Computer Vision, Speech Communication and Signal Processing Group
School of Electrical and Computer Engineering
National Technical University of Athens, Greece
URL: http://cvsp.cs.ntua.gr

Vector-Valued Image Interpolation by an Anisotropic Diffusion-Projection PDE

Anastasios Roussos and Petros Maragos

1st International Conference on Scale Space Methods and Variational Methods in Computer Vision,
Ischia, Italy, May 30 - June 2, 2007

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*
- One of the fundamental Image Processing problems
- Often required as a pre-processing step in various Computer Vision tasks, such as:
  - Image Segmentation, Feature Detection, Object Recognition and Motion Analysis

Image Interpolation Methods

- Classic *Linear* methods:
  - Bicubic, Quadratic, Spline interpolation, etc...
  - Convolve the input samples with a kernel (lowpass filtering)
- Adaptive *Nonlinear* methods:
  - They perform a processing adapted to the *local geometric structure* of the image
  - Main motivation: reconstruct the edges without blurring them
  - Variational & PDE-based methods belong to this class

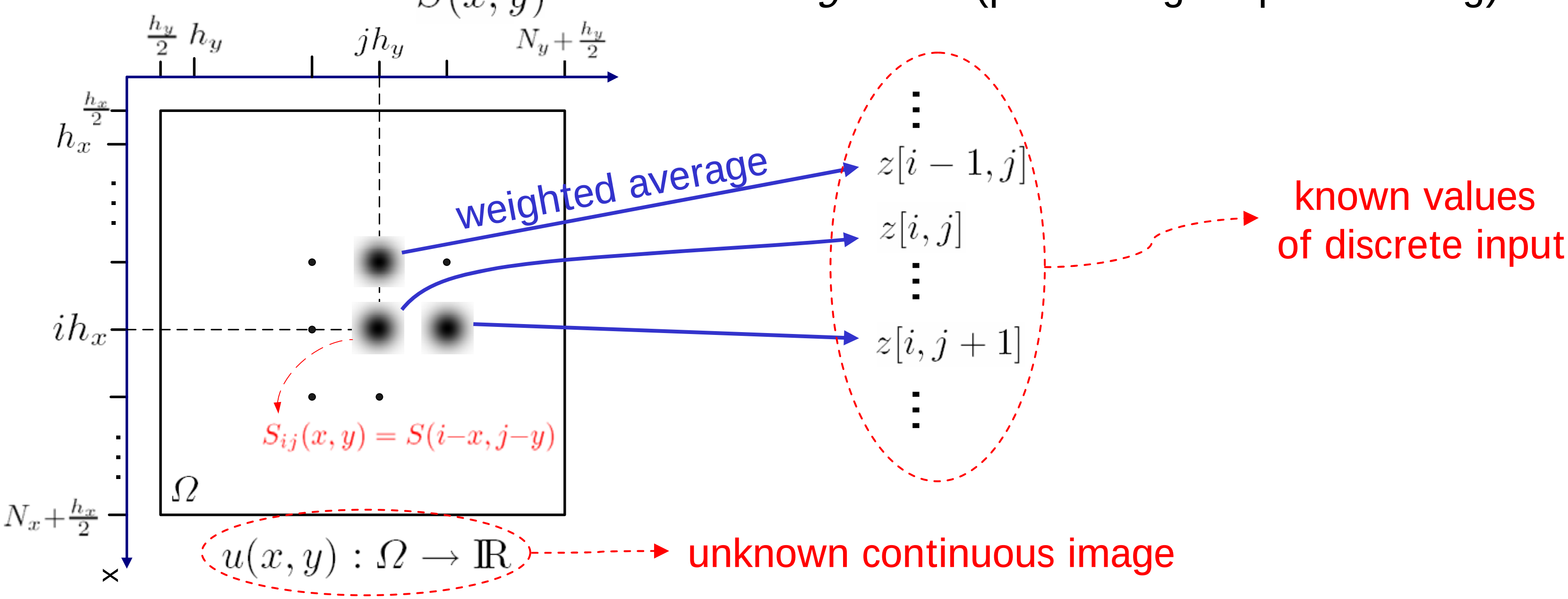
Overview of The Proposed Method

- We propose a *nonlinear* method:
  - designed for general vector-valued images
  - based on an anisotropic diffusion PDE with a projection operator
- This method:
  - *avoids most artifacts* of classic and other PDE-based methods
  - yields *improved error measures*

Reversibility Condition Approach (1)

- Ref: [Malgouyres,Guichard, *SIAM J. Num. Anal. '01*]
- Let :
  - $z[i,j] : \{1,..,N_x\} \times \{1,..,N_y\} \to \mathbb{R}$ be the discrete input
  - $u(x,y) : \Omega \to \mathbb{R}$ be the interpolation solution
- $u(x,y)$ must satisfy: $(S * u)(ih_x, jh_y) = z[i,j]\,, \text{ for all pixels } (i,j)$

  where :
  - h_x, h_y are the *grid steps*, for which we consider $h_x = h_y = 1$
  - $S(x,y)$ is a *smoothing kernel* (performing lowpass filtering)



Reversibility Condition Approach (2)

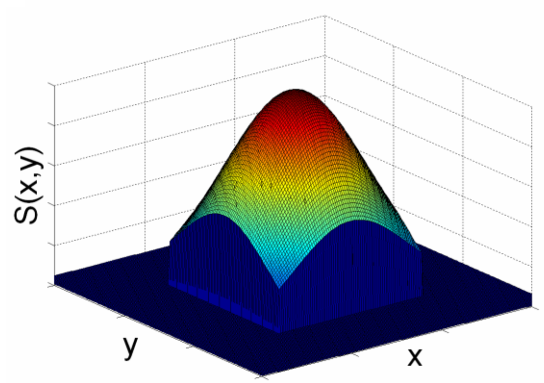
- We generalize this approach to vector-valued images:
- Let:
  - $\boldsymbol{z}[i,j] : \{1,..,N_x\}\times\{1,..,N_y\} \rightarrow \mathbb{R}^M$ be the discrete input
  - $\boldsymbol{u}(x,y) : \Omega \rightarrow \mathbb{R}^M$ be the interpolation solution

    $\boldsymbol{u} = [u_1,..,u_M]^{\mathrm{T}}$
- We restrict $\boldsymbol{u}(x,y)$ to satisfy:

$$\langle S_{ij}, u_m \rangle_{L^2(\Omega)} = z_m[i,j]\,, \quad \text{for all pixels } (i,j) \text{ and m } \in \{1,..,M\} \qquad (1)$$

- We have chosen:

$$S(x,y) = \mathbb{1}_{[-\frac{1}{2},\frac{1}{2}]^2}(x,y) \cdot \frac{G_{\hat{\sigma}}(x,y)}{\iint_{[-\frac{1}{2},\frac{1}{2}]^2} G_{\hat{\sigma}}(x',y')\mathrm{d}x'\mathrm{d}y'} \qquad (2)$$



Description of Our Method (1)

- We design a nonlinear diffusion flow, which:
  - lies on the subspace $\mathcal{U}_{z,S}$ of functions that satisfy the *Reversibility Condition*
  - performs *adaptive smoothing*, moving towards elements of $\mathcal{U}_{z,S}$ with better visual quality
- The above are accomplished by:
  - using an appropriate *projection operator*
  - modifying the PDE scheme of:

    [Tschumperle,Deriche, *IEEE-PAMI '05*]

Description of Our Method (2)

- Interpolated image $\longleftarrow$ equilibrium solution of:

$$\frac{\partial u_m(x,y,t)}{\partial t} = P_{\mathcal{U}_{0,S}}\left\{\mathrm{trace}\left(T\left(J_\rho(\nabla \boldsymbol{u}_\sigma)\right)\cdot D^2 u_m\right)\right\}\ ,\ m=1,..,M$$

Description of Our Method (2)

- Interpolated image ⟵ equilibrium solution of:

$$\frac{\partial u_m(x,y,t)}{\partial t} = P_{\mathcal{U}_{0,s}} \left\{ \text{trace}\left(T(J_\rho(\nabla \boldsymbol{u}_\sigma)) \cdot D^2 u_m \right) \right\} \ , m=1,..,M$$

RHS of PDE scheme in [Tschump.,Deriche, *IEEE-PAMI '05*]

artificial time

projection operator

2 x 2 diffusion tensor

spatial Hessian matrix of $u_m(x,y,t)$

Description of Our Method (3)

- Interpolated image $\longleftarrow$ equilibrium solution of:

$$\frac{\partial u_m(x,y,t)}{\partial t} = P_{\mathcal{U}_{0,s}} \left\{ \mathrm{trace}\Big(T\big(J_\rho(\nabla \boldsymbol{u}_\sigma)\big) \cdot D^2 u_m \Big) \right\} \quad , m=1,..,M$$

(initial conditions) : every $u_m(x,y,0)$ is derived from the *frequency zero-padding interpolation* of the *symmetrically extended* $z_m[i,j]$

the image that:

- satisfies Reversibility Condition
- has a Fourier Transform with all 0's outside baseband freqs.

Description of Our Method (4)

- Interpolated image $\longleftarrow$ equilibrium solution of:

$$\frac{\partial u_m(x,y,t)}{\partial t} = P_{\mathcal{U}_{0,S}}\left\{\mathrm{trace}\Big(T\big(J_\rho(\nabla \boldsymbol{u}_\sigma)\big)\cdot D^2 u_m\Big)\right\} \quad , m=1,..,M$$

where :

- $P_{\mathcal{U}_{0,S}}\{\cdot\}$ is the orthogonal projection on the space $\mathcal{U}_{0,S}$ of functions v that satisfy:

$$\langle S_{ij}, v\rangle_{L^2(\Omega)} = 0\,,\ \text{for all pixels } (i,j)$$

$$S_{ij} = S(i-x, j-y)$$

- Since $S(x,y)=0\,,\ \forall (x,y)\notin [-1/2,1/2]^2$, we have:

$$P_{\mathcal{U}_{0,S}}\{v\} = v - \|S\|^{-2}_{L^2(\mathbb{R}^2)} \sum_{i=1}^{N_x}\sum_{j=1}^{N_y} \langle S_{ij}, v\rangle_{L^2(\Omega)} \cdot S_{ij}$$

Description of Our Method (5)

- Interpolated image $\longleftarrow$ equilibrium solution of:

$$\frac{\partial u_m(x,y,t)}{\partial t} = P_{\mathcal{U}_{0,s}} \left\{ \text{trace}\left(T(J_\rho(\nabla \boldsymbol{u}_\sigma)) \cdot D^2 u_m \right) \right\} \quad , \; m=1,..,M$$

where :

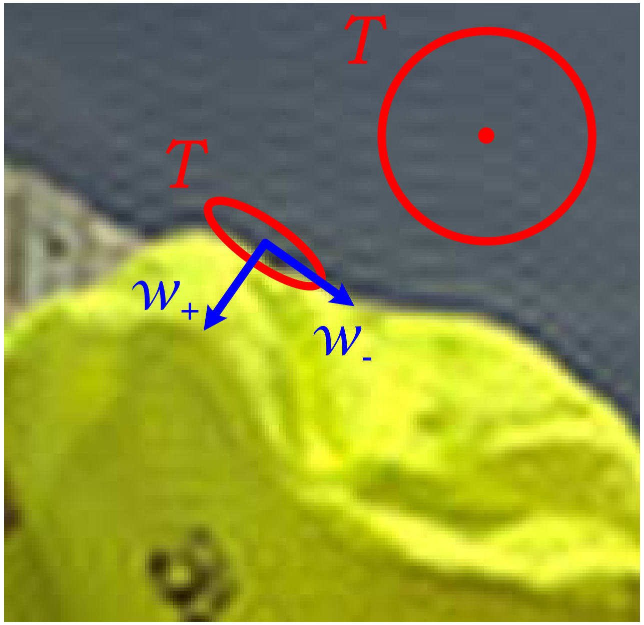
- $J_\rho(\nabla \boldsymbol{u}_\sigma)$ is the 2 x 2 *structure tensor* of image $\boldsymbol{u}(x,y,t)$:

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

Let $\left\{ \begin{matrix} \lambda_- \leq \lambda_+ \\ \boldsymbol{w}_-, \boldsymbol{w}_+ \end{matrix} \right\}$: $\left\{ \begin{matrix} \text{eigenvalues} \\ \text{eigenvectors} \end{matrix} \right\}$ of $J_\rho(\nabla \boldsymbol{u}_\sigma)$

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

$\mathcal{N} = \sqrt{\lambda_+ + \lambda_-}$ (edge strength predictor)



Numerical Implementation

- The discrete image $\boldsymbol{u}[i', j']$ approximates the continuous interpolation result $\boldsymbol{u}(x, y)$
  - The grid of $\boldsymbol{u}[i', j']$ is finer than and includes the grid of input $\boldsymbol{z}[i, j]$
  - $\boldsymbol{u}[i', j']$ is a $d \times d$ magnification of input (d integer)
- Discretization of the proposed PDE:

  explicit scheme with finite differences

Related Methods (1)

- TV-based Interpolation [Malgouyres,Guichard, *SIAM J. Num. Anal. '01*]

  minimize the TV: $E[u] = \iint_{\Omega} \|\nabla u\| \, \mathrm{d}x\mathrm{d}y$
  under the constraint that $u(x,y)$ satisfies the *Reversibility Condition*

  $$\Rightarrow \partial u(x,y,t)/\partial t = P_{\mathcal{U}_{0,s}} \{ \mathrm{div} \left(\nabla u / \|\nabla u\| \right) \}$$

  - $u(x,y,0)$ = zero-padding interpolation of $z[i,j]$

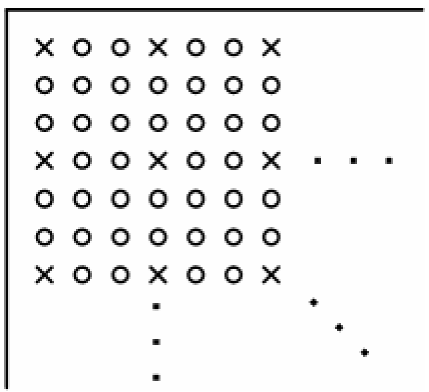
- Method of [Belahmidi,Guichard, *ICIP* 04] (BG)

  $$\frac{\partial u(x,y,t)}{\partial t} = \frac{\partial^2 u}{\partial \boldsymbol{\xi}^2} + g(\|\nabla u\|) \frac{\partial^2 u}{\partial \boldsymbol{n}^2} - \sum_{i=1}^{N_x} \sum_{j=1}^{N_y} \left\{ \iint_{\Omega_{i,j}} u(x,y,t) \mathrm{d}x\mathrm{d}y - z[i,j] \right\} \mathbb{1}_{\Omega_{i,j}}(x,y)$$

  - $u(x,y,0)$ = simple Zero Order Hold of $z[i,j]$

Related Methods (2)

- Interpolation method of [Tschumperle,Deriche, *IEEE-PAMI '05*] (TD)
  - The *exact interpolation condition* is posed
  - The problem is faced as a special case of *image inpainting*



×: pixels with known values
o: pixels forming the inpainting domain

  - In the inpainting domain only:

$$\frac{\partial u_m(x,y,t)}{\partial t} = \mathrm{trace}\left(\left[1 + (\mathcal{N}/K)^2\right]^{-\frac{1}{2}} \boldsymbol{w}_{-} \boldsymbol{w}_{-}^{\mathrm{T}} \cdot D^2 u_m \right), \quad m = 1,..,M$$

$u_m(x,y,0)$ = bilinear interpolation of $z_m[i,j]$

Framework for the Experiments

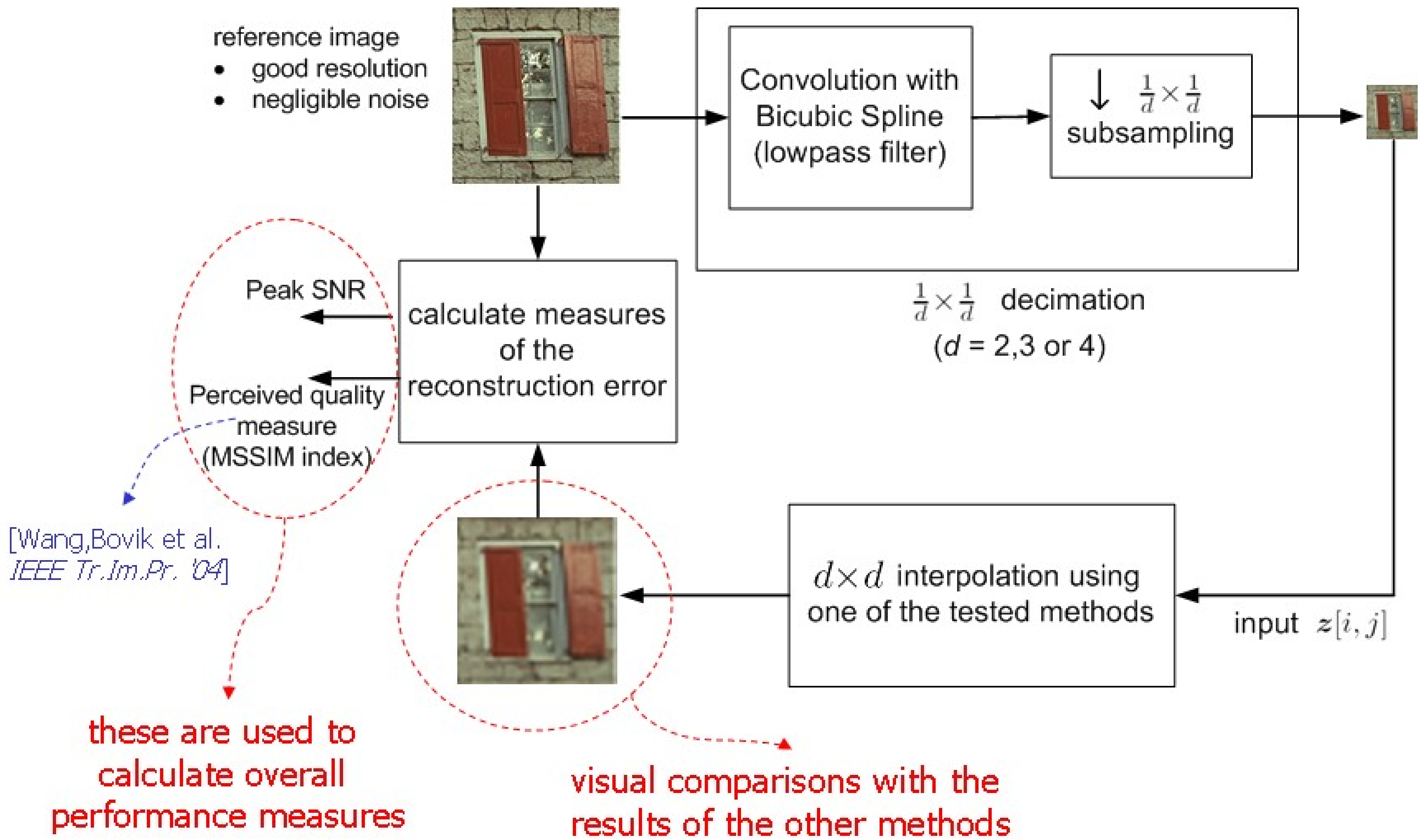
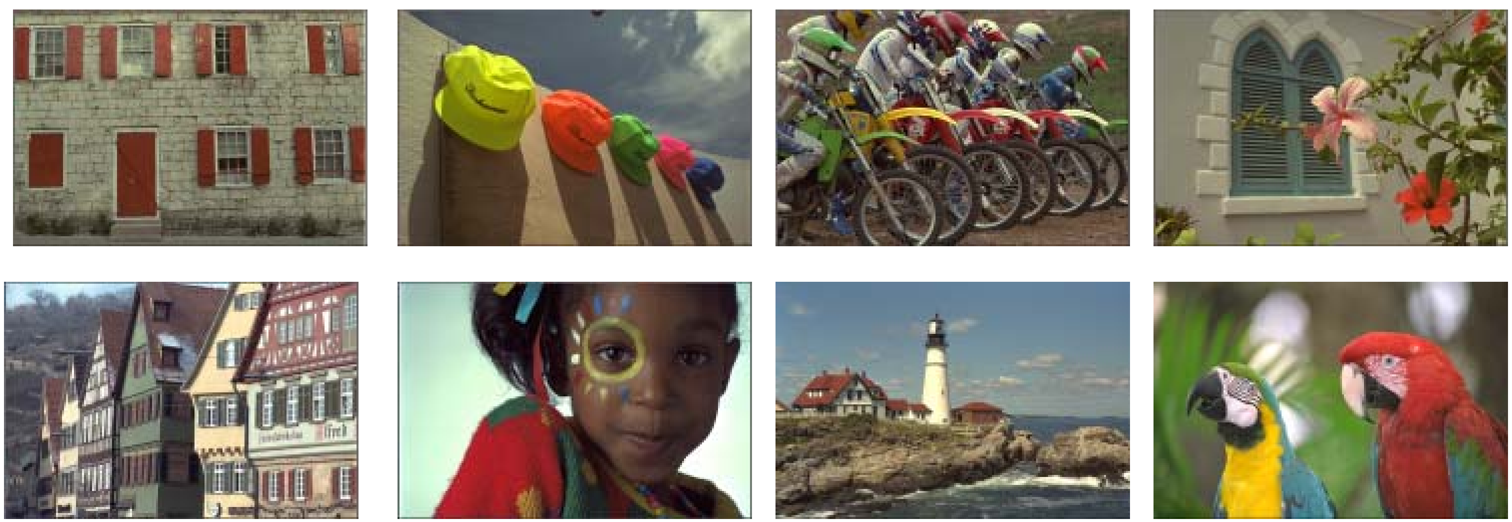


Image Interpolation Experiments

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

  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

Examples from the Color Results (1)

(a) Reference Image (detail)

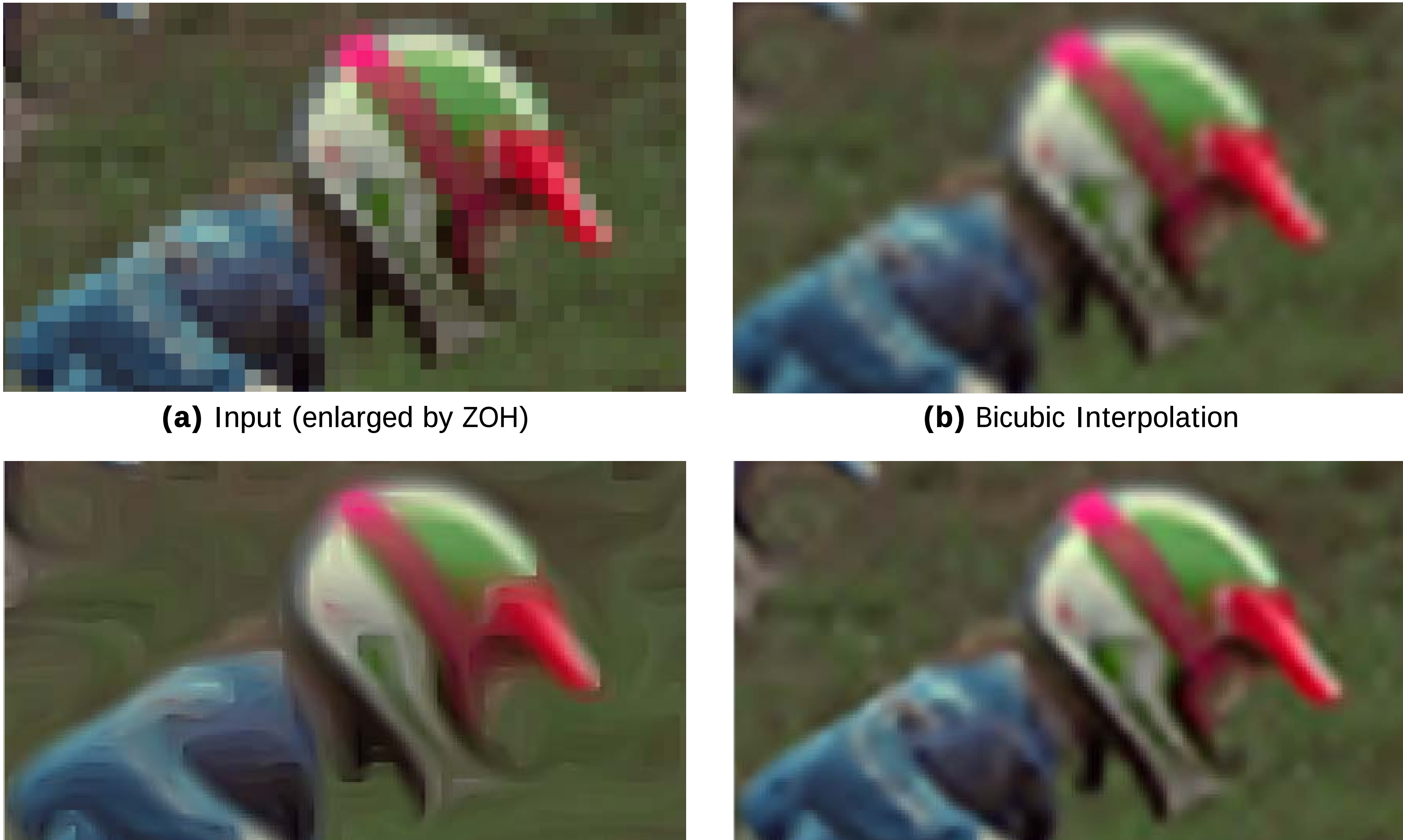
$\frac{1}{4} \times \frac{1}{4}$
decimation

(b) Input (detail)

(c) Input, enlarged by simple ZOH (detail)

Derivation of the input for 4 x 4 interpolation using the 5th reference image

Examples from the Color Results (2)



(a) Input (enlarged by ZOH)

(b) Bicubic Interpolation

(c) TD interpolation

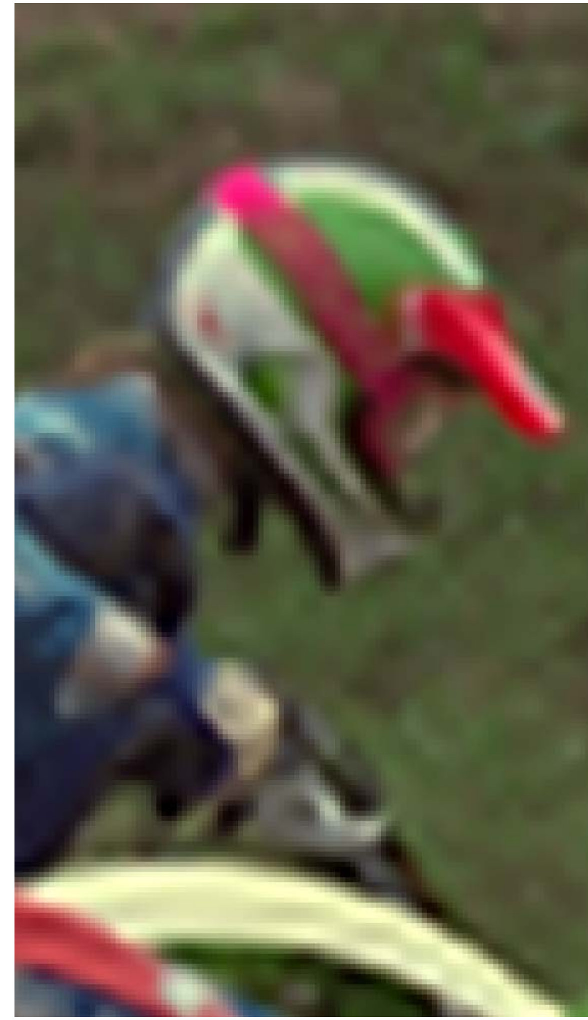
(d) Our method

Details of 4 x 4 color interpolation using the 5th reference image

Examples from the Color Results (3)

(a) Input (enlarged by ZOH)

(b) Initialization of our method



(c) Final result of our method

4 x 4 interpolation by the proposed method

Examples from the Color Results (4)

(a) Reference Image (detail)

$\frac{1}{4} \times \frac{1}{4}$ decimation

(b) Input (detail)

(c) Input, enlarged by simple ZOH (detail)

Derivation of the input for 4 x 4 interpolation using the 17th reference image

Examples from the Color Results (5)

(a) Input (enlarged by ZOH)

(b) Bicubic Interpolation

(c) TD interpolation

(d) Our method

Details of 4 x 4 color interpolation using the 17th reference image

Examples from the Color Results (6)

(a) Input (enlarged by ZOH)

(b) Initialization of our method

(c) Final result of our method

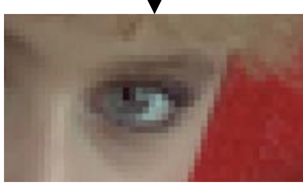
4 x 4 interpolation by the proposed method

Examples from the Color Results (7)

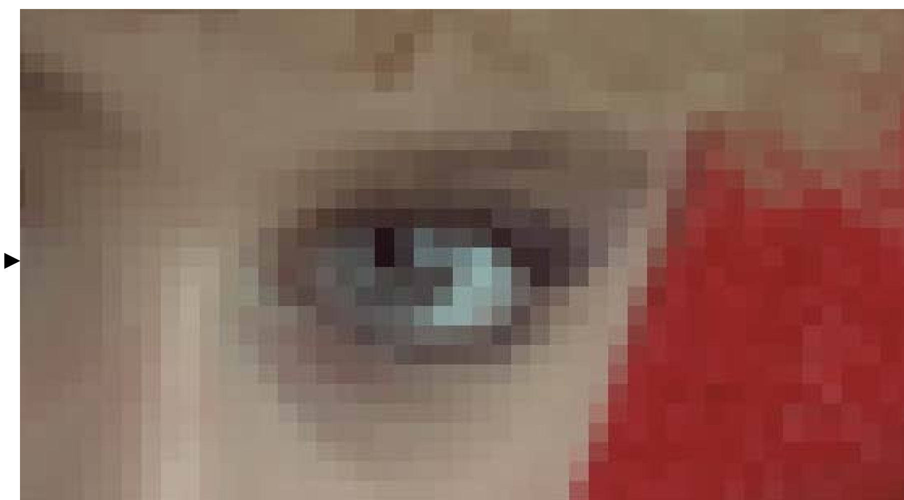
(a) Reference Image (detail)

$\frac{1}{4} \times \frac{1}{4}$

decimation



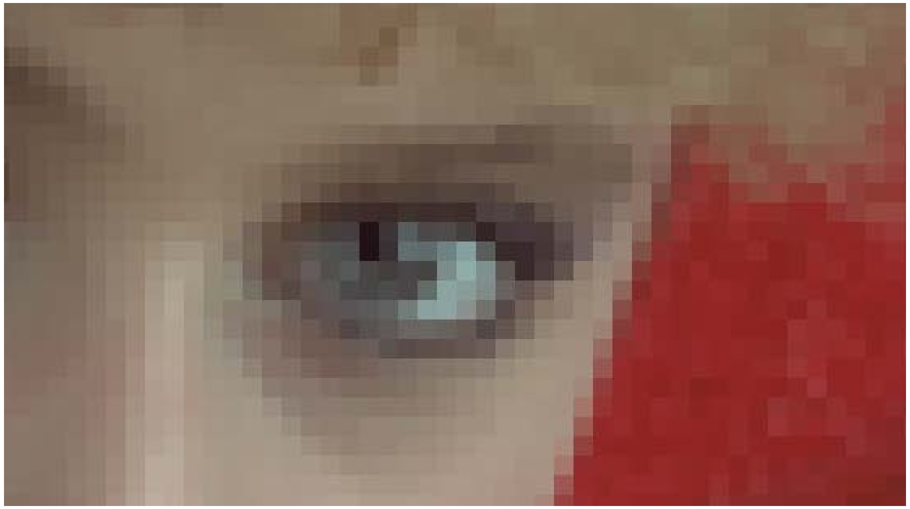
(b) Input (detail)



(c) Input, enlarged by simple ZOH (detail)

Derivation of the input for 4 x 4 interpolation using the 4th reference image

Examples from the Color Results (8)



(a) Input (enlarged by ZOH)

(b) Bicubic Interpolation

(c) TD interpolation

(d) Our method

Details of 4 x 4 color interpolation using the 4th reference image

Examples from the Graylevel Results (1)

(a) Reference Image (detail)

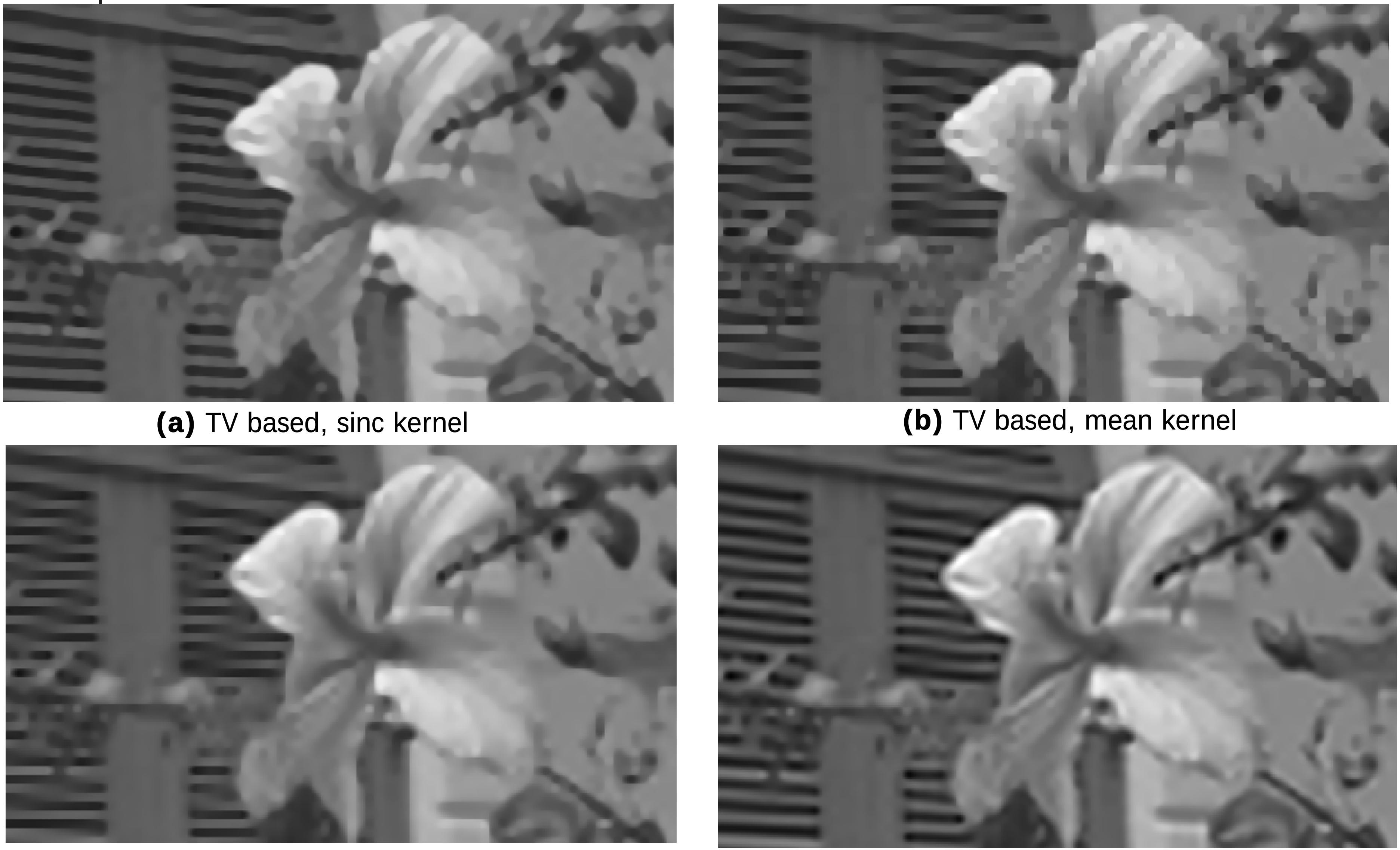
(b) Input (enlarged by ZOH)

(c) Initialization of our method

(d) Our method (Final result)

Results of our method in 4 x 4 interpolation using the 7th reference image

Examples from the Graylevel Results (2)



(a) TV based, sinc kernel

(b) TV based, mean kernel

(c) BG interpolation

(d) Our method

Details of 4 x 4 interpolation using the 4th reference image

Examples from the Graylevel Results (3)

(a) Reference Image (detail)

$\frac{1}{3} \times \frac{1}{3}$ decimation

(b) Input (detail)

(c) Input, enlarged by simple ZOH (detail)

Derivation of the input for 3 x 3 interpolation using the 14th reference image

Examples from the Graylevel Results (4)

(a) TV based, sinc kernel

(b) TV based, mean kernel

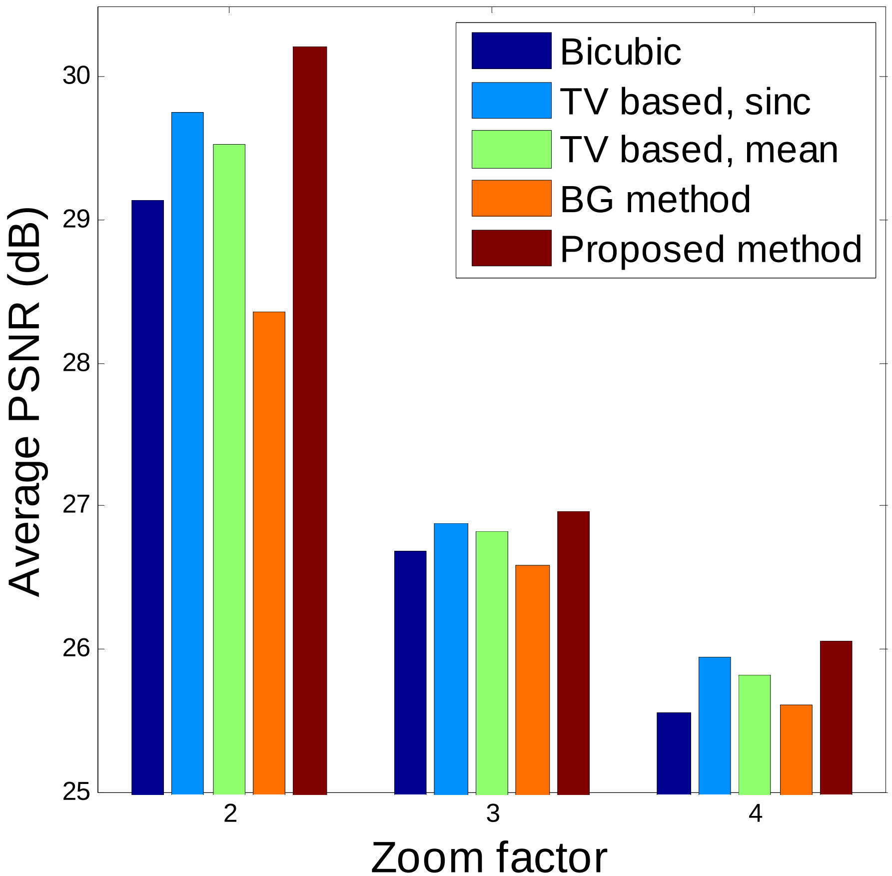
(c) BG interpolation

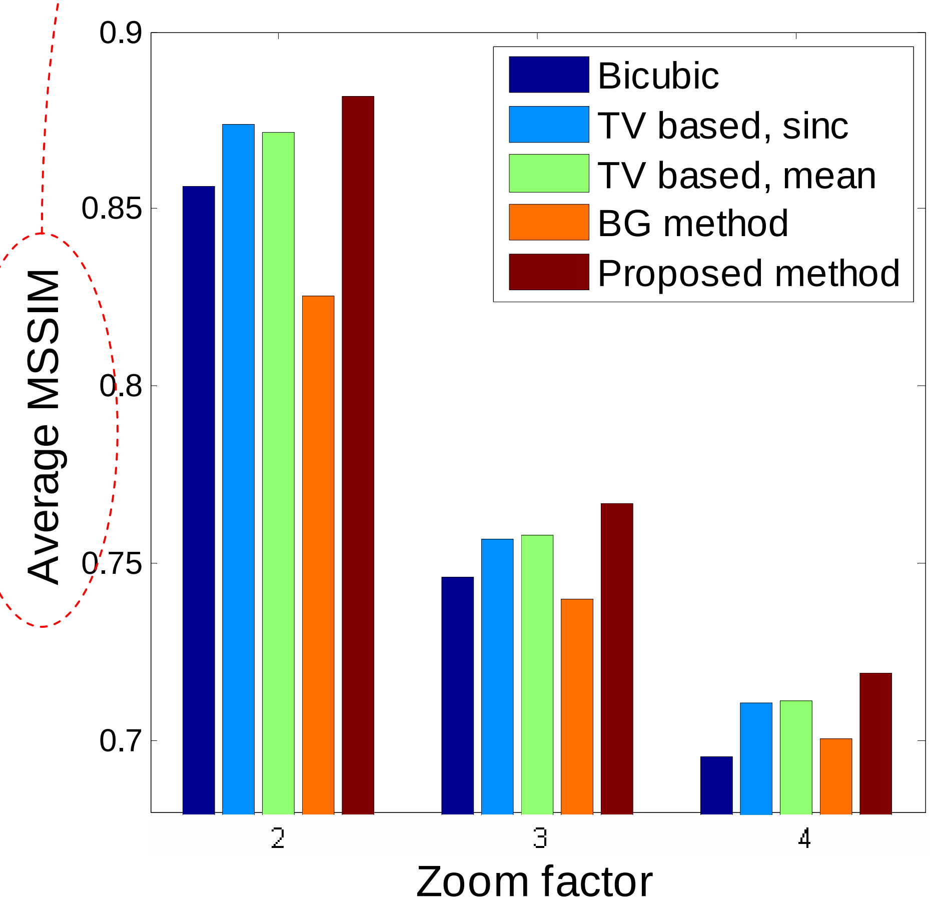
(d) Our method

Details of 3 x 3 interpolation using the 14^{th} reference image

Overall Performance Measures (1)

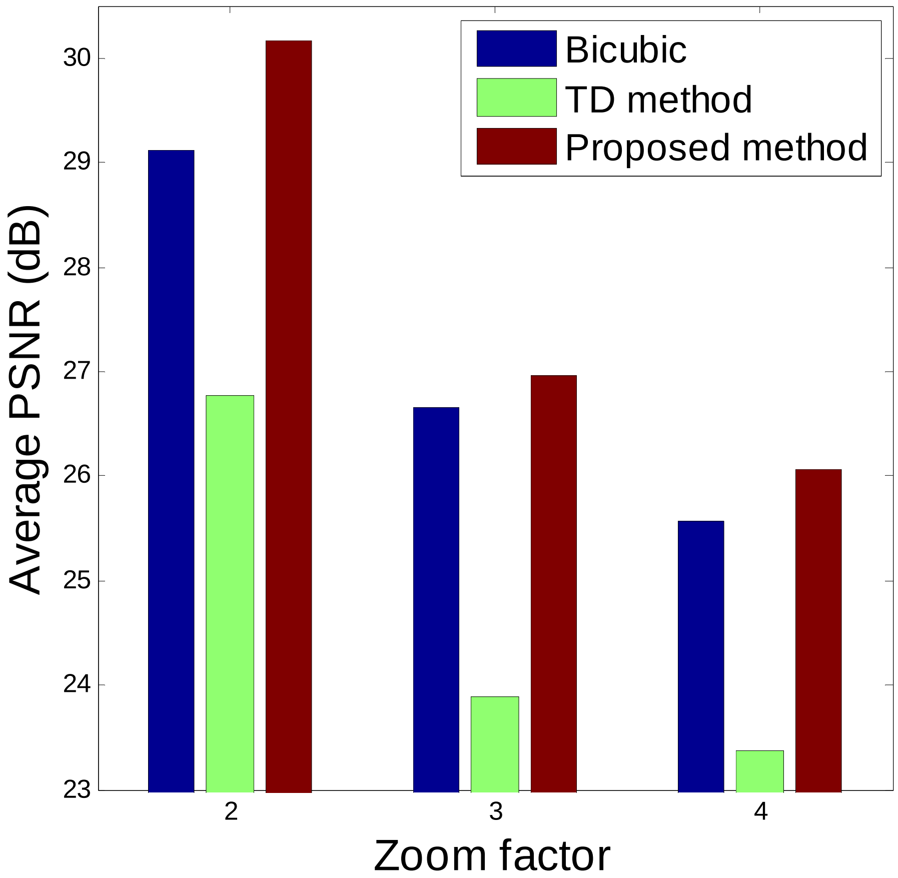
Mean Structural Similarity index
[Wang,Bovik et al. *IEEE Tr.Im.Pr. '04*]

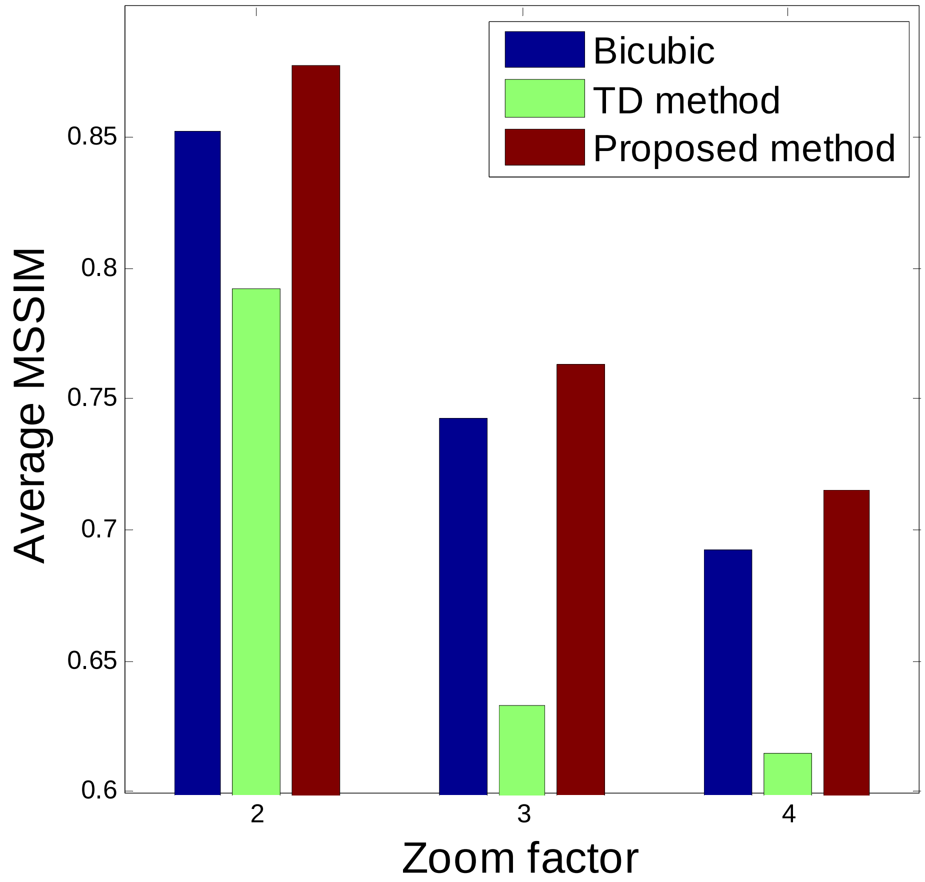




Graylevel Experiments in all 23 reference images

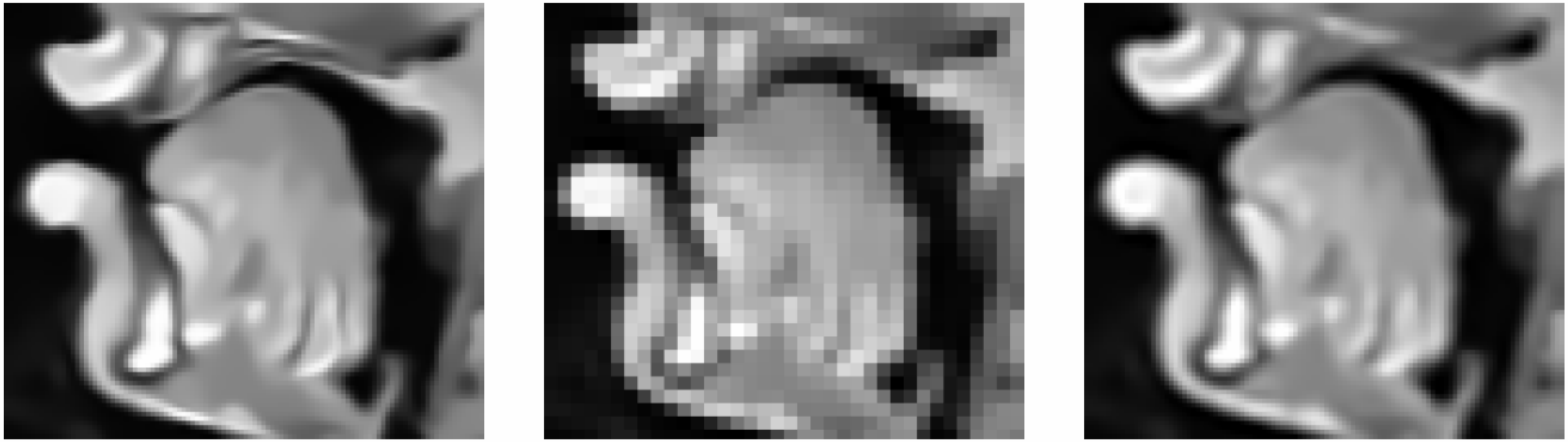
Overall Performance Measures (2)





Color Experiments in all 23 reference images

Example with Biomedical Image



(a) Reference image (MRI midsagittal)

(b) Reference after decimation (enlarged by ZOH)

(c) Interpolation of (b) using our method

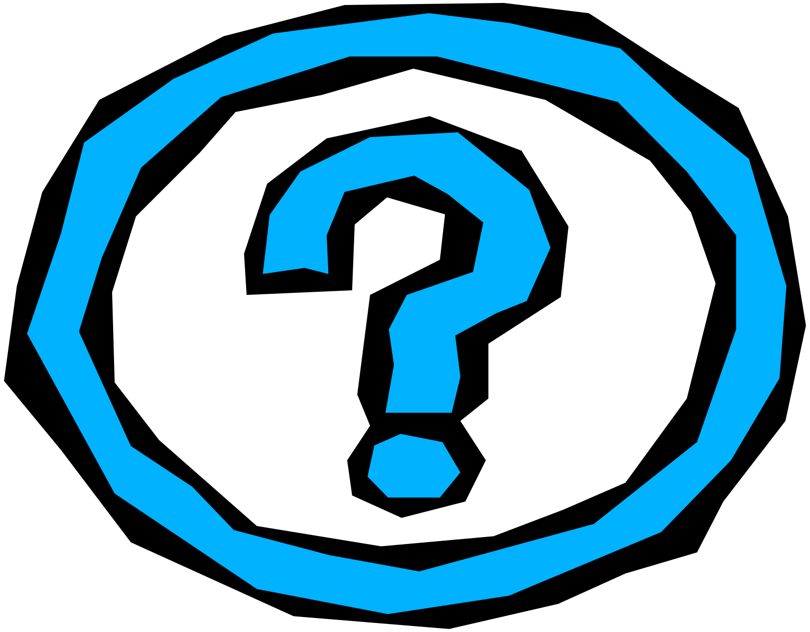
3x3 interpolation of a vocal tract image using the proposed method

Summary & Conclusions

- We have proposed a *nonlinear* method which:
  - is designed for *general vector-valued* images
  - is based on an *anisotropic diffusion PDE* with a *projection operator*
  - efficiently combines:
    - Reversibility Condition approach with
    - PDE model of [Tschumperle,Deriche, *IEEE-PAMI '05*]
- The experiments showed that this method:
  - *avoids most artifacts* of classic and other PDE-based methods
  - yields *improved error measures*

Thank You!!

Questions??



CVSP Group Web Site:

cvsp.cs.ntua.gr

Demonstration of Experimental Results:

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

Appendix

Appendix: Description of Our Method (4b)

- Interpolated image $\longleftarrow$ equilibrium solution of:

$$\frac{\partial u_m(x,y,t)}{\partial t} = P_{\mathcal{U}_{0,S}}\left\{\operatorname{trace}\Big(T\big(J_\rho(\nabla \boldsymbol{u}_\sigma)\big)\cdot D^2 u_m\Big)\right\} \quad , m=1,..,M$$

where :

- $P_{\mathcal{U}_{0,S}}\{\cdot\}$ is the orthogonal projection on the space $\mathcal{U}_{0,S}$ of functions v that satisfy:

$$\langle S_{ij}, v\rangle_{L^2(\Omega)} = 0\,, \text{ for all pixels } (i,j)$$

$$S_{ij} = S(i-x, j-y)$$

- Since $S(x,y)=0\,,\ \forall (x,y)\notin[-1/2,1/2]^2$, we have:

$$\Rightarrow \langle S_{ij}, S_{i'j'}\rangle_{L^2(\Omega)} = \|S\|^2_{L^2(\mathbb{R}^2)}\,\delta_{i-i',j-j'}$$

$$\Rightarrow P_{\mathcal{U}_{0,S}}\{v\} = v - \|S\|^{-2}_{L^2(\mathbb{R}^2)}\sum_{i=1}^{N_x}\sum_{j=1}^{N_y}\langle S_{ij}, v\rangle_{L^2(\Omega)}\cdot S_{ij}$$

Appendix: Examples from Color Results (A1)

(a) Reference Image

(b) Input (enlarged by ZOH)

(c) Bicubic Interpolation

(d) TD interpolation

(e) Initialization of our method

(f) Our method

Details of 4 x 4 color interpolation using the 5th reference image

Appendix: Examples from Color Results (A2)

(a) Reference Image

(b) Input (enlarged by ZOH)

(c) Bicubic Interpolation

(d) TD interpolation

(e) Initialization of our method

(f) Our method

Details of 3x3 color interpolation using the 23rd reference image

Appendix: Overall Performance Measures

Average error measures in all results using the 23 images:

Mean Structural Similarity index [Wang,Bovik et al. *IEEE Tr.Im.Pr. '04*]

zoom factor

Experiments with graylevel images

Interpolation Method	Average PSNR (dB)			Average MSSIM		
	d=2	d=3	d=4	d=2	d=3	d=4
Bicubic interpolation	29.14	26.68	25.55	0.8561	0.7464	0.6953
TV based, sinc kernel	29.75	26.87	25.94	0.8739	0.7567	0.7105
TV based, mean kernel	29.53	26.83	25.82	0.8714	0.7578	0.7114
BG interpolation	28.36	26.58	25.60	0.8253	0.7402	0.7004
Proposed method	**30.22**	**26.96**	**26.05**	**0.8816**	**0.7671**	**0.7194**

Experiments with color images

Interpolation Method	Average PSNR (dB)			Average MSSIM		
	d=2	d=3	d=4	d=2	d=3	d=4
Bicubic interpolation	29.11	26.66	25.56	0.8524	0.7425	0.6921
TD interpolation	26.77	23.89	23.37	0.7925	0.6330	0.6147
Proposed method	**30.16**	**26.96**	**26.06**	**0.8779**	**0.7631**	**0.7157**