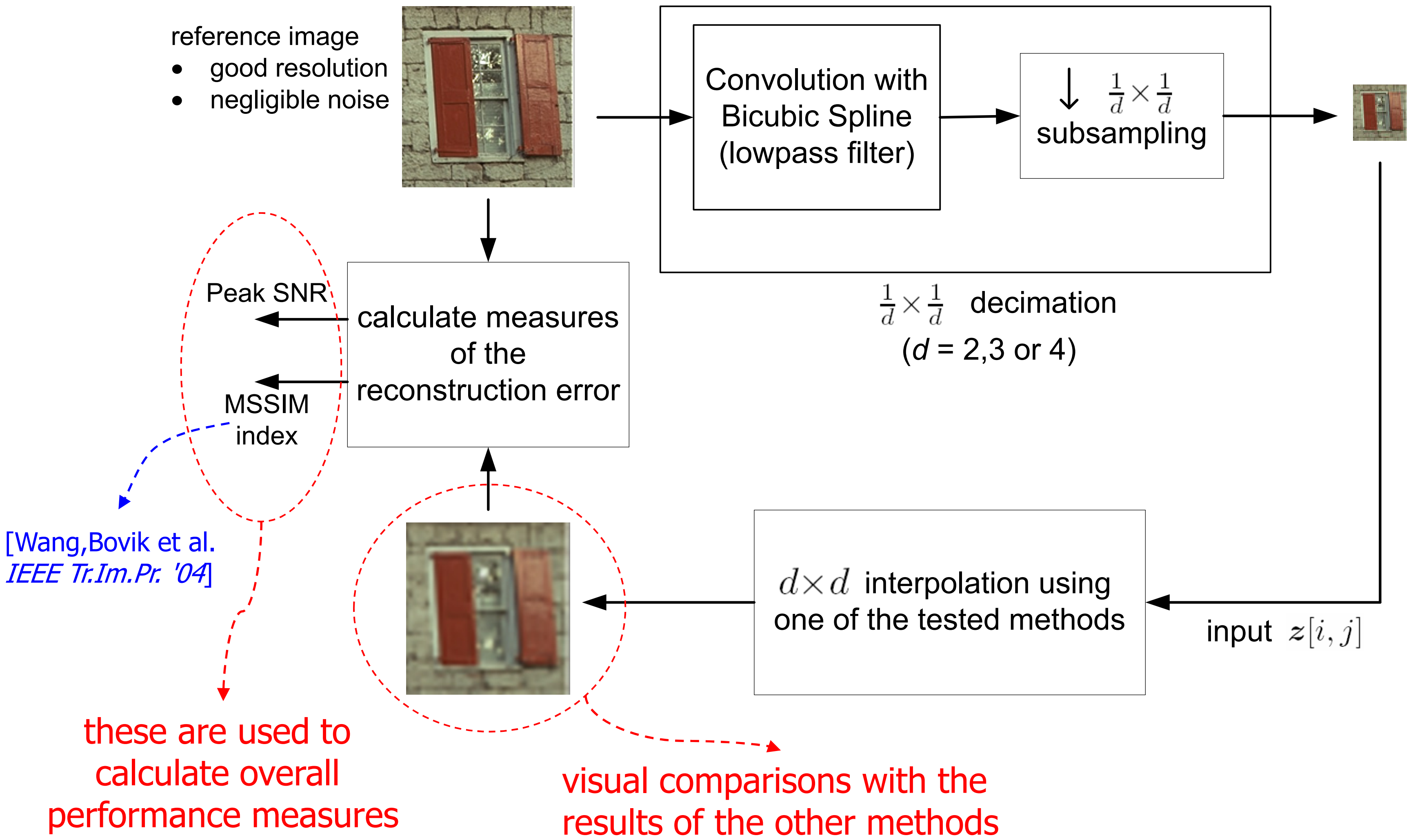
Previous PDE-based interpolation methods

- Total Variation (TV) - based Interpolation
  [Malgouyres,Guichard, *SIAM J. Num. Anal. 01*]

- Belahmidi-Guichard method (BG)
  [Belahmidi,Guichard, *ICIP* 04]

- Tschumperle-Deriche (TD) method
  [Tschumperle,Deriche, IEEE-PAMI 05]

Interpolation Experiments: Framework



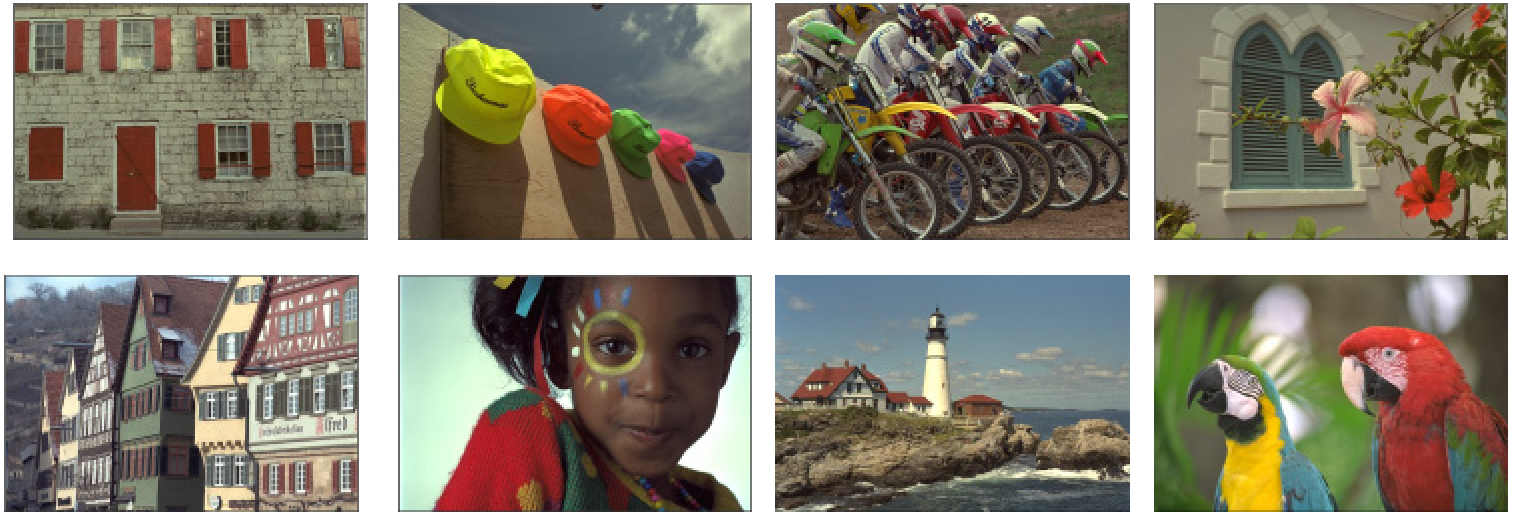
Interpolation Experiments: Data Set

- This framework has been repeated for reference images from the CIPR dataset:

  www.cipr.rpi.edu/resource/stills/kodak.html

  23 natural images of size 768 x 512 pixels

- Both graylevel & color versions of images have been used



8 out of 23 images of the dataset

Graylevel Image Interpolation Example (4x4)

(a) Input (enlarged by ZOH)
PSNR=25.58, MSSIM=0.758

(b) Bicubic interpolation
PSNR=26.95, MSSIM=0.815

(c) TV, sinc kernel
PSNR=27.92, MSSIM=0.846

(d) TV, mean kernel
PSNR=27.27, MSSIM=0.831

(e) BG interpolation
PSNR=26.89, MSSIM=0.818

(f) Our method
PSNR=28.54, MSSIM=0.868

Color Image Interpolation Example (4x4)

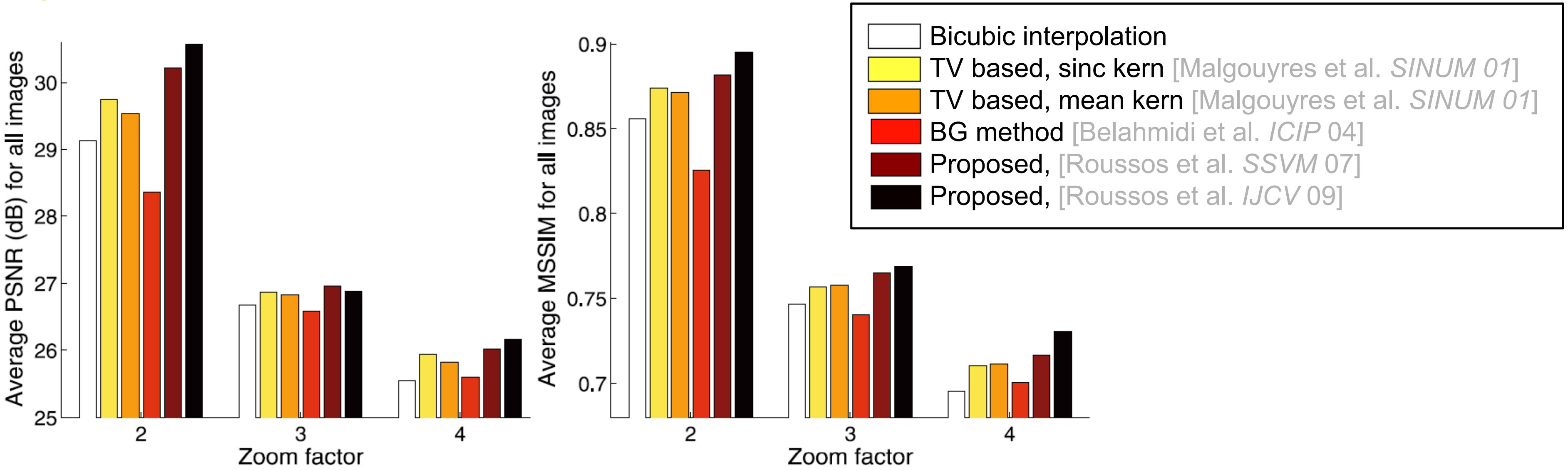
(a) Input (enlarged by ZOH)
PSNR=20.87, MSSIM=0.523

(b) Bicubic interpolation
PSNR=21.85, MSSIM=0.579

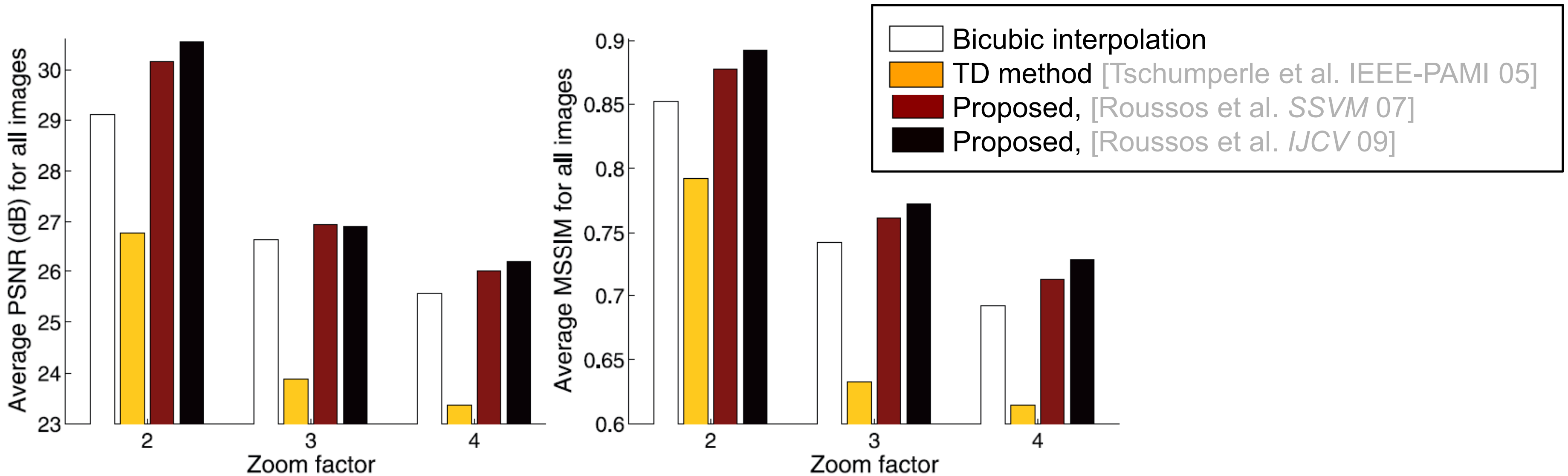
(c) TD interpolation
PSNR=19.89, MSSIM=0.458

(d) Proposed method
PSNR=22.63, MSSIM=0.652

Interpolation Experiments: Overall Measures



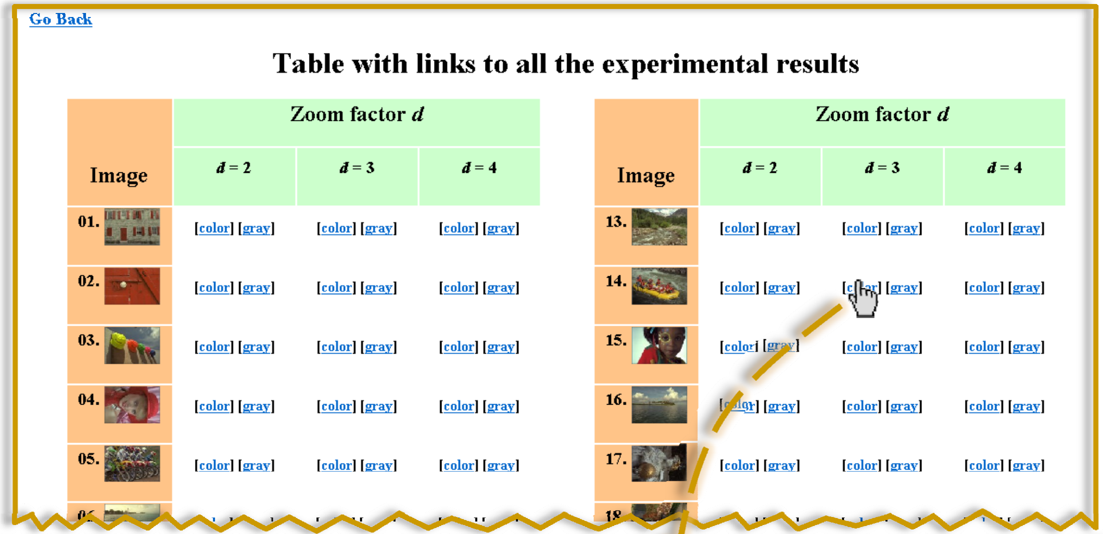
(a) Experiments with graylevel images



(b) Experiments with color images

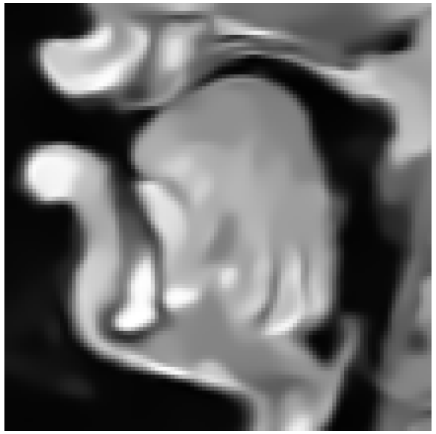
Full set of results available online

cvsp.cs.ntua.gr/~tassos/PDEinterp/ssvm07res

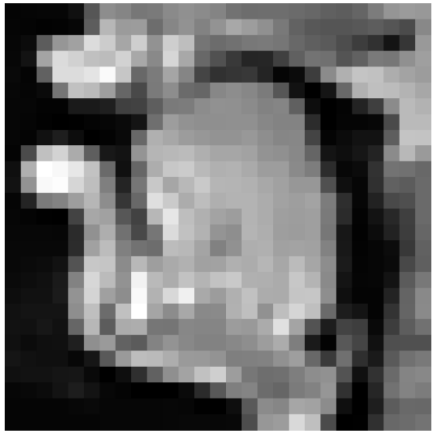


- Comparative demonstrations of all ~830 result images

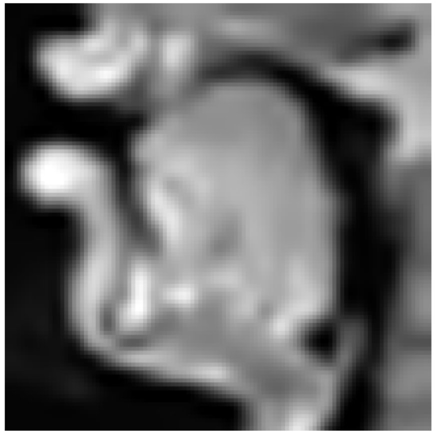
Vocal Tract Image Interpolation Example (4x4)



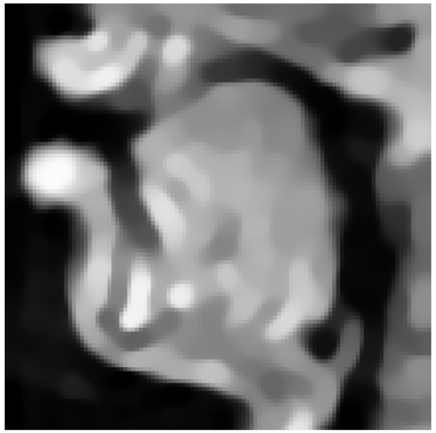
(a) Reference image
(108×108 pixels)



(b) Input (enlarged by ZOH)
PSNR=21.60, MSSIM=0.713

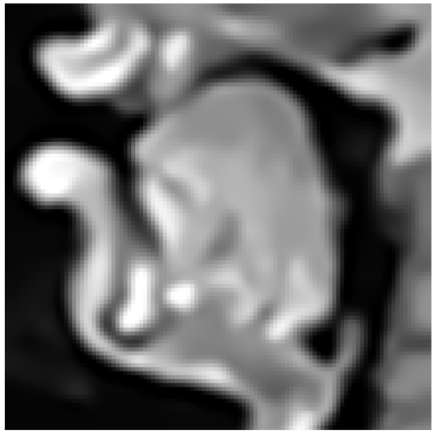


(c) Bicubic interpolation
PSNR=25.39, MSSIM=0.852



(d) TV, sinc kernel
PSNR=26.14, MSSIM=0.870

(e) BG interpolation
PSNR=25.88, MSSIM=0.870



(f) Proposed method
PSNR=27.69, MSSIM=0.904

Contents

Variational Frameworks for Diffusion: Motivation (1/2)

- Nonlinear diffusion models for Computer Vision
  - Class A: Directly-designed PDEs
    - Perona-Malik method [ieeeT-PAMI'90]
    - CLMC regularized PDE [Catte et al, siamJNA'92]
    - Coherence-enhancing diffusion [Weickert, IJCV'99]
    - Method of [Tschumperlé & Deriche, ieeeT-PAMI'05]

      ⋮
  - Class B: Variational Methods
    - Total Variation [Rudin, Osher & Fatemi, PhysicaD'92]
    - Vectorial Total Variation [Sapiro, CVIU'97]
    - Color Total Variation [Blomgren & Chan, ieeeT-IP'98]
    - Beltrami Flow [Sochen, Kimmel & Maladi, ieeeT-IP'98]

      ⋮
- For some methods of Class A: known connection with Class B, e.g. :
  - Perona-Malik model $\dfrac{\partial u(x,y,t)}{\partial t} = \operatorname{div}\left(g(\|\nabla u\|^2)\nabla u\right)$
  - $\displaystyle\min_u \int_\Omega \varphi(\|\nabla u\|^2)\mathrm{d}\boldsymbol{x} \rightsquigarrow \frac{\partial u}{\partial t} = \operatorname{div}\left(2\varphi'(\|\nabla u\|^2)\nabla u\right)$

    $g(s^2) = 2\varphi'(s^2)$
- But, for several types of PDE-based diffusion methods no variational interpretation existed

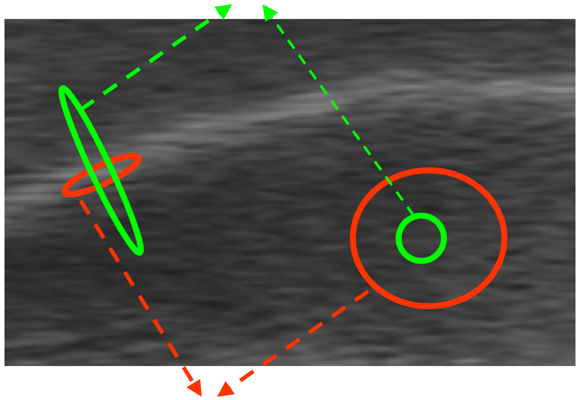
Variational Frameworks for Diffusion: Motivation (2/2)

- Advantages of variational interpretation of diffusion methods
  - conceptually clear formalism
  - helps with the reduction of model parameters
  - easier application to problems that can be formulated as constrained energy minimization, e.g.:
    - image restoration, inpainting, interpolation
  - can lead to efficient implementations based on optimization techniques

- Advantages of using tensors in image diffusion
  - Structure tensor

    reliable measure of the image variation & geometry in the neighborhood of each point
  - Diffusion tensor

    flexible adaptation to the image structures

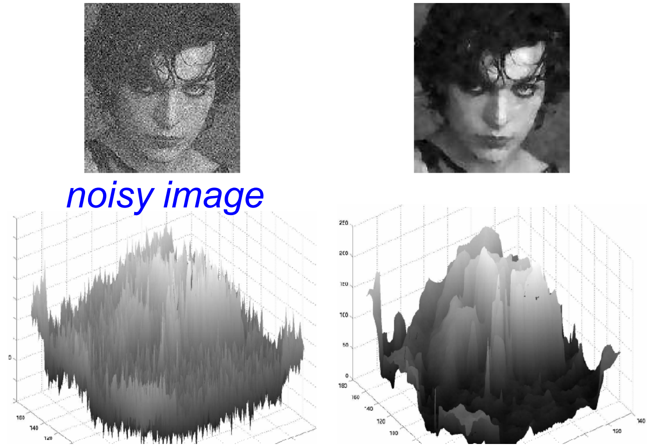


Generalization of the Beltrami Functional (1/2)

- Original Beltrami Flow
  [Sochen, Kimmel & Maladi, IEEE T-IP 98]
  - Interpretation of a vector-valued image u with n channels as a 2D surface embedded in R^{n+2}:

$$(x, y) \longrightarrow (x\,,\; y\,,\; u_1(x,y)\,,\; u_2(x,y)\,,\; \ldots\; u_n(x,y))$$

  - Flow towards the minimization of the surface area: tensor-based diffusion



example for the simplest case n=1

Figure from [Tschumperle, Thesis'02]

  - It offers an elegant way to:
    - couple the image channels and
    - extend in the vector-valued case the properties of Total Variation
  - But, the diffusion tensor is not regularized (no neighborhood info)
    → limitations on the robustness to noise & edge enhancement

- To overcome these limitations, we generalize the Beltrami Functional …

Generalization of the Beltrami Functional (2/2)

- Proposed generalization of the Beltrami functional:

  - We use higher dimensional mappings of the form:

$$\boldsymbol{x} \longrightarrow (\boldsymbol{x}, \mathcal{P}^{\boldsymbol{u}}(\boldsymbol{x}))$$

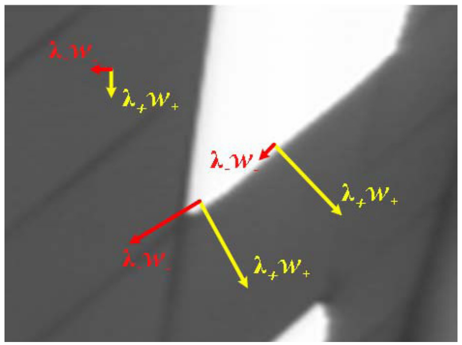
image patch [Tschumperle & Brun, ICIP'09], *that contains weighted image values not only at point* $\boldsymbol{x}$ *but also at points in a window around it*

  - In this way, each $\boldsymbol{x}$ contributes to the area of the embedded surface by considering the image variation in its neighborhood

  - If the patch sampling step → 0, the area of the embedded surface tends to:

$$A[\boldsymbol{u}] = \int_{\Omega} \sqrt{(\alpha^2 + \lambda_1)\,(\alpha^2 + \lambda_2)}\mathrm{d}\boldsymbol{x}$$

    - $\lambda_i = \lambda_i(J_K(\nabla \boldsymbol{u}))$: eigenvalues of the structure tensor $J_K(\nabla \boldsymbol{u}) = K * \sum \nabla u_i \otimes \nabla u_i$



Generalized Functional based on the Structure Tensor

- $E[\boldsymbol{u}] = \int_\Omega \psi\left(\lambda_1(J_K(\nabla \boldsymbol{u})), \lambda_2(J_K(\nabla \boldsymbol{u}))\right) \mathrm{d}\boldsymbol{x}$
  - $\psi(\lambda_1, \lambda_2)$: cost function (increasing)
  - $J_K(\nabla \boldsymbol{u}) = K * \sum_{i=1}^{N} \nabla u_i \otimes \nabla u_i$: 2x2 structure tensor with:
    - eigenvalues λ_1, λ_2 , eigenvectors $\boldsymbol{\theta}_1, \boldsymbol{\theta}_2$ (depend on *K*)
- Difficulty in the theoretical analysis:

  In contrast to most variational methods, Euler-Lagrange equations not applicable here
- Theorem: we have shown that the functional minimization leads to:

$$\begin{cases} \partial u_i / \partial t = \mathrm{div}\left(D_K \nabla u_i\right), \; i = 1, .., N, \\ D_K = K * \left(2 \dfrac{\partial \psi}{\partial \lambda_1} \boldsymbol{\theta}_1 \otimes \boldsymbol{\theta}_1 + 2 \dfrac{\partial \psi}{\partial \lambda_2} \boldsymbol{\theta}_2 \otimes \boldsymbol{\theta}_2 \right) \end{cases}$$

novel general type of anisotropic diffusion

Tensor Total Variation

- 1^{st} special case of the novel generic functional:

$$E[\boldsymbol{u}] = \int_\Omega \psi\left(\,\lambda_1(J_K(\nabla \boldsymbol{u})), \lambda_2(J_K(\nabla \boldsymbol{u}))\,\right) \mathrm{d}\boldsymbol{x}$$

with $\psi(\lambda_1, \lambda_2) = \sqrt{\lambda_1} + \sqrt{\lambda_2}$

- Steepest descent (applying the proved theorem):

$$\frac{\partial u_i}{\partial t} = \operatorname{div}\left(\left[K * \left(\frac{1}{\sqrt{\lambda_1}}\boldsymbol{\theta}_1 \otimes \boldsymbol{\theta}_1 + \frac{1}{\sqrt{\lambda_2}}\boldsymbol{\theta}_2 \otimes \boldsymbol{\theta}_2\right)\right] \nabla u_i\right)\ ,\ i = 1, .., N$$

- Classic TV: special sub-case with:

  N=1(graylevel images) and $K = \delta(\boldsymbol{x})$

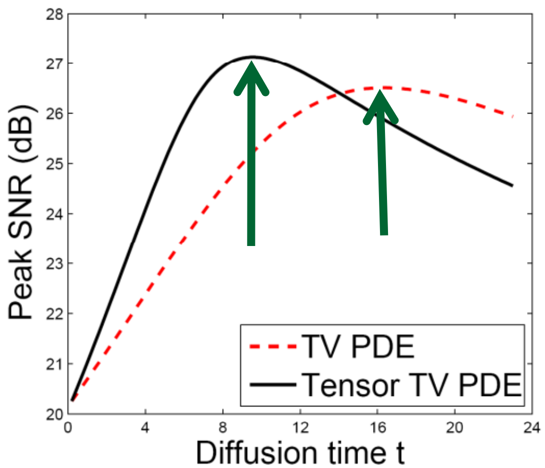
- The proposed method:
  - adaptively smooths the image
  - combines the advantages of TV minimization and tensor-based diffusion methods

Tensor Total Variation: Example (1)

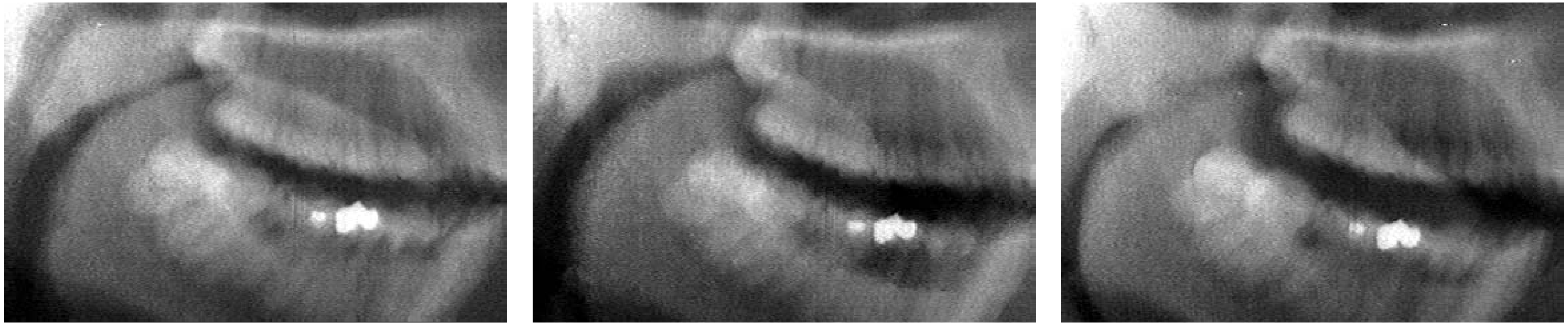
(a) Noisy Input
(PSNR=20 dB)

(b) TV PDE
(PSNR=26.5 dB, t=16.4)

(c) Tensor TV PDE
(PSNR=27.1 dB, t=9.6)



Tensor Total Variation: Example (2)



Input sequence

Output sequence

Denoising of an X-ray video of a speaker's vocal tract

Generalized Beltrami Flow

- 2^{nd} special case of the novel generic functional:

$$E[\boldsymbol{u}] = \int_\Omega \psi\left(\,\lambda_1(J_K(\nabla \boldsymbol{u})), \lambda_2(J_K(\nabla \boldsymbol{u}))\,\right) \mathrm{d}\boldsymbol{x}$$

with $\psi(\lambda_1, \lambda_2) = \sqrt{(\alpha^2 + \lambda_1)\,(\alpha^2 + \lambda_2)}$

  - Steepest descent (applying the proved theorem):

$$\frac{\partial u_i}{\partial t} = \operatorname{div}\left(\left[K * \left(\sqrt{\frac{\alpha^2 + \lambda_2}{\alpha^2 + \lambda_1}}\boldsymbol{\theta}_1 \otimes \boldsymbol{\theta}_1 + \sqrt{\frac{\alpha^2 + \lambda_1}{\alpha^2 + \lambda_2}}\boldsymbol{\theta}_2 \otimes \boldsymbol{\theta}_2\right)\right] \nabla u_i\right)$$

  - Classic Beltrami flow [Sochen et. al, IEEE T-IP 98]: special sub-case with $K = \delta(\boldsymbol{x})$ and minimization in the space of embeddings

(a) Noisy Input (PSNR=20 dB) | (b) Beltrami Flow (PSNR=23.4 dB) | (c) Gener. Beltrami Flow (PSNR=24.0 dB)

Other Interesting Special Cases

- Other special cases of the novel generic functional:

  $E[\boldsymbol{u}] = \int_\Omega \psi\,(\,\lambda_1(J_K(\nabla \boldsymbol{u})), \lambda_2(J_K(\nabla \boldsymbol{u}))\,)\,\mathrm{d}\boldsymbol{x}$ with:

  - $\psi(\lambda_1, \lambda_2) = \phi(\lambda_1 + \lambda_2)$: Steepest descent:

    $$\partial u_i / \partial t = \mathrm{div}\left(2\left[K * \varphi'(K * \|\nabla \boldsymbol{u}\|^2)\right] \nabla u_i\right)$$

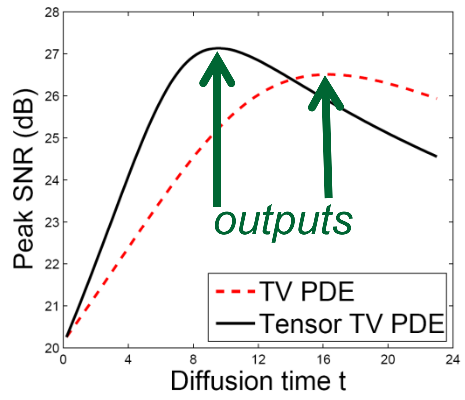
    →novel regularization of the Perona-Malik model, alternative to the classic CLMC [Catte et al, siamJNA'92]

    →regularization of Sapiro's Vectorial TV: $\psi = \sqrt{\lambda_1 + \lambda_2}$

  - $K = \delta(\boldsymbol{x})$ (no regularizing convolution):
    - Studied in [Blomgren & Chan T-IP'98, Tschumperlé & Deriche, T-PAMI'05]
    - The corresponding diffusion is anisotropic only if the image channels are $N \geq 2$
    - No incorporation of neighborhood info

Denoising Experiments: Framework

- Experimental Framework
  - take a noise-free reference image
  - add gaussian noise
  - input in the compared diffusion methods
  - compute PSNR during each PDE flow and output the image with the maximum PSNR



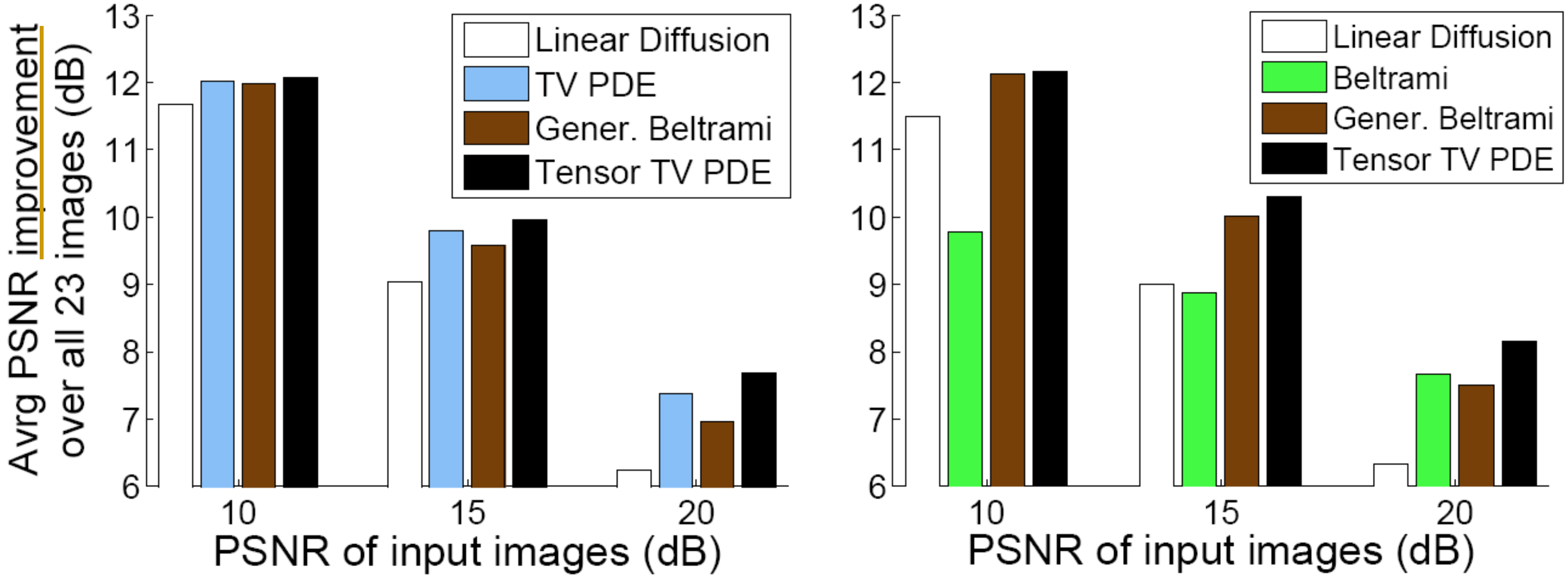
- This framework has been repeated for reference images from a dataset of *CIPR*: www.cipr.rpi.edu/resource/stills/kodak.html

23 natural images of size 768 x 512 pixels

…

- Both graylevel & color versions of images have been used

Denoising Experiments: Performance Measures

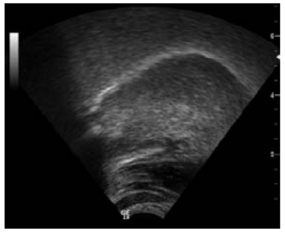
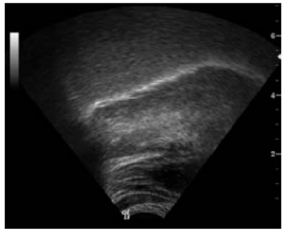
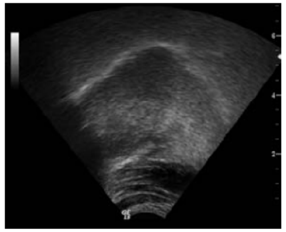


(a) Graylevel images (b) Color images

Contents

Tongue Tracking in Ultrasound (US) Images

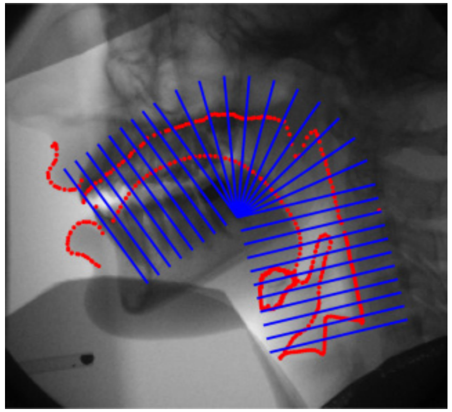
[Roussos,Maragos, ICIP 2010], [Aron,Roussos et. al EUSIPCO 08]

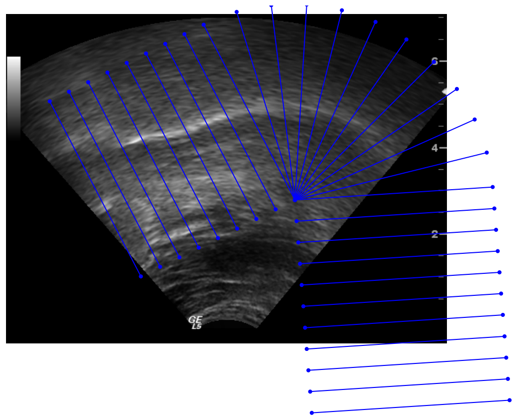
- Especially useful for cases of large databases of ultrasound videos
- Difficulties
  - high amounts of speckle noise
  - weak visibility of the tongue contour
  - the tongue is highly and quickly deforming
  - landmark points cannot be manually specified
- We proposed a novel tracking method that:
  - is built on a variant of Active Appearance Models (AAM)
  - incorporates prior information about the tongue shape variation

Tongue Tracking: Data Exploitation

- Acquired data from the same speaker: **Ultrasound videos, EM sensors, MRI, X-ray videos**

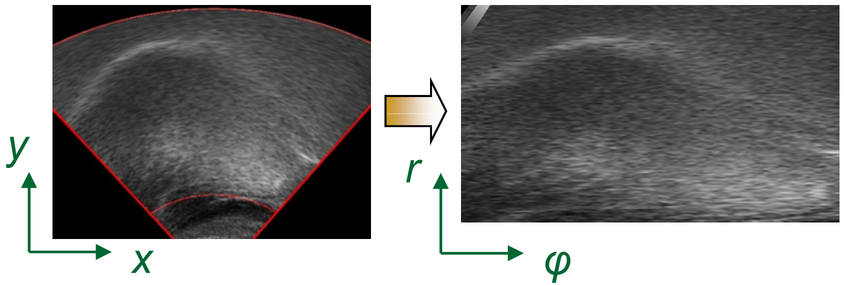


- Exploitation of X-rays to model the tongue shape variation
  - Use of a Vocal Tract (VT) Grid for the tongue shape representation [Maeda, BookChap'90]

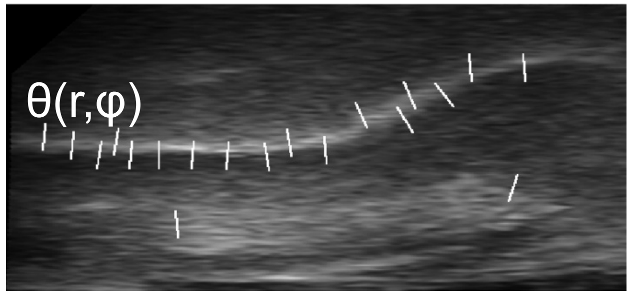


- Estimation of the VT grid's pose at every ultrasound frame, using EM sensors and MRI data

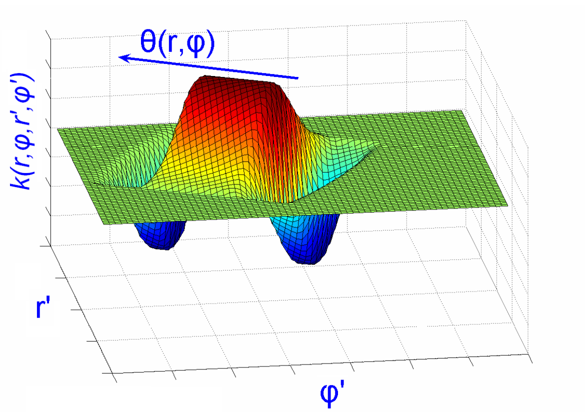
Filtering of Ultrasound Frames: Method's Steps



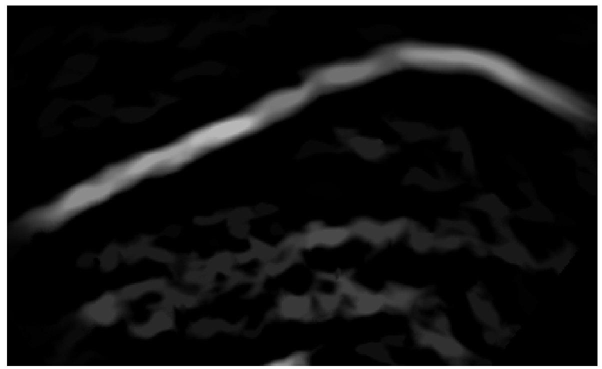
1. Convert u(x,y) to u(r,φ)



2. Robust estimation of the *orientation* *θ(r,φ)* normal to edges

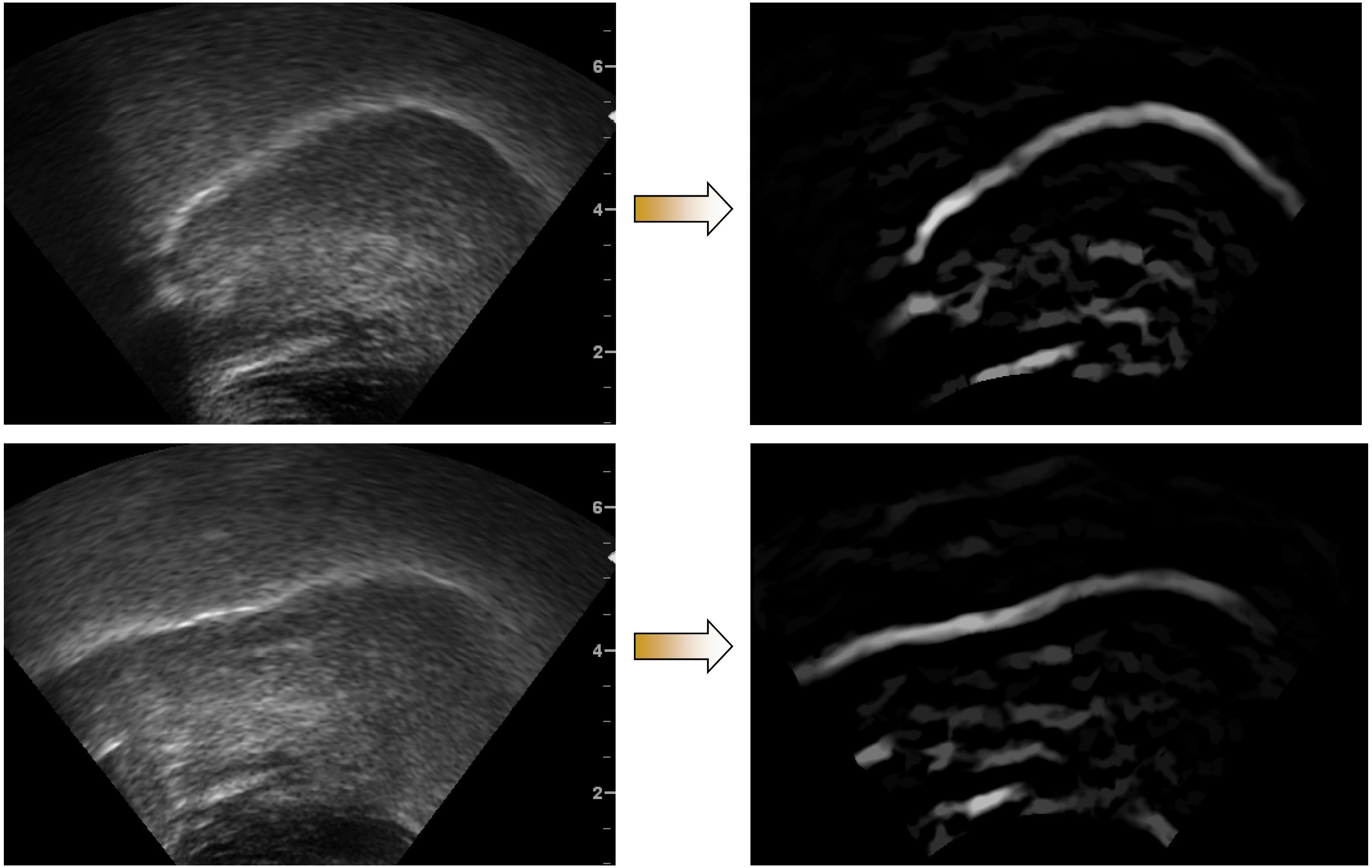


3. Correlate u(r,φ) with a varying kernel k(r,φ;r',φ'), aligned to θ(r,φ)



4. Keep only values>0, convert back to (x,y) coords. & apply *Area Opening*

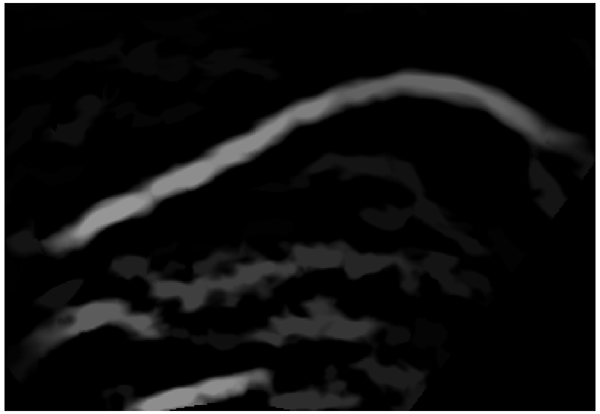
Filtering of Ultrasound Frames: Examples



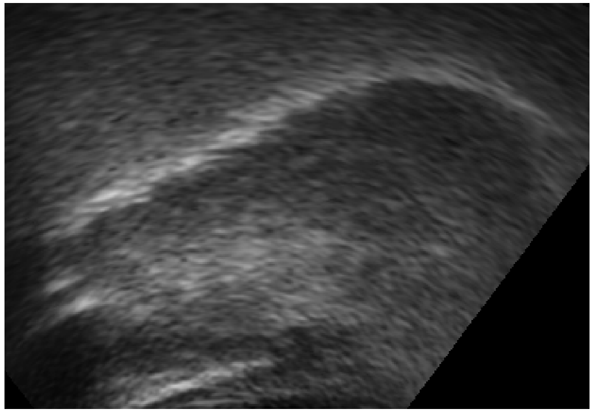
Input US frames

Filtered US frames

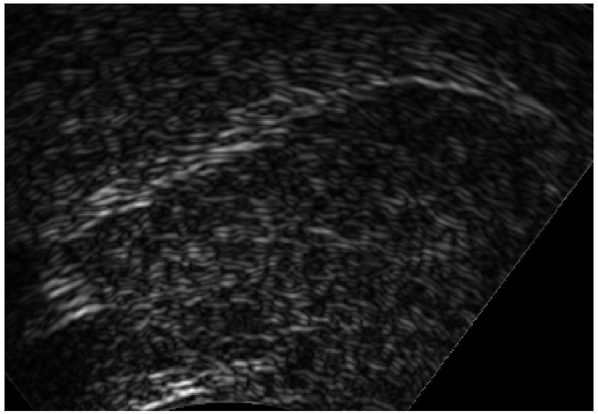
Filtering of US Frames: Comparisons



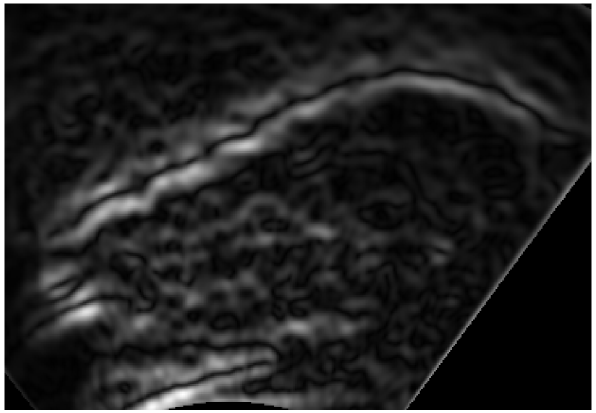
Our Filtering [Eusipco'08]



Input US frame $u(x,y)$

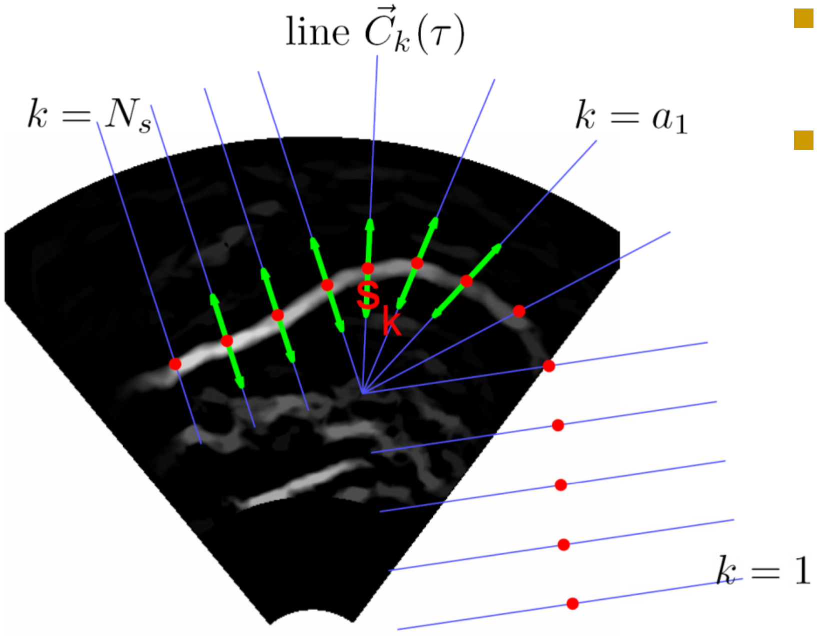


Classic Edge Strength
$|\nabla G_\sigma * u(x,y)|$, σ=1



Classic Edge Strength
$|\nabla G_\sigma * u(x,y)|$, σ=4

Tongue Appearance Representation



- Shape $\boldsymbol{s} = [s_1, .., s_{N_s}]^T$
- Texture $\boldsymbol{g}(\boldsymbol{s}) = \Big[\underbrace{[u_{a_1}(s_{a_1}+t)]^T_{t\in W}}_{1\times N_W} \cdots \underbrace{[u_{a_{N_a}}(s_{a_{N_a}}+t)]^T_{t\in W}}_{1\times N_W}\Big]^T$
  - only the *texture-active grid lines* G_{act} are used for texture
  - $W = \{-d, -d+1, .., d\} \cdot \delta\ell$: *sampling window*
  - $u_k(\tau) = u(\vec{C}_k(\tau))$: restriction of the image to grid line k

- Differences from classic AAMs
  - Various modifications to exploit application-specific properties
  - Reduced complexity of the appearance representation & model
  - Lighter optimization problem for the model fitting

Modeling Appearance Variation

- Shape model

$$\boldsymbol{s} \approx \boldsymbol{s_0} + \mathrm{Q_s}\boldsymbol{b}$$

  - b: normalized shape parameters vector with $\mathrm{p}\,(\boldsymbol{b})=\mathcal{N}(\boldsymbol{b}|\mathbf{0},\mathrm{I}_{N_b})$
  - Principal Component Analysis (PCA) to learn $\boldsymbol{s_0}$, $\mathrm{Q_s}$
    - Training vectors from manually annotated tongue contours on 700 X-ray frames

- Texture model

$$\boldsymbol{g} = \boldsymbol{g_0} + \mathrm{Q_g}\boldsymbol{\lambda} + \boldsymbol{\varepsilon}$$

  - $\boldsymbol{\lambda}$: texture parameters with $\mathrm{p}\,(\boldsymbol{\lambda})=\mathcal{N}(\boldsymbol{\lambda}|\mathbf{0},\mathrm{I}_{N_\lambda})$
  - $\boldsymbol{\varepsilon}$: texture reconstruction error with :

$$\mathrm{p}\,(\boldsymbol{\varepsilon})=\mathcal{N}(\boldsymbol{\varepsilon}|\mathbf{0},\Sigma_{\boldsymbol{\varepsilon}})\,,\quad \Sigma_{\boldsymbol{\varepsilon}}=\widetilde{\mathrm{Q}}_{\mathrm{g}}\mathrm{diag}(\rho_1,..,\rho_{N_g})\widetilde{\mathrm{Q}}_{\mathrm{g}}^T$$

  - Training of the model
    - Manual annotations at 400 US frames. This training set is divided into 2 subsets T1 and T2
    - Subset T1 is used to learn $\boldsymbol{g_0}$ and $\mathrm{Q_g}$ using PCA
    - Subset T2 is used to learn the optimum parameters $\rho_1,..,\rho_{N_g}$

Tracking via Model Fitting

- Model fitting in every ultrasound frame
- *Maximum a posteriori (MAP)* estimation of parameters $\mathbf{b}$ and $\boldsymbol{\lambda}$ by maximizing:

$$\mathrm{p}\left(\boldsymbol{b},\boldsymbol{\lambda}|u(x,y)\right) \propto \mathrm{p}\left(u|\boldsymbol{b},\boldsymbol{\lambda}\right)\mathrm{p}\left(\boldsymbol{b},\boldsymbol{\lambda}\right) = \mathrm{p}\left(\boldsymbol{\varepsilon}\right)\mathrm{p}\left(\boldsymbol{b}\right)\mathrm{p}\left(\boldsymbol{\lambda}\right)$$

$$\boldsymbol{\varepsilon} = \boldsymbol{g}(\boldsymbol{s}(\boldsymbol{b})) - \boldsymbol{g_0} - \mathrm{Q_g}\boldsymbol{\lambda}$$

- Equivalently: minimization of the energy:

$$E(\boldsymbol{b},\boldsymbol{\lambda}) = -\ln\mathrm{p}\left(\boldsymbol{b},\boldsymbol{\lambda}|u\right) = C + \tfrac{1}{2}\left\{\|\boldsymbol{b}\|^2 + \|\boldsymbol{\lambda}\|^2 + \boldsymbol{\varepsilon}^T\Sigma_{\boldsymbol{\varepsilon}}^{-1}\boldsymbol{\varepsilon}\right\}$$

- Gradients of the energy:

$$\nabla_{\boldsymbol{b}}E = \boldsymbol{b} + \mathrm{Q_s}^T\left(\partial\boldsymbol{g}/\partial\boldsymbol{s}\right)^T\Sigma_{\boldsymbol{\varepsilon}}^{-1}\boldsymbol{\varepsilon}$$

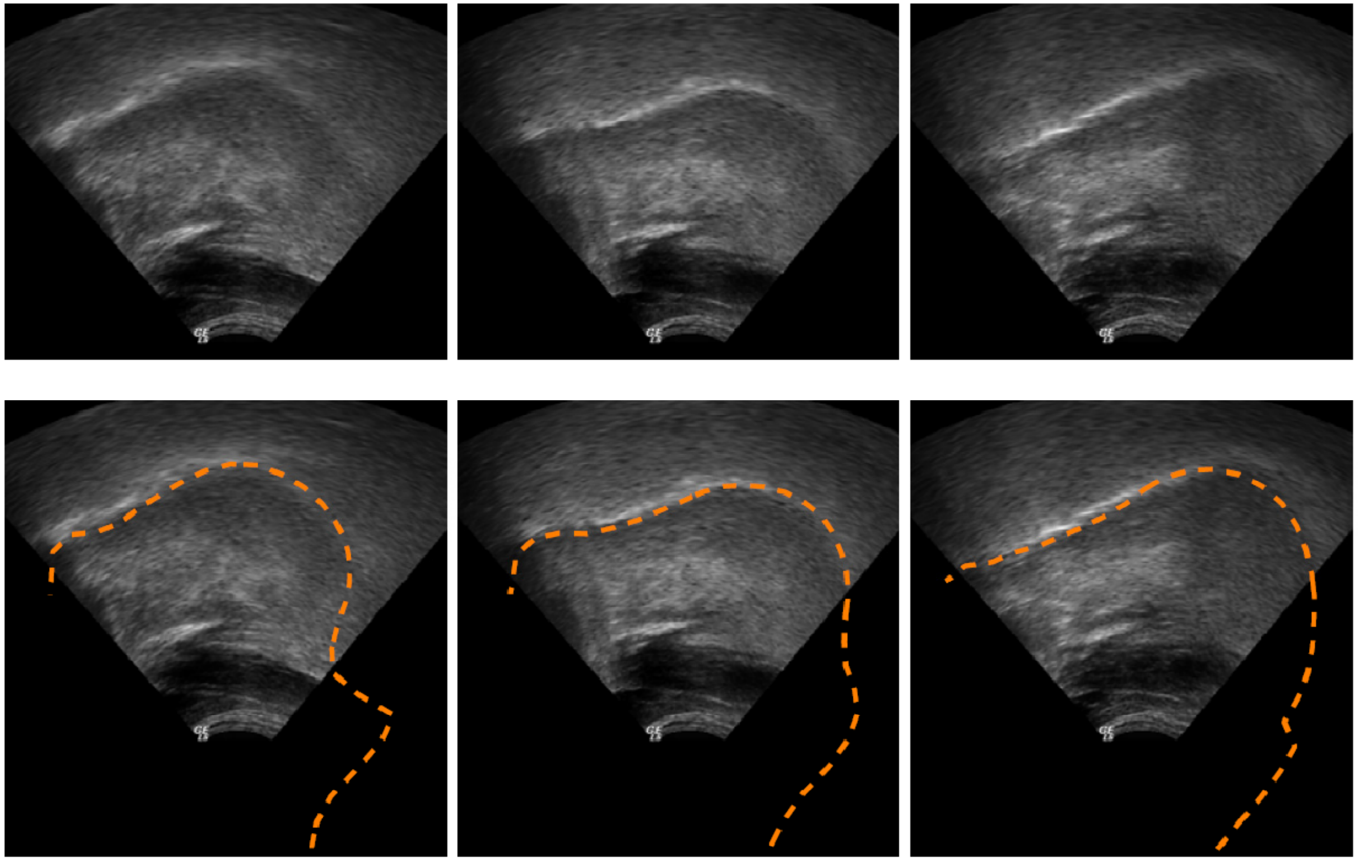
$$\nabla_{\boldsymbol{\lambda}}E = \boldsymbol{\lambda} - \mathrm{Q_g}^T\Sigma_{\boldsymbol{\varepsilon}}^{-1}\boldsymbol{\varepsilon}$$

where: $$\frac{\partial\boldsymbol{g}}{\partial s_k} = \begin{cases} [0\cdots\cdots 0]^T, & \text{if } k \notin G_{act} \\ [\,\underbrace{0\cdots 0}_{(k-1)N_W}\;[u_k'(s_k+t)]_{t\in W}^T\;\underbrace{0\cdots 0}_{(N_s-k)N_W}\,]^T, & \text{if } k \in G_{act} \end{cases}$$

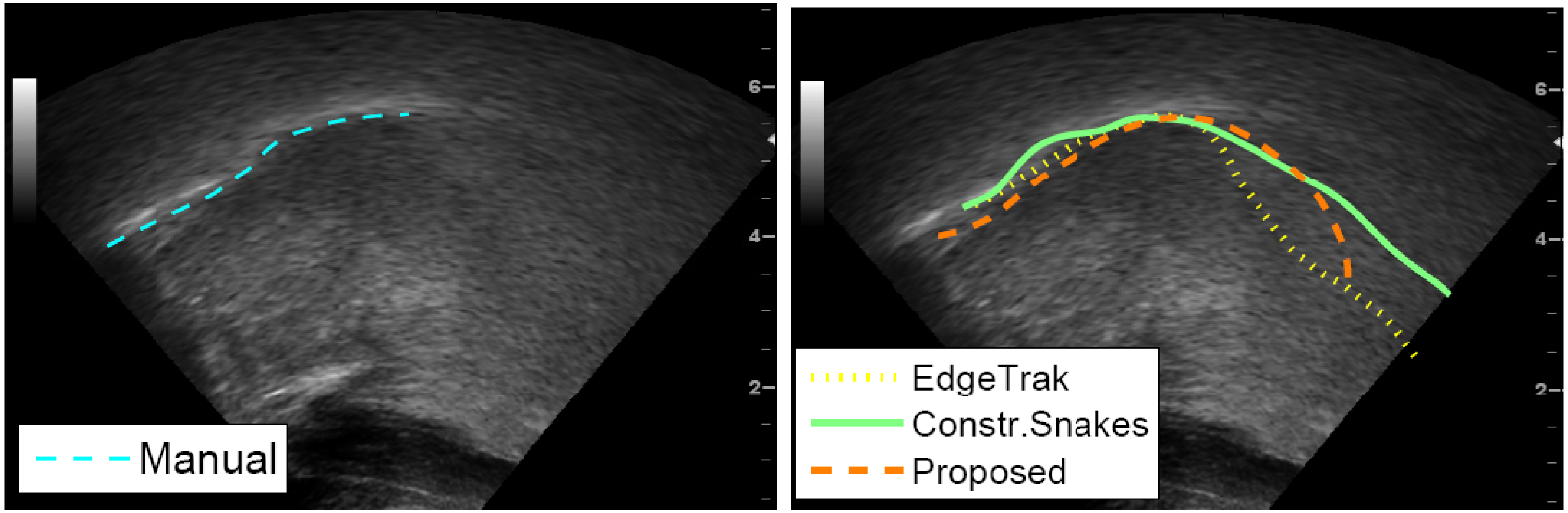
- Optimization algorithm:
  - Gradient descent
  - Parameters initialization:
    - $\boldsymbol{b_0}$: from previous frame result
    - $\boldsymbol{\lambda_0}$: maximization of the posterior $\mathrm{p}\left(\boldsymbol{\lambda}|\boldsymbol{g}(\boldsymbol{s}(\boldsymbol{b_0}))\right)$

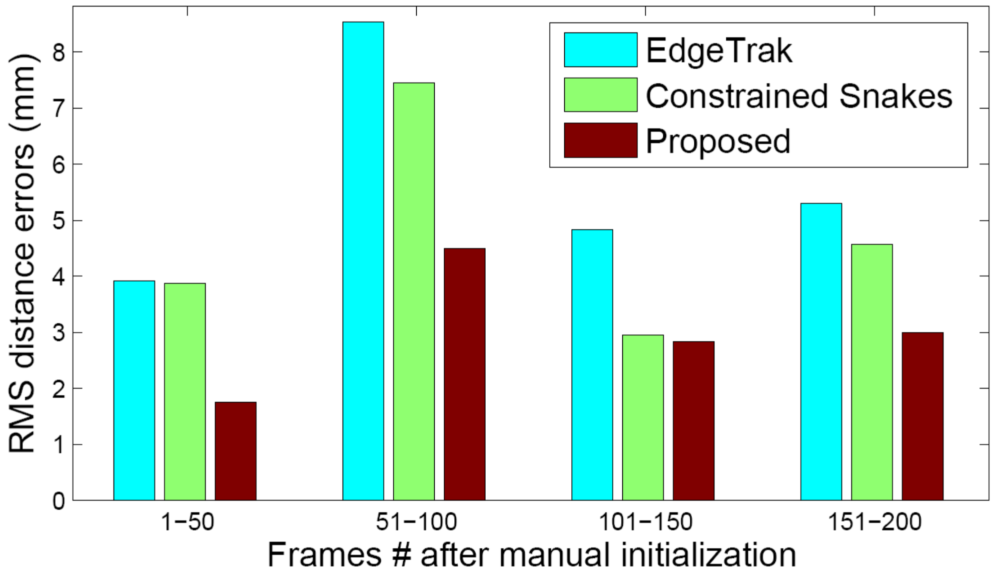
Tongue Tracking Results of the Proposed Method

	Dimensionality of original vector	Number of model parameters	Variance explained (% of the total)
Shape	30	6	96%
Texture	1215	35	93%



Comparisons with other methods





$$e_d = \sqrt{(d_{om}^2 + d_{mo}^2)/2}$$

Contents

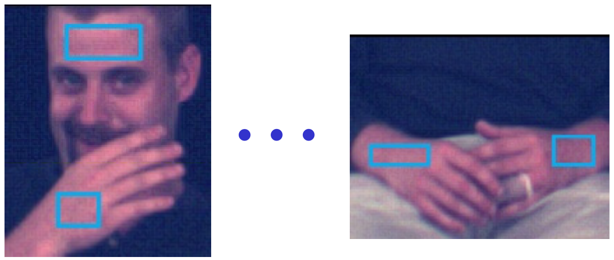
- Introduction
  - PDEs & Shape Models in Computer Vision
  - Applications
  - Research Contributions
- Nonlinear Diffusion for Image Interpolation
- Variational Frameworks for Tensor-based Diffusion
- Tongue Tracking with Active Appearance Models
- Handshape Modeling for Sign Language
- Conclusions

Handshape Modeling for Sign Language

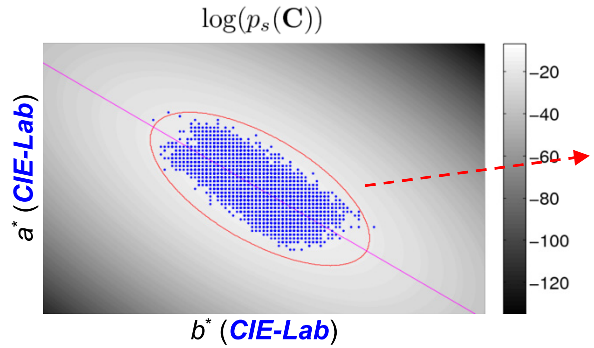
- Analysis of videos of continuous signing
- Goals
  - localization & tracking of the signer's hands+head
  - extraction of features that reliably describe the pose and configuration of the signer's hands
- Ultimate goal
  - automatic sign language recognition

Initial Head & Hands Tracking (1/2)

- Skin color modeling



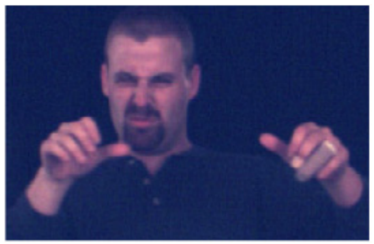
training samples



color thresholding for the skin mask S_0

fitted probability density function

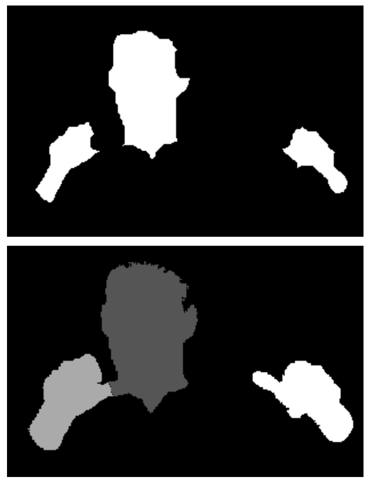
- Morphological processing of the skin mask



input

skin mask S_0

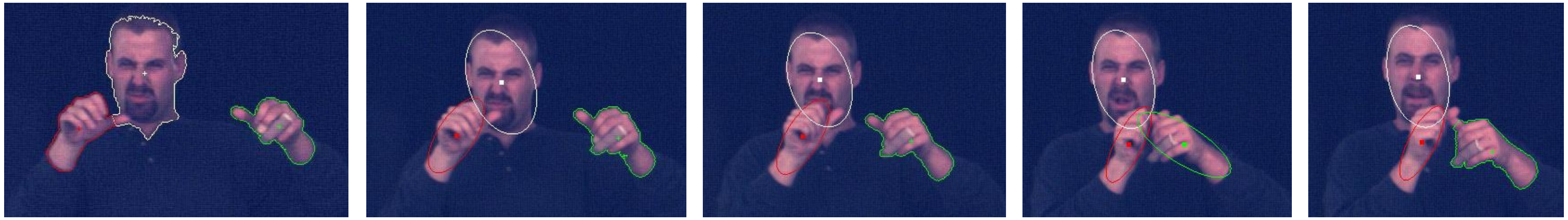
refinement of S_0
- generalized hole filling
- area opening



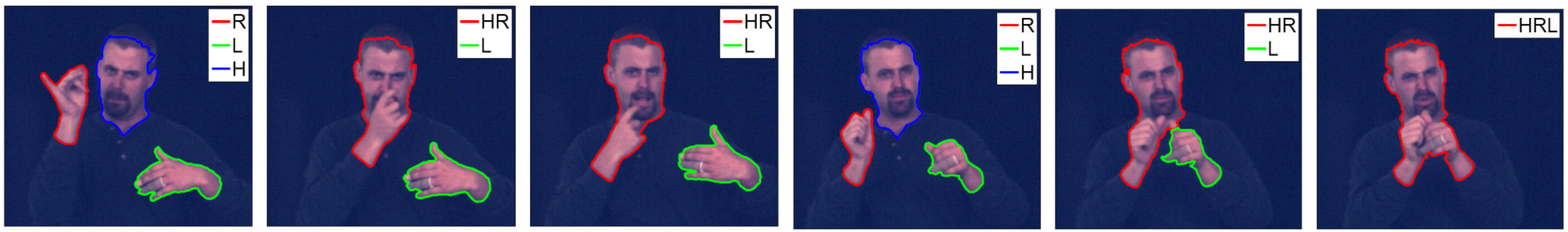
segmentation
- connected components
- competitive rec. opening

Initial Head & Hands Tracking (2/2)

- Main parts of tracking:
  - fwd-bkwd prediction,
  - template matching,
  - ellipses fitting,
  - probabilistic constraints



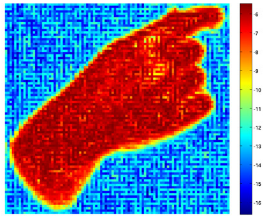
- Output: set of skin region masks + label(s) assignment {H,R,L} to each mask

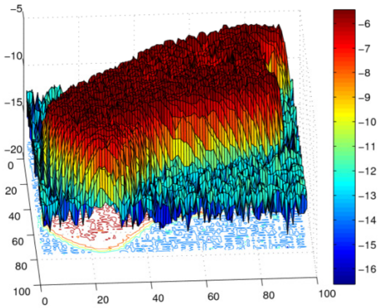


Shape-Appearance Model: Representation

$\boldsymbol{I}(\boldsymbol{x})$

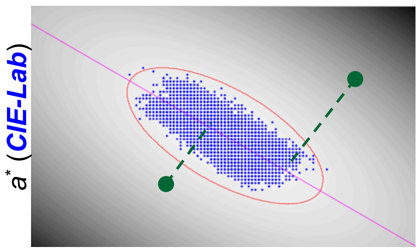
initial cropped hand image

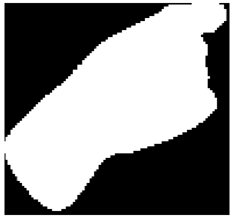




$g(\boldsymbol{I}(\boldsymbol{x}))$

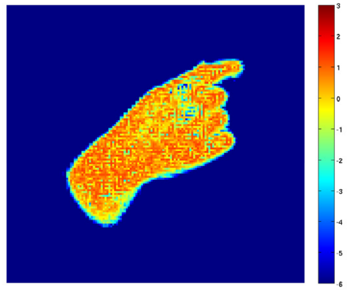
projection of each pixel's color (a^*,b^*) on the principal axis of gaussian $p_s(a^*,b^*)$

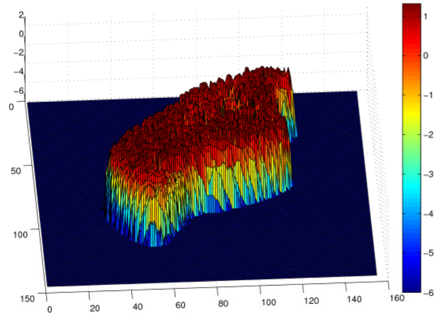




M

skin mask





Shape-Appearance Image

$$f(\boldsymbol{x}) = \begin{cases} g(\boldsymbol{I}(\boldsymbol{x})), & \text{if } \boldsymbol{x} \in M \\ -c_b & \text{else} \end{cases}$$

constant for the balance between shape and appearance

Shape-Appearance Generative Modeling

Shape-Appearance image of the hand

base image

eigenimages

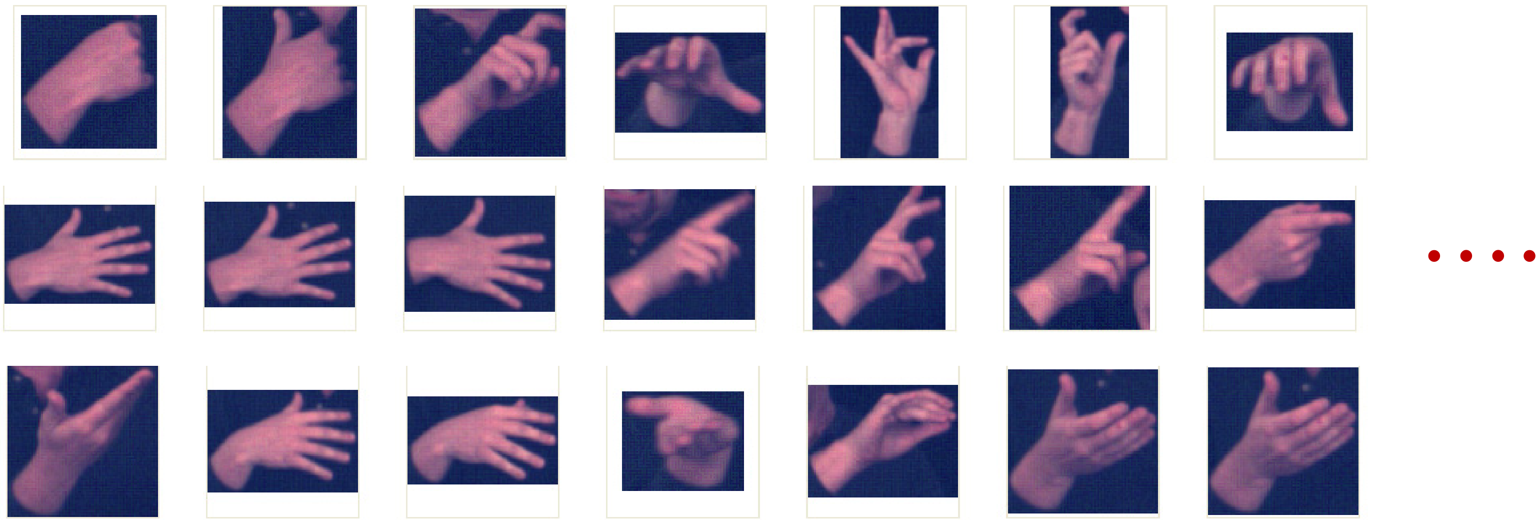
$$f(W_{\boldsymbol{p}}(\boldsymbol{x})) \approx A_0(\boldsymbol{x}) + \sum_{i=1}^{N_c} \lambda_i A_i(\boldsymbol{x})$$

2D affine transform

$$W_{\boldsymbol{p}}(x,y) = \begin{pmatrix} 1+p_1 & p_3 & p_5 \\ p_2 & 1+p_4 & p_6 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

Shape-Appearance Model: Learning of $A_i(\boldsymbol{x})$ (1/2)

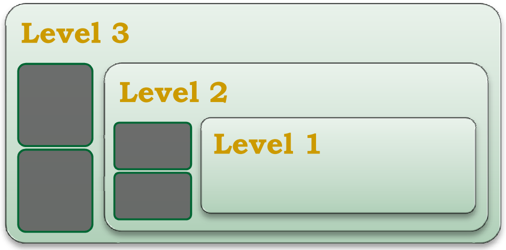
- Training set
  - extraction of hand images without occlusions from training videos
  - random selection of 500 such images



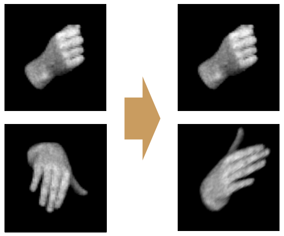
- Affine alignment of the training set

  …….

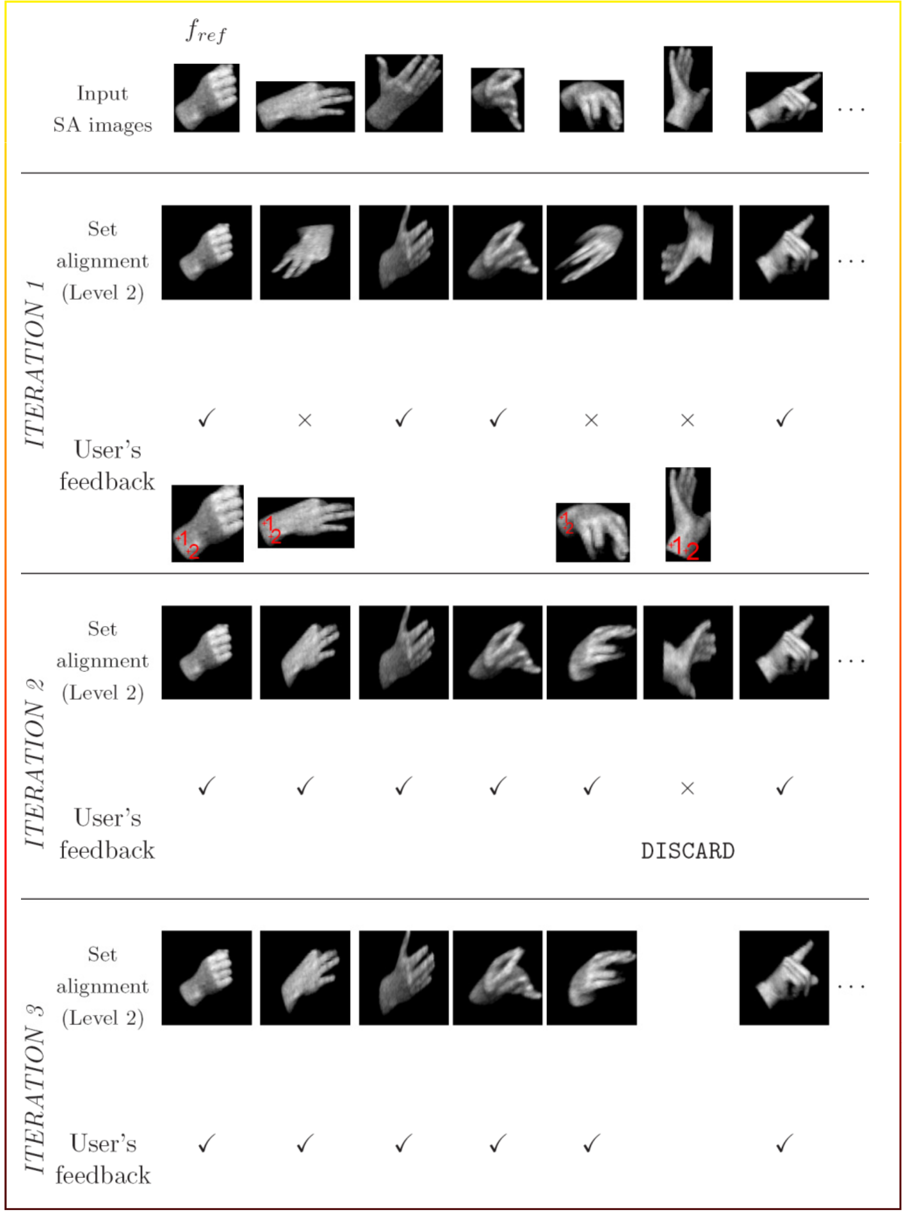
Affine alignment of the training set



- Level 1: 1-1 alignment
  - Use of the Inverse-Compositional Algorithm [Gross,Matthews,Baker, IVC'05]

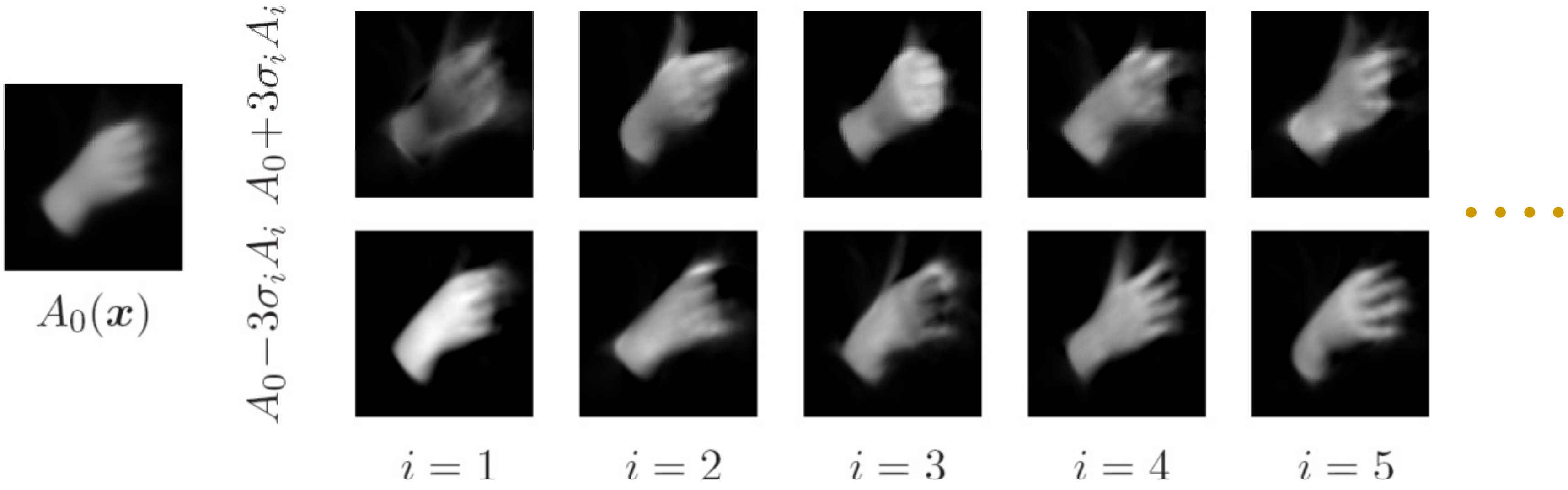


- Level 2: Training set alignment
  - Generalization of *Procrustes Analysis* [Cootes,Taylor, TecRep'04]
- Level 3: Iterative manual feedback



Shape-Appearance Model: Learning of $A_i(\boldsymbol{x})$ (2/2)

- Principal Component Analysis (PCA) of the affinely aligned training set
- Keep only 35 eigenimages $A_i(\boldsymbol{x})$, which explain 78% of the variance
- Affine alignment offers significant reduction on the variability of hand SA images



Shape-Appearance Model: Fitting

- Outputs: robust hand tracking & hand feature extraction
- Find optimum parameters $\boldsymbol{\lambda}, \boldsymbol{p}$ that minimize the regularized energy:

$$E(\boldsymbol{\lambda},\boldsymbol{p}) = E_{rec}(\boldsymbol{\lambda},\boldsymbol{p}) + w_S E_S(\boldsymbol{\lambda},\boldsymbol{p}) + w_D E_D(\boldsymbol{\lambda},\boldsymbol{p})$$

$$E_{rec}(\boldsymbol{\lambda},\boldsymbol{p}) = \frac{1}{N_M}\sum_{\boldsymbol{x}}\left\{A_0(\boldsymbol{x}) + \sum_{i=1}^{N_c}\lambda_i A_i(\boldsymbol{x}) - f(W_{\boldsymbol{p}}(\boldsymbol{x}))\right\}^2$$ *mean square reconstruction error*

$$E_S(\boldsymbol{\lambda},\boldsymbol{p}) = \frac{1}{N_c}\|\boldsymbol{\lambda}-\boldsymbol{\lambda}_0\|^2_{\Sigma_{\boldsymbol{\lambda}}} + \frac{1}{N_p}\|\boldsymbol{p}-\boldsymbol{p}_0\|^2_{\Sigma_{\boldsymbol{p}}}$$ *static priors term*

$$E_D(\boldsymbol{\lambda},\boldsymbol{p}) = \frac{1}{N_c}\|\boldsymbol{\lambda}-\boldsymbol{\lambda}^e\|^2_{\Sigma_{\boldsymbol{\epsilon_\lambda}}} + \frac{1}{N_p}\|\boldsymbol{p}-\boldsymbol{p}^e\|^2_{\Sigma_{\boldsymbol{\epsilon_p}}}$$ *dynamic priors term*

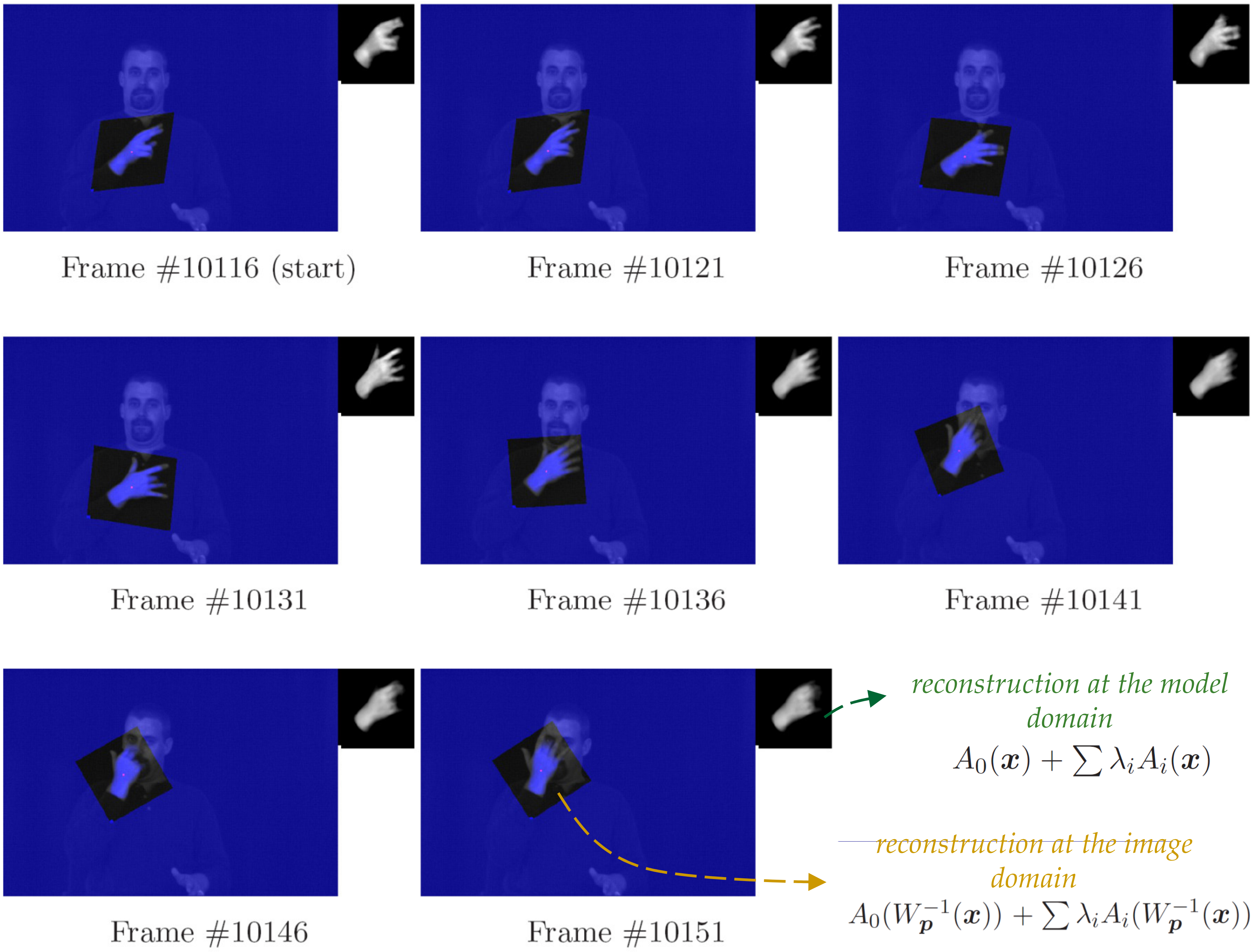
  - Dynamical Models for Linear Prediction:

$$\boldsymbol{\lambda}^e[n] = \sum_{\nu\in W(K)} \mathrm{A}_{K,\nu}\,\boldsymbol{\lambda}[n-\nu] \quad , \quad \widetilde{\boldsymbol{p}}^e[n] = \sum_{\nu\in W(K)} \mathrm{B}_{K,\nu}\,\widetilde{\boldsymbol{p}}[n-\nu]$$

  - $\Sigma_{\boldsymbol{\lambda}}, \Sigma_{\boldsymbol{p}}, \Sigma_{\boldsymbol{\epsilon_\lambda}}, \Sigma_{\boldsymbol{\epsilon_p}}$: covariance matrices
  - $\boldsymbol{\lambda}_0, \boldsymbol{p}_0$: mean values

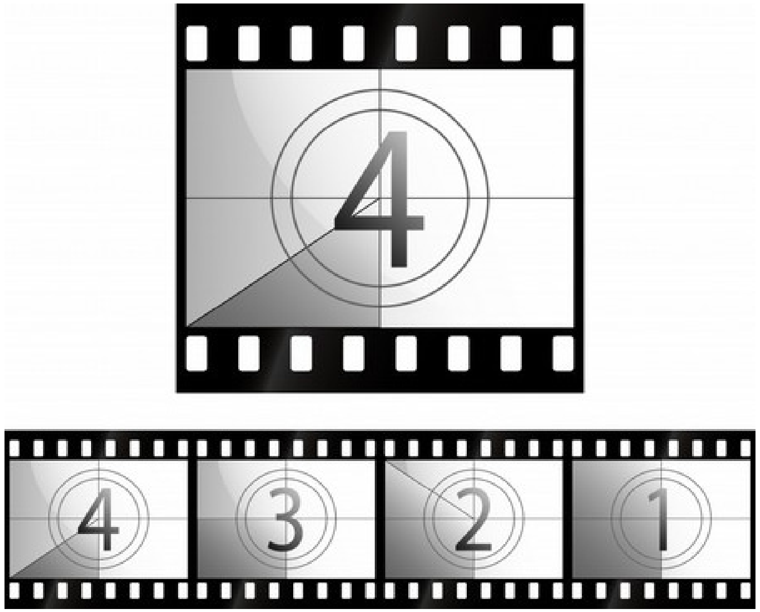
learned in a training phase using non-occluded frames

- Algorithm: Simultaneous Inverse-Compositional [Baker et al, TecRep'04]



Frame #10116 (start)
Frame #10121
Frame #10126
Frame #10131
Frame #10136
Frame #10141
Frame #10146
Frame #10151
reconstruction at the model domain
$A_0(\boldsymbol{x}) + \sum \lambda_i A_i(\boldsymbol{x})$
reconstruction at the image domain
$A_0(W_{\boldsymbol{p}}^{-1}(\boldsymbol{x})) + \sum \lambda_i A_i(W_{\boldsymbol{p}}^{-1}(\boldsymbol{x}))$

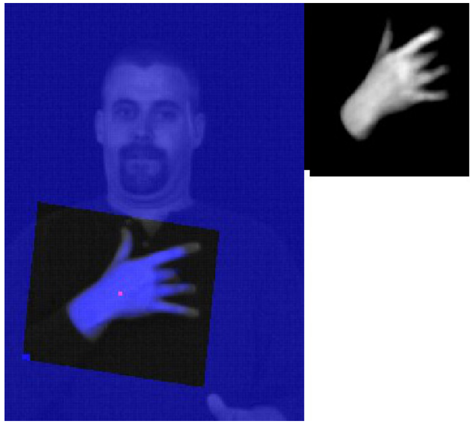
Shape-Appearance Model Fitting: Example

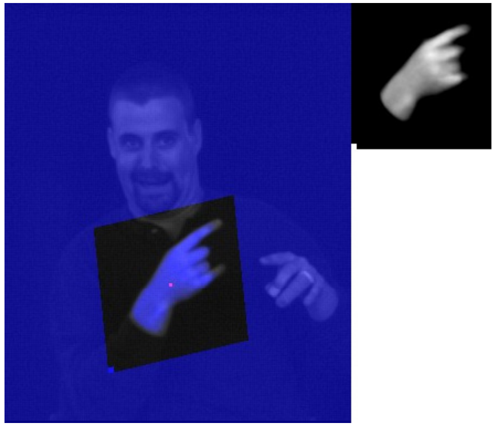


(Video)

Hand Feature Extraction

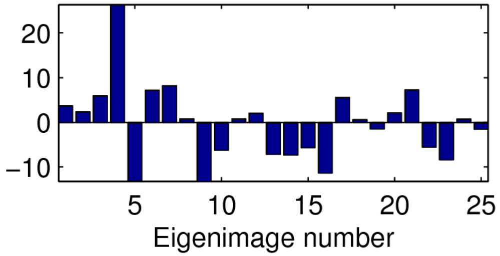
input frames
+
model fitting

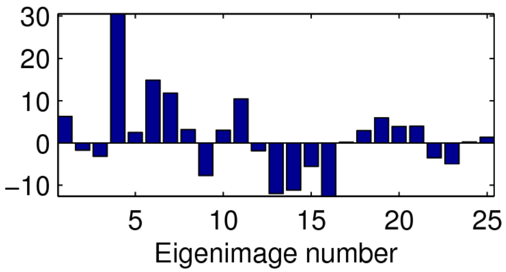




weights of the eigenimages

$\boldsymbol{\lambda}$





handshape features

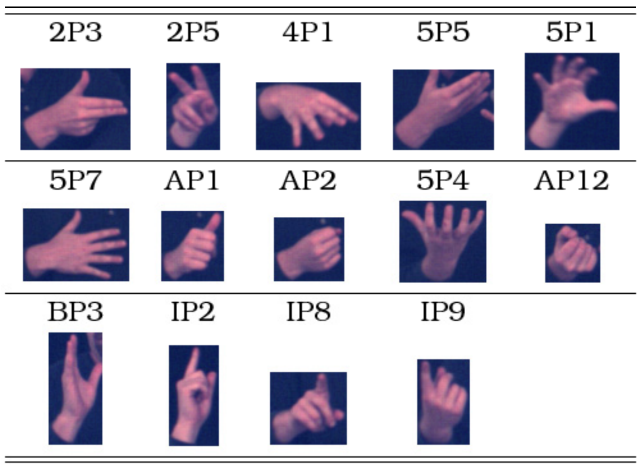
parameters of the affine transform

$\boldsymbol{p}$

$$\begin{pmatrix} -0.0061 & -0.0944 & -78.1642 \\ 0.1033 & 0.0552 & -128.2917 \end{pmatrix}$$

$$\begin{pmatrix} -0.0185 & -0.0278 & -95.0785 \\ 0.1260 & -0.0499 & -139.1400 \end{pmatrix}$$

Handshape Classification Experiments

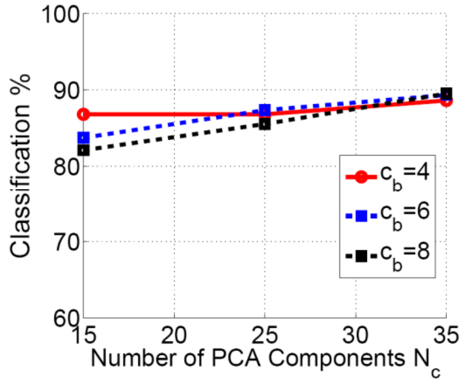


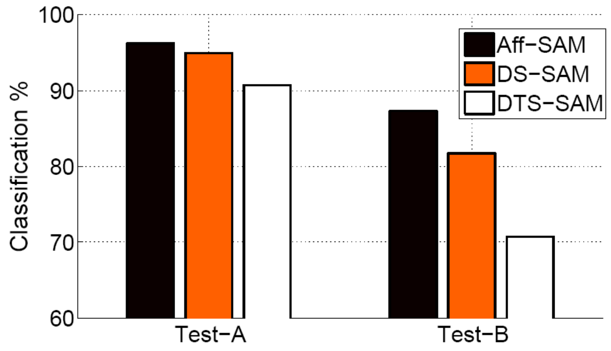
Classes for Test-A

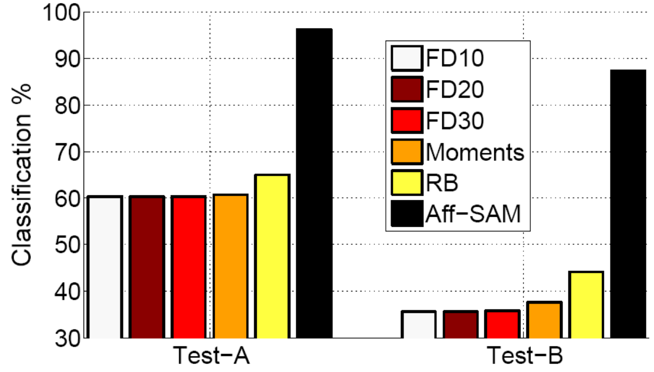
Classes for Test-B

Handshape Classification Results

For all methods, classification is done using 1-mixture GMM per class & maximum likelihood







Proposed method for Test-B: Variation of main parameters:

- # of PCA components
- Cb: Background constant for SA images

Comparison of the **proposed method** with its **simplified versions**:

- **Aff-SAM**: Affine Shape - Appearance Modeling (proposed)
- **DS-SAM**: Direct Similarity Shape-Appearance Modeling
- **DTS-SAM**: Direct Translation + Scale Shape-Appearance Modeling

Comparison of the **proposed method** with **baseline methods**:

- **FD**: Fourier Descriptors with 10,20,30 coefficients
- **Moments**: Hu moment invariants of hand region
- **RB**: Region-based descriptors (area, eccentricity, compactness and minor+major axis lengths)
- **Aff-SAM**: proposed

Contents

Contributions

- Novel nonlinear diffusion methods for image enhancement
  - Anisotropic diffusion-projection method for vector-valued image interpolation
  - Theoretical framework that is based on the image structure tensor and generalizes various nonlinear diffusion methods

- Design of statistical shape models for object tracking and classification
  - Statistical model for tongue tracking during speech
  - Affine-invariant modeling of handshapes during signing. Regularized hand tracking and handshape feature extraction

Publications

1. A. Roussos and P. Maragos. Reversible interpolation of vectorial images by an anisotropic diffusion-projection PDE. *International Journal of Computer Vision*, 84(2), August 2009.

2. A. Roussos, S. Theodorakis, V. Pitsikalis, and P. Maragos. Dynamic affine-invariant shape-appearance model for hand tracking and feature extraction in continuous sign language. Under preparation to be submitted to the *International Journal of Computer Vision*.

3. A. Roussos and P. Maragos. Vector-valued image interpolation by an anisotropic diffusion-projection PDE. In *Scale Space and Variational Methods in Computer Vision, First International Conference, SSVM-2007 Proceedings*, volume 4485 of *Lecture Notes in Computer Science*, pages 104–115. Springer-Verlag, 2007.

4. M. Aron, A. Roussos, M.-O. Berger, E. Kerrien, and P. Maragos. Multimodality Acquisition of Articulatory Data and Processing. In *Proceedings of the European Signal Processing Conference (EUSIPCO), Lausanne*, 2008.

5. A. Katsamanis, A. Roussos, P. Maragos, M. Aron, and M.-O. Berger. Inversion from audiovisual speech to articulatory information by exploiting multimodal data. In *International Seminar on Speech Production*, December 2008.

6. A. Roussos, A. Katsamanis, and P. Maragos. Tongue tracking in ultrasound images with active appearance models. In *Proceedings of the International Conference on Image Processing*, November 2009.

7. A. Roussos and P. Maragos. Tensor-based image diffusions derived from generalizations of the total variation and beltrami functionals. In *Proceedings of the International Conference on Image Processing*, September 2010.

8. A. Roussos, S. Theodorakis, V. Pitsikalis, and P. Maragos. Affine-invariant modeling of shape-appearance images applied on sign language handshape classification. In *Proc. Int'l Conf. on Image Processing*, September 2010.

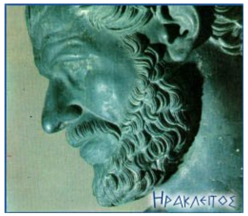
9. A. Roussos, S. Theodorakis, V. Pitsikalis, and P. Maragos. Hand tracking and affine shape-appearance handshape sub-units in continuous sign language recognition. In *Proc. of Workshop on Sign, Gesture and Activity, 11th ECCV*, September 2010.

Thank you for your attention!

Questions;

Computer Vision, Speech Communication &
Signal Processing Group: cvsp.cs.ntua.gr

Personal website: www.troussos.gr



Τα πάντα ῥεῖ
Everything flows