Geometric Total Variation for Image Vectorization, Zooming and ~~Pixel Art~~ Pixel Art Depixelizing

Bertrand Kerautret[1] Jacques-Olivier Lachaud[2]

[1]LIRIS, University of Lyon, France

[2]LAMA, University Savoie Mont Blanc, France

30 March 2021 - CIRM - Marseille (virtually)

DGDVC

Work initially presented during the
Asian Conference on Pattern Recognition 2019

LIRIS

Laboratoire LAMA de Mathématiques

Geometric Total Variation for Image Vectorization, Zooming and Pixel Art Depixelizing

Context and Motivation

Variational formulation of Image Structuration

Vectorized contour regularization

Zoomed image with smooth spline contours

Experiments

Geometric Total Variation for Image Vectorization, Zooming and Pixel Art Depixelizing

Context and Motivation

Variational formulation of Image Structuration

Vectorized contour regularization

Zoomed image with smooth spline contours

Experiments

Image Vectorization, Zooming and Pixel Art Depixelizing

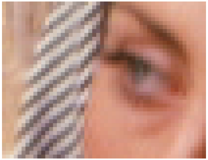
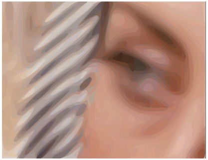
	Image zooming x16	Image Vectorization	Pixel Art Depixelizing
input			
	raster x16	vector image	raster x16
output			
	[Roussos-Maragios]	[Potrace]	[Kopf, Lischinsky]

Image Vectorization, Zooming and Pixel Art Depixelizing

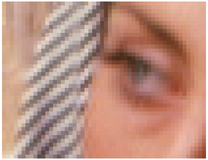
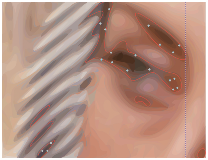
	Image zooming x16	Image Vectorization	Pixel Art Depixelizing
input			
	raster x16	vector image	raster x16
output			
	[Roussos-Maragios]	[Potrace]	[Kopf, Lischinsky]

Image Vectorization, Zooming and Pixel Art Depixelizing

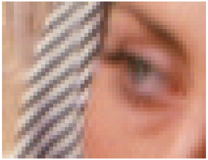
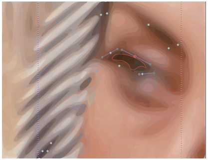
	Image zooming x16	Image Vectorization	Pixel Art Depixelizing
input			
	raster x16	vector image	raster x16
output			
	[Roussos-Maragios]	[Potrace]	[Kopf, Lischinsky]

Image Vectorization, Zooming and Pixel Art Depixelizing

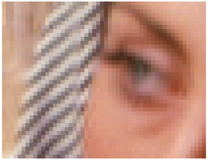
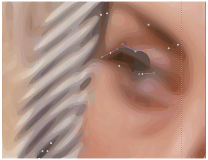
	Image zooming x16	Image Vectorization	Pixel Art Depixelizing
input			
	raster x16	vector image	raster x16
output			
	[Roussos-Maragios]	[Potrace]	[Kopf, Lischinsky]

Image Vectorization, Zooming and Pixel Art Depixelizing

Method	Photo	Quant. image	Math. model	Sample	Min-max

Image Vectorization, Zooming and Pixel Art Depixelizing

Method	Photo	Quant. image	Math. model	Sample	Min-max

Image zooming ⇒ non editable raster image					
Interpolation	fair	smooth	C^k fcts	yes	yes/no
Reprojection	good	bad	variational	no	no
CNN	good	bad	learning	no	no

Image Vectorization, Zooming and Pixel Art Depixelizing

Method	Photo	Quant. image	Math. model	Sample	Min-max

Image zooming ⇒ non editable raster image					
Interpolation	fair	smooth	C^k fcts	yes	yes/no
Reprojection	good	bad	variational	no	no
CNN	good	bad	learning	no	no

Image vectorization ⇒ editable vector image					
levelsets	fair/bad	good	contour	no	yes
Delaunay	fair/bad	good	geometry	no	yes

Image Vectorization, Zooming and Pixel Art Depixelizing

Method	Photo	Quant. image	Math. model	Sample	Min-max
Image zooming ⇒ non editable raster image					
Interpolation	fair	smooth	C^k fcts	yes	yes/no
Reprojection	good	bad	variational	no	no
CNN	good	bad	learning	no	no
Image vectorization ⇒ editable vector image					
levelsets	fair/bad	good	contour	no	yes
Delaunay	fair/bad	good	geometry	no	yes
Pixel art depixelizing ⇒ non editable raster image					
Linear HQx	bad	fair	local	yes	yes
Agregation	bad	good	ad hoc	yes	yes

Image Vectorization, Zooming and Pixel Art Depixelizing

Method	Photo	Quant. image	Math. model	Sample	Min-max
Image zooming ⇒ non editable raster image					
Interpolation	fair	smooth	C^k fcts	yes	yes/no
Reprojection	good	bad	variational	no	no
CNN	good	bad	learning	no	no
Image vectorization ⇒ editable vector image					
levelsets	fair/bad	good	contour	no	yes
Delaunay	fair/bad	good	geometry	no	yes
Pixel art depixelizing ⇒ non editable raster image					
Linear HQx	bad	fair	local	yes	yes
Agregation	bad	good	ad hoc	yes	yes
Our approach: raster + vector image					
Geom. TV	good	good	variational	yes	yes

Geometric Total Variation for Image Vectorization, Zooming and Pixel Art Depixelizing

Context and Motivation

Variational formulation of Image Structuration

Vectorized contour regularization

Zoomed image with smooth spline contours

Experiments

A Geometric Total Variation I

- Idea: find linear structures in image before zooming/vectorizing
- Total Variation (TV): classical energy for image restoration, inpainting
- Continuous TV: image/function $f : \Omega \to \mathbb{R}^3$, some norm $\|\cdot\|_K$

$$TV(f) := \int_\Omega \|\nabla f(x)\|_K dx, \quad \text{(or by duality)}$$

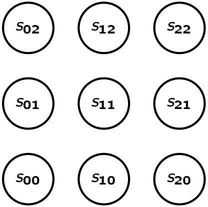
A Geometric Total Variation I

- Idea: find linear structures in image before zooming/vectorizing
- Total Variation (TV): classical energy for image restoration, inpainting
- Continuous TV: image/function $f : \Omega \to \mathbb{R}^3$, some norm $\|\cdot\|_K$

$$TV(f) := \int_\Omega \|\nabla f(x)\|_K dx, \quad \text{(or by duality)}$$

- Discretized TV: image $s : \Omega \cap \mathbb{Z}^2 \to \mathbb{R}^3$,

$$TV(s) := \sum_{\text{pixels } (i,j)} \|\widehat{\mathrm{grad}}\ s(i,j)\|_K dx,$$



A Geometric Total Variation I

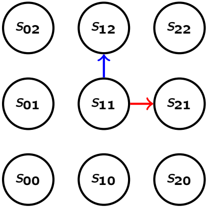
- Idea: find linear structures in image before zooming/vectorizing
- Total Variation (TV): classical energy for image restoration, inpainting
- Continuous TV: image/function $f : \Omega \to \mathbb{R}^3$, some norm $\|\cdot\|_K$

$$TV(f) := \int_\Omega \|\nabla f(x)\|_K dx, \quad \text{(or by duality)}$$

- Discretized TV: image $s : \Omega \cap \mathbb{Z}^2 \to \mathbb{R}^3$,

$$TV(s) := \sum_{\text{pixels } (i,j)} \|\widehat{\mathrm{grad}}\, s(i,j)\|_K dx,$$

- **Many** $\widehat{\mathrm{grad}}$: $\begin{bmatrix} s_{i+1,j} - s_{i,j} \\ s_{i,j+1} - s_{i,j} \end{bmatrix}$, etc.



A Geometric Total Variation I

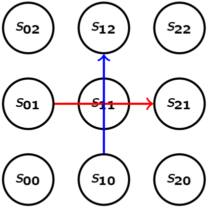
- Idea: find linear structures in image before zooming/vectorizing
- Total Variation (TV): classical energy for image restoration, inpainting
- Continuous TV: image/function $f : \Omega \to \mathbb{R}^3$, some norm $\|\cdot\|_K$

$$TV(f) := \int_\Omega \|\nabla f(x)\|_K dx, \quad \text{(or by duality)}$$

- Discretized TV: image $s : \Omega \cap \mathbb{Z}^2 \to \mathbb{R}^3$,

$$TV(s) := \sum_{\text{pixels } (i,j)} \|\widehat{\text{grad}}\, s(i,j)\|_K dx,$$

- **Many** $\widehat{\text{grad}}$: $\begin{bmatrix} s_{i+1,j} - s_{i,j} \\ s_{i,j+1} - s_{i,j} \end{bmatrix}$, etc.



A Geometric Total Variation I

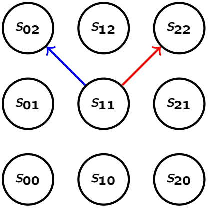
- Idea: find linear structures in image before zooming/vectorizing
- Total Variation (TV): classical energy for image restoration, inpainting
- Continuous TV: image/function $f : \Omega \to \mathbb{R}^3$, some norm $\|\cdot\|_K$

$$TV(f) := \int_\Omega \|\nabla f(x)\|_K dx, \quad \text{(or by duality)}$$

- Discretized TV: image $s : \Omega \cap \mathbb{Z}^2 \to \mathbb{R}^3$,

$$TV(s) := \sum_{\text{pixels } (i,j)} \|\widehat{\mathrm{grad}}\ s(i,j)\|_K dx,$$

- **Many** $\widehat{\mathrm{grad}}$: $\begin{bmatrix} s_{i+1,j} - s_{i,j} \\ s_{i,j+1} - s_{i,j} \end{bmatrix}$, etc.



A Geometric Total Variation I

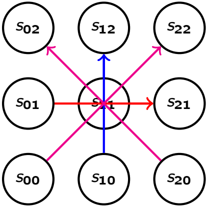
- Idea: find linear structures in image before zooming/vectorizing
- Total Variation (TV): classical energy for image restoration, inpainting
- Continuous TV: image/function $f : \Omega \to \mathbb{R}^3$, some norm $\|\cdot\|_K$

$$TV(f) := \int_\Omega \|\nabla f(x)\|_K dx, \quad \text{(or by duality)}$$

- Discretized TV: image $s : \Omega \cap \mathbb{Z}^2 \to \mathbb{R}^3$,

$$TV(s) := \sum_{\text{pixels } (i,j)} \|\widehat{\mathrm{grad}}\; s(i,j)\|_K dx,$$

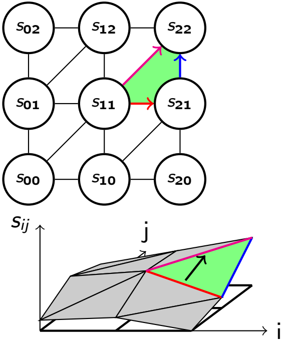
- **Many** $\widehat{\mathrm{grad}}$: $\begin{bmatrix} s_{i+1,j} - s_{i,j} \\ s_{i,j+1} - s_{i,j} \end{bmatrix}$, etc.



A Geometric Total Variation II

- $\mathcal{T}(\Omega)$: Set of triangulation of $\Omega \cap \mathbb{Z}^2$
- $\widehat{\text{grad}}$ per triangle:

$$\widehat{\text{grad}}\, s(\mathbf{pqr}) := s(\mathbf{p})(\mathbf{r}-\mathbf{q})^{\perp} + s(\mathbf{q})(\mathbf{p}-\mathbf{r})^{\perp} + s(\mathbf{r})(\mathbf{q}-\mathbf{p})^{\perp}.$$



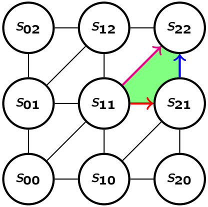
A Geometric Total Variation II

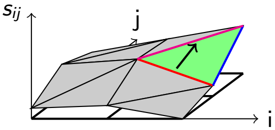
- $\mathcal{T}(\Omega)$: Set of triangulation of $\Omega \cap \mathbb{Z}^2$
- $\widehat{\text{grad}}$ per triangle:

$$\widehat{\text{grad}}\; s(\mathbf{pqr}) := s(\mathbf{p})(\mathbf{r}-\mathbf{q})^{\perp} + s(\mathbf{q})(\mathbf{p}-\mathbf{r})^{\perp} + s(\mathbf{r})(\mathbf{q}-\mathbf{p})^{\perp}.$$

- For any triangulation $T \in \mathcal{T}(\Omega)$, geometric TV of image s is

$$\mathrm{GTV}(T,s) = \frac{1}{2} \sum_{\mathbf{pqr} \in T} \left\| \widehat{\text{grad}}\; s(\mathbf{pqr}) \right\|_K$$

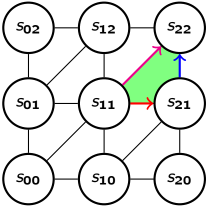




A Geometric Total Variation II

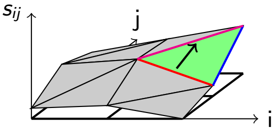
- $\mathcal{T}(\Omega)$: Set of triangulation of $\Omega \cap \mathbb{Z}^2$
- $\widehat{\text{grad}}$ per triangle:

$$\widehat{\text{grad}}\, s(\mathbf{pqr}) := s(\mathbf{p})(\mathbf{r}-\mathbf{q})^{\perp} + s(\mathbf{q})(\mathbf{p}-\mathbf{r})^{\perp} + s(\mathbf{r})(\mathbf{q}-\mathbf{p})^{\perp}.$$



- For any triangulation $T \in \mathcal{T}(\Omega)$, geometric TV of image s is

$$\text{GTV}(T,s) = \frac{1}{2} \sum_{\mathbf{pqr} \in T} \left\| \widehat{\text{grad}}\, s(\mathbf{pqr}) \right\|_K$$



- Minimizing $\text{GTV}(T,s)$ for $T \in \mathcal{T}(\Omega)$
  $\Leftrightarrow$ Minimizing $\text{TV}(f)$ among piecewise linear functions f with $f(x,y) = s(x,y)$ on lattice points (for all components R,G,B).

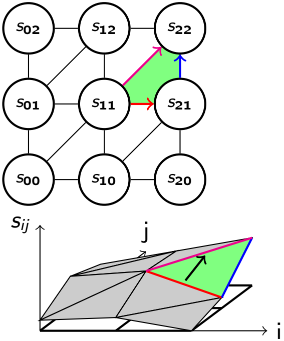
A Geometric Total Variation II

- $\mathcal{T}(\Omega)$: Set of triangulation of $\Omega \cap \mathbb{Z}^2$
- $\widehat{\mathrm{grad}}$ per triangle:

$$\widehat{\mathrm{grad}}\, s(\mathbf{pqr}) := s(\mathbf{p})(\mathbf{r}-\mathbf{q})^{\perp} + s(\mathbf{q})(\mathbf{p}-\mathbf{r})^{\perp} + s(\mathbf{r})(\mathbf{q}-\mathbf{p})^{\perp}.$$

- For any triangulation $T \in \mathcal{T}(\Omega)$, geometric TV of image s is

$$\mathrm{GTV}(T, s) = \frac{1}{2} \sum_{\mathbf{pqr} \in T} \left\| \widehat{\mathrm{grad}}\, s(\mathbf{pqr}) \right\|_K$$



- Minimizing $\mathrm{GTV}(T, s)$ for $T \in \mathcal{T}(\Omega)$
  $\Leftrightarrow$ Minimizing $\mathrm{TV}(f)$ among piecewise linear functions f with $f(x, y) = s(x, y)$ on lattice points (for all components R,G,B).
- Generally much lower than $\mathrm{TV}(s)$ since $\widehat{\mathrm{grad}}$ may not be local.

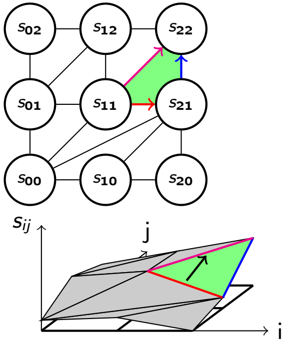
A Geometric Total Variation II

- $\mathcal{T}(\Omega)$: Set of triangulation of $\Omega \cap \mathbb{Z}^2$
- $\widehat{\mathrm{grad}}$ per triangle:

$$\widehat{\mathrm{grad}}\, s(\mathbf{pqr}) := s(\mathbf{p})(\mathbf{r}-\mathbf{q})^{\perp} + s(\mathbf{q})(\mathbf{p}-\mathbf{r})^{\perp} + s(\mathbf{r})(\mathbf{q}-\mathbf{p})^{\perp}.$$

- For any triangulation $T \in \mathcal{T}(\Omega)$, geometric TV of image s is

$$\mathrm{GTV}(T, s) = \frac{1}{2} \sum_{\mathbf{pqr} \in T} \left\| \widehat{\mathrm{grad}}\, s(\mathbf{pqr}) \right\|_K$$



- Minimizing $\mathrm{GTV}(T, s)$ for $T \in \mathcal{T}(\Omega)$
  $\Leftrightarrow$ Minimizing $\mathrm{TV}(f)$ among piecewise linear functions f with $f(x, y) = s(x, y)$ on lattice points (for all components R,G,B).
- Generally much lower than $\mathrm{TV}(s)$ since $\widehat{\mathrm{grad}}$ may not be local.

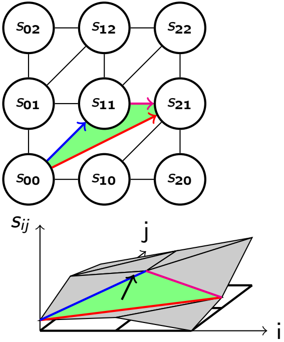
A Geometric Total Variation II

- $\mathcal{T}(\Omega)$: Set of triangulation of $\Omega \cap \mathbb{Z}^2$
- $\widehat{\mathrm{grad}}$ per triangle:

$$\widehat{\mathrm{grad}}\ s(\mathbf{pqr}) := s(\mathbf{p})(\mathbf{r}-\mathbf{q})^{\perp} + s(\mathbf{q})(\mathbf{p}-\mathbf{r})^{\perp} + s(\mathbf{r})(\mathbf{q}-\mathbf{p})^{\perp}.$$

- For any triangulation $T \in \mathcal{T}(\Omega)$, geometric TV of image s is

$$\mathrm{GTV}(T, s) = \frac{1}{2} \sum_{\mathbf{pqr} \in T} \left\| \widehat{\mathrm{grad}}\ s(\mathbf{pqr}) \right\|_K$$



  - Minimizing $\mathrm{GTV}(T, s)$ for $T \in \mathcal{T}(\Omega)$
    $\Leftrightarrow$ Minimizing $\mathrm{TV}(f)$ among piecewise linear functions f with $f(x, y) = s(x, y)$ on lattice points (for all components R,G,B).
  - Generally much lower than $\mathrm{TV}(s)$ since $\widehat{\mathrm{grad}}$ may not be local.

Example of GTV

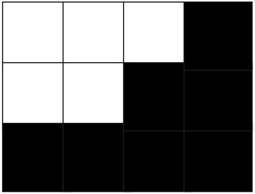
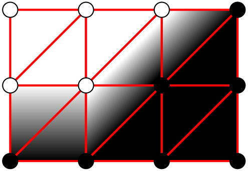
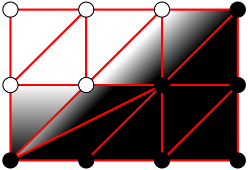


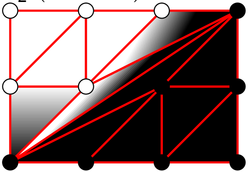
image s



$\frac{1}{2}\left(2+3\sqrt{2}\right) \approx 3.121$



$\frac{1}{2}\left(1+\sqrt{5}+2\sqrt{2}\right) \approx 3.032$



$\frac{1}{2}\left(1+\sqrt{13}+\sqrt{2}\right) \approx 3.010$

Example of GTV

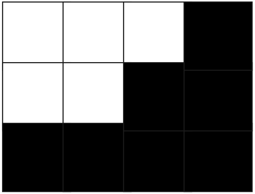
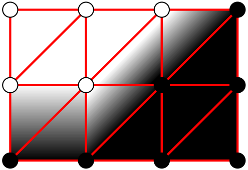
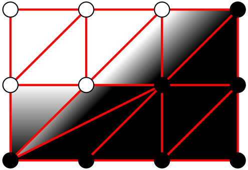


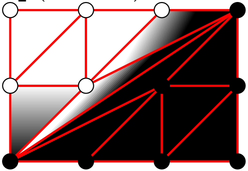
image s



$\frac{1}{2}\left(2+3\sqrt{2}\right) \approx 3.121$



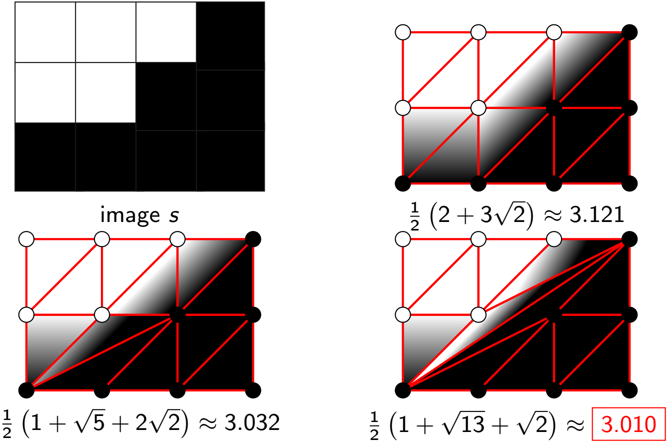
$\frac{1}{2}\left(1+\sqrt{5}+2\sqrt{2}\right) \approx 3.032$



$\frac{1}{2}\left(1+\sqrt{13}+\sqrt{2}\right) \approx \boxed{3.010}$

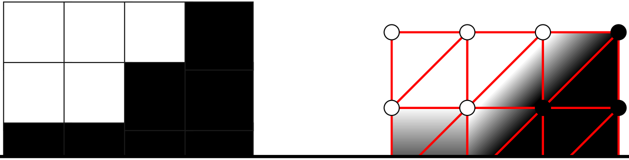
- Optimal triangulations for GTV align with digital straight lines

Example of GTV

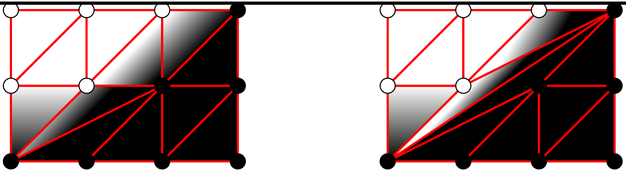


- Optimal triangulations for GTV align with digital straight lines
- GTV structures the image along strong gradients / linear features

Example of GTV



Optimal triangulation is a **vectorized** representation of image s.



$\frac{1}{2}\left(1+\sqrt{5}+2\sqrt{2}\right) \approx 3.032$ $\qquad$ $\frac{1}{2}\left(1+\sqrt{13}+\sqrt{2}\right) \approx \boxed{3.010}$

- Optimal triangulations for GTV align with digital straight lines
- GTV structures the image along strong gradients / linear features

Optimization algorithm for GTV

Greedy randomized optimization algorithm

Start Image s, trivial triangulation T of $\mathrm{Domain}(s)$

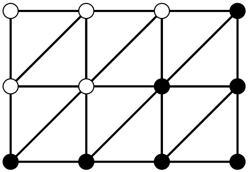
Process by rounds edges that may decrease GTV are queued

Greedy optimization flip an edge of T if it decreases energy $\mathrm{GTV}(T, s)$
or

Randomized if it does not change $\mathrm{GTV}(T, s)$ with a probability $\frac{1}{2}$

Update arcs surrounding a flip are queued for next round

Stop when no *decreasing* flip occurred in last round



Optimization algorithm for GTV

Greedy randomized optimization algorithm

Start Image s, trivial triangulation T of $\mathrm{Domain}(s)$

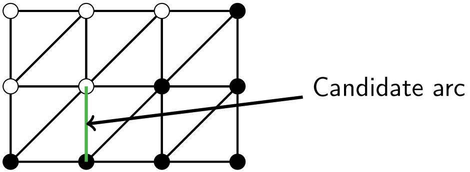
Process by rounds edges that may decrease GTV are queued

Greedy optimization flip an edge of T if it decreases energy $\mathrm{GTV}(T, s)$
or

Randomized if it does not change $\mathrm{GTV}(T, s)$ with a probability $\frac{1}{2}$

Update arcs surrounding a flip are queued for next round

Stop when no *decreasing* flip occurred in last round



Optimization algorithm for GTV

Greedy randomized optimization algorithm

Start Image s, trivial triangulation T of $\mathrm{Domain}(s)$

Process by rounds edges that may decrease GTV are queued

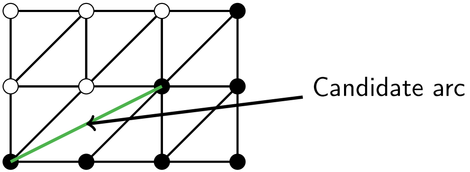
Greedy optimization flip an edge of T if it decreases energy $\mathrm{GTV}(T, s)$

or

Randomized if it does not change $\mathrm{GTV}(T, s)$ with a probability $\frac{1}{2}$

Update arcs surrounding a flip are queued for next round

Stop when no *decreasing* flip occurred in last round



Optimization algorithm for GTV

Greedy randomized optimization algorithm

Start Image s, trivial triangulation T of $\mathrm{Domain}(s)$

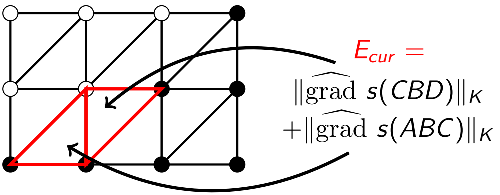
Process by rounds edges that may decrease GTV are queued

Greedy optimization flip an edge of T if it decreases energy $\mathrm{GTV}(T, s)$
or

Randomized if it does not change $\mathrm{GTV}(T, s)$ with a probability $\frac{1}{2}$

Update arcs surrounding a flip are queued for next round

Stop when no *decreasing* flip occurred in last round



Optimization algorithm for GTV

Greedy randomized optimization algorithm

Start Image s, trivial triangulation T of $\mathrm{Domain}(s)$

Process by rounds edges that may decrease GTV are queued

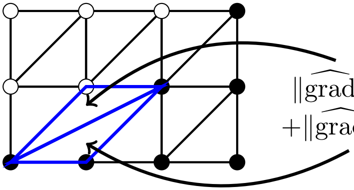
Greedy optimization flip an edge of T if it decreases energy $\mathrm{GTV}(T, s)$

or

Randomized if it does not change $\mathrm{GTV}(T, s)$ with a probability $\frac{1}{2}$

Update arcs surrounding a flip are queued for next round

Stop when no *decreasing* flip occurred in last round



$E_{flip} =$
$\|\widehat{\mathrm{grad}}\ s(ADC)\|_K$
$+\|\widehat{\mathrm{grad}}\ s(ABD)\|_K$

Optimization algorithm for GTV

Greedy randomized optimization algorithm

Start Image s, trivial triangulation T of $\mathrm{Domain}(s)$

Process by rounds edges that may decrease GTV are queued

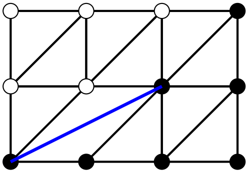
Greedy optimization flip an edge of T if it decreases energy $\mathrm{GTV}(T, s)$

or

Randomized if it does not change $\mathrm{GTV}(T, s)$ with a probability $\frac{1}{2}$

Update arcs surrounding a flip are queued for next round

Stop when no *decreasing* flip occurred in last round



Flip arc since

$E_{flip} < E_{cur}$

Optimization algorithm for GTV

Greedy randomized optimization algorithm

Start Image s, trivial triangulation T of $\mathrm{Domain}(s)$

Process by rounds edges that may decrease GTV are queued

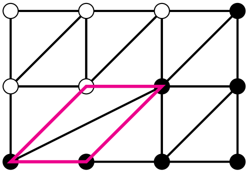
Greedy optimization flip an edge of T if it decreases energy $\mathrm{GTV}(T, s)$

or

Randomized if it does not change $\mathrm{GTV}(T, s)$ with a probability $\frac{1}{2}$

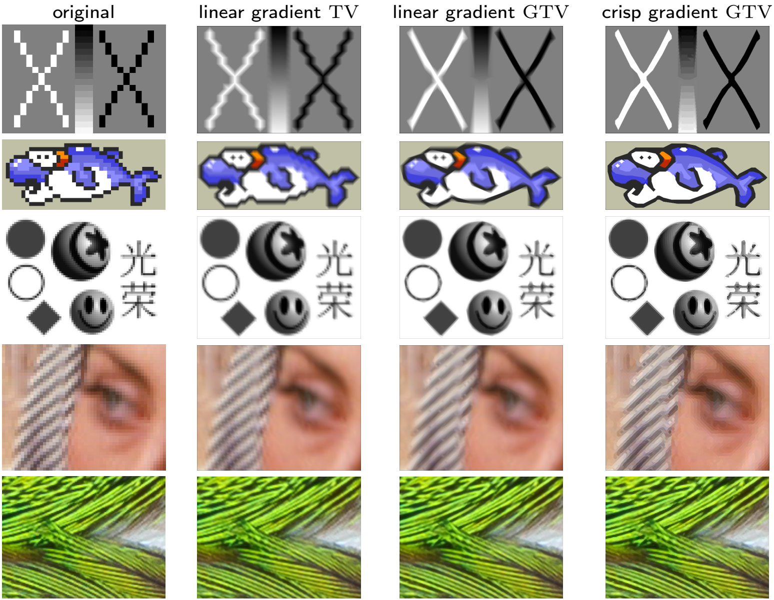
Update arcs surrounding a flip are queued for next round

Stop when no *decreasing* flip occurred in last round



Put nearby arcs in queue

Results of GTV: rasterized at ×16



Geometric Total Variation for Image Vectorization, Zooming and Pixel Art Depixelizing

Context and Motivation

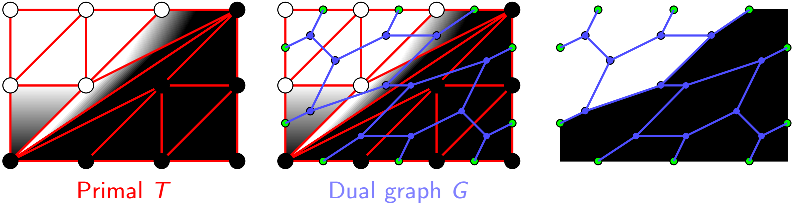
Variational formulation of Image Structuration

Vectorized contour regularization

Zoomed image with smooth spline contours

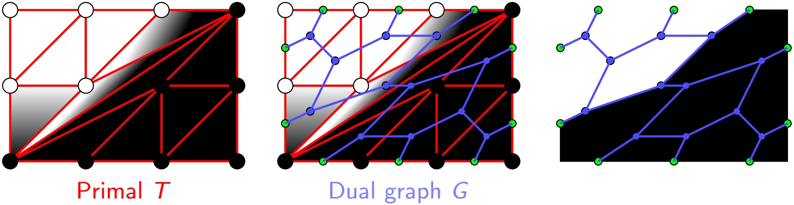
Experiments

Contour extraction and regularization



- Contours are in-between pixels ⇒ Dual graph G of T
- One pixel per dual face ⇒ One color per dual face

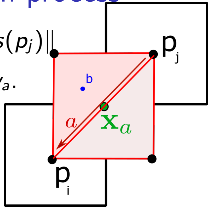
Contour extraction and regularization



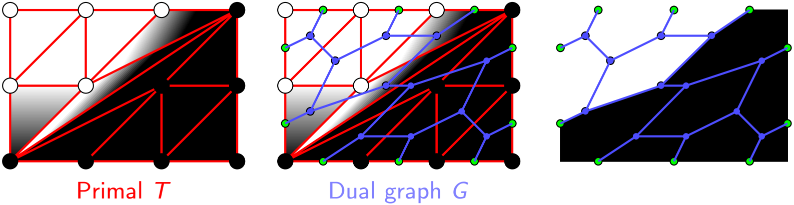
- Contours are in-between pixels $\Rightarrow$ Dual graph G of T
- One pixel per dual face $\Rightarrow$ One color per dual face

Dual graph improved by a variational regularization process

- Dissimilarity weights on arcs $a = (p_i, p_j)$: $w_a = \|s(p_i) - s(p_j)\|$
- Barycenters b_i are made closer to arcs of T with strong w_a.



Contour extraction and regularization



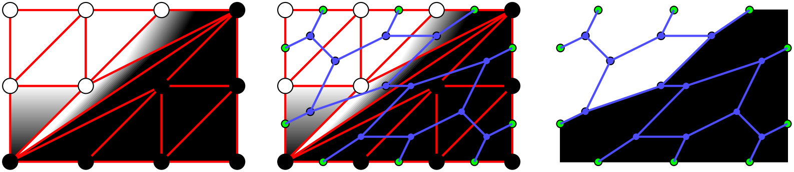
- Contours are in-between pixels $\Rightarrow$ Dual graph G of T
- One pixel per dual face $\Rightarrow$ One color per dual face

Dual graph improved by a variational regularization process

- Dissimilarity weights on arcs $a = (p_i, p_j)$: $w_a = \|s(p_i) - s(p_j)\|$
- Barycenters b_i are made closer to arcs of T with strong w_a.
- 10 iterations for convergence

iteration 0

Contour extraction and regularization

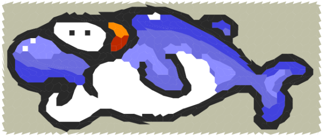


Primal T Dual graph G

- Contours are in-between pixels ⇒ Dual graph G of T
- One pixel per dual face ⇒ One color per dual face

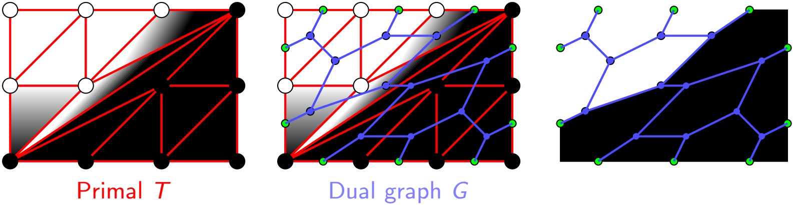
Dual graph improved by a variational regularization process

- Dissimilarity weights on arcs $a = (p_i, p_j)$: $w_a = \|s(p_i) - s(p_j)\|$
- Barycenters b_i are made closer to arcs of T with strong w_a.
- 10 iterations for convergence



iteration 1

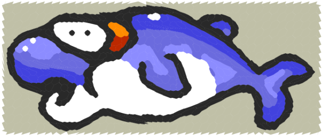
Contour extraction and regularization



- Contours are in-between pixels $\Rightarrow$ Dual graph G of T
- One pixel per dual face $\Rightarrow$ One color per dual face

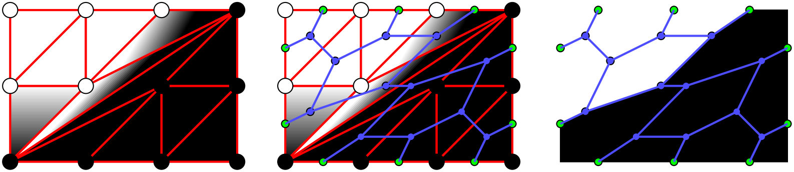
Dual graph improved by a variational regularization process

- Dissimilarity weights on arcs $a = (p_i, p_j)$: $w_a = \|s(p_i) - s(p_j)\|$
- Barycenters b_i are made closer to arcs of T with strong w_a.
- 10 iterations for convergence



iteration 2

Contour extraction and regularization



Primal T Dual graph G

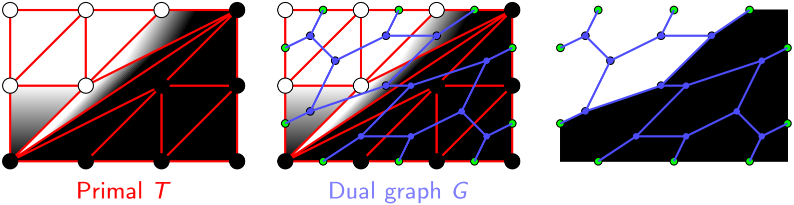
- Contours are in-between pixels $\Rightarrow$ Dual graph G of T
- One pixel per dual face $\Rightarrow$ One color per dual face

Dual graph improved by a variational regularization process

- Dissimilarity weights on arcs $a = (p_i, p_j)$: $w_a = \|s(p_i) - s(p_j)\|$
- Barycenters b_i are made closer to arcs of T with strong w_a.
- 10 iterations for convergence

iteration 3

Contour extraction and regularization



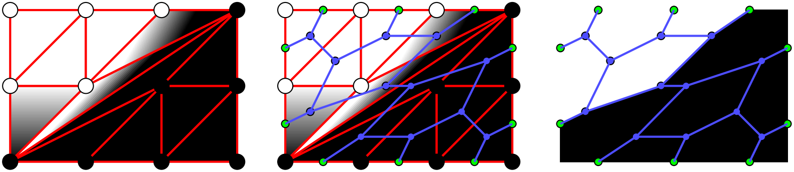
- Contours are in-between pixels $\Rightarrow$ Dual graph G of T
- One pixel per dual face $\Rightarrow$ One color per dual face

Dual graph improved by a variational regularization process

- Dissimilarity weights on arcs $a = (p_i, p_j)$: $w_a = \|s(p_i) - s(p_j)\|$
- Barycenters b_i are made closer to arcs of T with strong w_a.
- 10 iterations for convergence

iteration 4

Contour extraction and regularization



Primal T Dual graph G

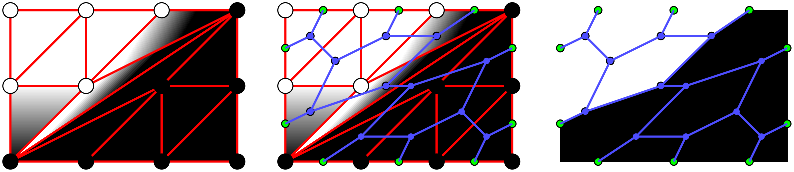
- Contours are in-between pixels ⇒ Dual graph G of T
- One pixel per dual face ⇒ One color per dual face

Dual graph improved by a variational regularization process

- Dissimilarity weights on arcs $a = (p_i, p_j)$: $w_a = \|s(p_i) - s(p_j)\|$
- Barycenters b_i are made closer to arcs of T with strong w_a.
- 10 iterations for convergence

iteration 5

Contour extraction and regularization



Primal T Dual graph G

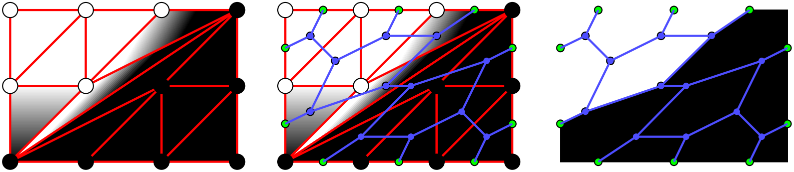
- Contours are in-between pixels ⇒ Dual graph G of T
- One pixel per dual face ⇒ One color per dual face

Dual graph improved by a variational regularization process

- Dissimilarity weights on arcs $a = (p_i, p_j)$: $w_a = \|s(p_i) - s(p_j)\|$
- Barycenters b_i are made closer to arcs of T with strong w_a.
- 10 iterations for convergence

iteration 6

Contour extraction and regularization



Primal T Dual graph G

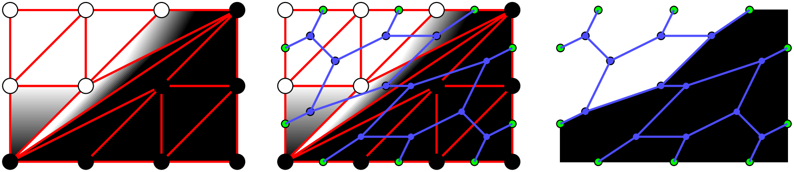
- Contours are in-between pixels ⇒ Dual graph G of T
- One pixel per dual face ⇒ One color per dual face

Dual graph improved by a variational regularization process

- Dissimilarity weights on arcs $a = (p_i, p_j)$: $w_a = \|s(p_i) - s(p_j)\|$
- Barycenters b_i are made closer to arcs of T with strong w_a.
- 10 iterations for convergence

iteration 7

Contour extraction and regularization



Primal T Dual graph G

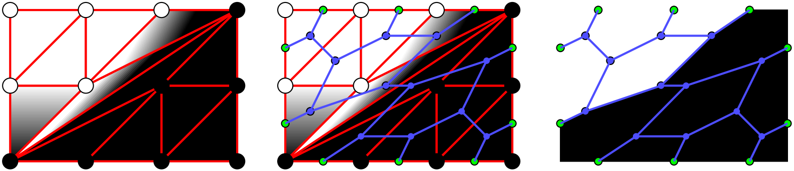
- Contours are in-between pixels $\Rightarrow$ Dual graph G of T
- One pixel per dual face $\Rightarrow$ One color per dual face

Dual graph improved by a variational regularization process

- Dissimilarity weights on arcs $a = (p_i, p_j)$: $w_a = \|s(p_i) - s(p_j)\|$
- Barycenters b_i are made closer to arcs of T with strong w_a.
- 10 iterations for convergence

iteration 8

Contour extraction and regularization



Primal T Dual graph G

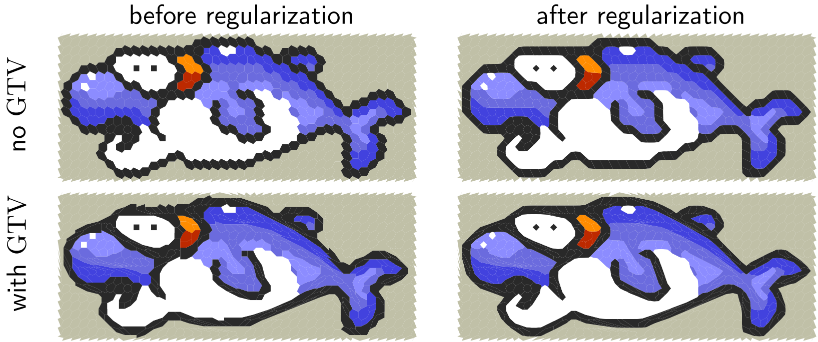
- Contours are in-between pixels ⇒ Dual graph G of T
- One pixel per dual face ⇒ One color per dual face

Dual graph improved by a variational regularization process

- Dissimilarity weights on arcs $a = (p_i, p_j)$: $w_a = \|s(p_i) - s(p_j)\|$
- Barycenters b_i are made closer to arcs of T with strong w_a.
- 10 iterations for convergence

iteration 9

Importance of both GTV and contour regularization



Geometric Total Variation for Image Vectorization, Zooming and Pixel Art Depixelizing

Context and Motivation

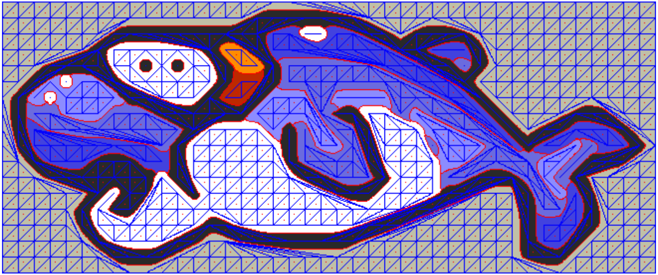
Variational formulation of Image Structuration

Vectorized contour regularization

Zoomed image with smooth spline contours

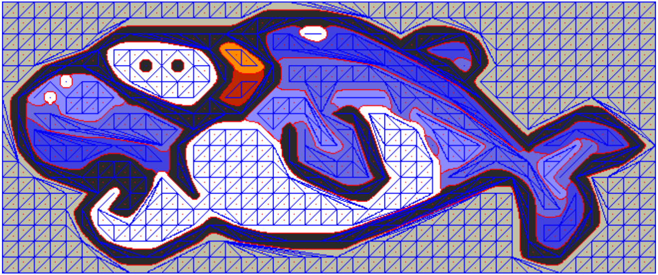
Experiments

Raster approach to spline contours



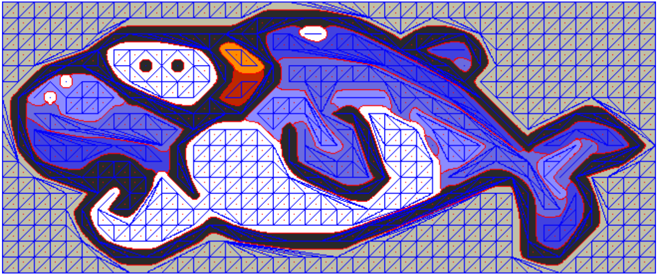
1. Draw a similarity graph S alongs arcs with weight $w_a = 0$

Raster approach to spline contours



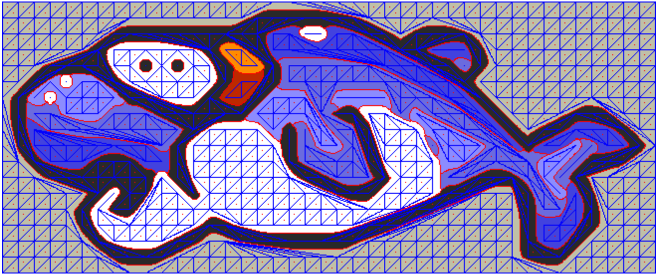
1. Draw a similarity graph S alongs arcs with weight $w_a = 0$
2. Point $\mathbf{x}_a$ are intersection of arc a with regularized contour

Raster approach to spline contours



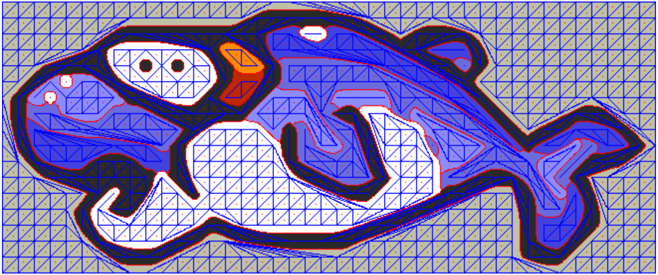
1. Draw a similarity graph S alongs arcs with weight $w_a = 0$
2. Point $\mathbf{x}_a$ are intersection of arc a with regularized contour
3. Draw a contour graph C with Bezier curves between dissimilar $\mathbf{x}$ in each triangle

Raster approach to spline contours



1. Draw a similarity graph S alongs arcs with weight $w_a = 0$
2. Point $\mathbf{x}_a$ are intersection of arc a with regularized contour
3. Draw a contour graph C with Bezier curves between dissimilar $\mathbf{x}$ in each triangle
4. Voronoi maps for S and C: each pixel p has closest point on S and C

Raster approach to spline contours

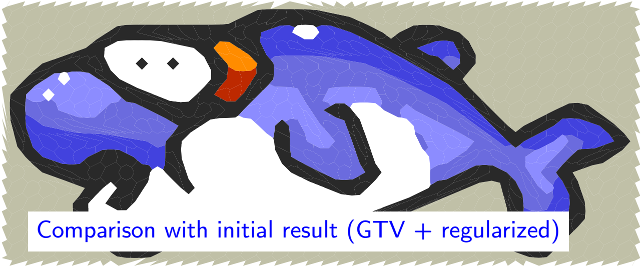


1. Draw a similarity graph S alongs arcs with weight $w_a = 0$
2. Point $\mathbf{x}_a$ are intersection of arc a with regularized contour
3. Draw a contour graph C with Bezier curves between dissimilar $\mathbf{x}$ in each triangle
4. Voronoi maps for S and C: each pixel p has closest point on S and C
5. Interpolation between S and C: color(pixel p) = mix color closest points on S and C, according to distances and some stiffness parameter β.

Raster approach to spline contours

1. Draw a similarity graph S alongs arcs with weight $w_a = 0$
2. Point $\mathbf{x}_a$ are intersection of arc a with regularized contour
3. Draw a contour graph C with Bezier curves between dissimilar $\mathbf{x}$ in each triangle
4. Voronoi maps for S and C: each pixel p has closest point on S and C
5. Interpolation between S and C: color(pixel p) = mix color closest points on S and C, according to distances and some stiffness parameter β.
6. Colors on original pixels are kept !

Raster approach to spline contours



1. Draw a similarity graph S alongs arcs with weight $w_a = 0$
2. Point $\mathbf{x}_a$ are intersection of arc a with regularized contour
3. Draw a contour graph C with Bezier curves between dissimilar $\mathbf{x}$ in each triangle
4. Voronoi maps for S and C: each pixel p has closest point on S and C
5. Interpolation between S and C: color(pixel p) = mix color closest points on S and C, according to distances and some stiffness parameter β.
6. Colors on original pixels are kept !

Geometric Total Variation for Image Vectorization, Zooming and Pixel Art Depixelizing

Context and Motivation

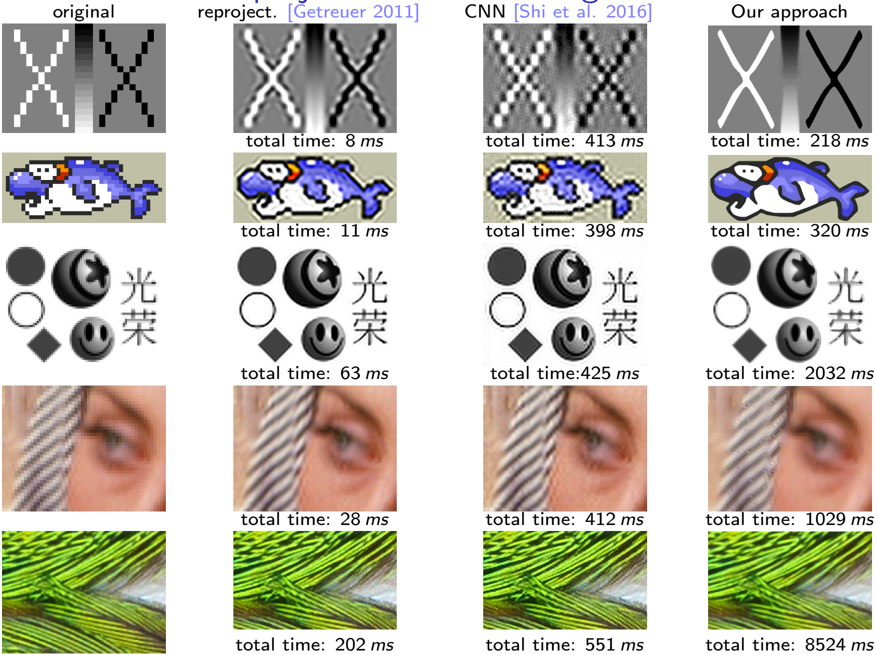
Variational formulation of Image Structuration

Vectorized contour regularization

Zoomed image with smooth spline contours

Experiments

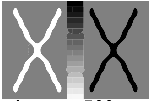
Zoomed ×16 vs reprojection and learning methods



Other comparisons

Depixelizing [Kopf, Lischinsky]

time: ≈ 500 *ms*



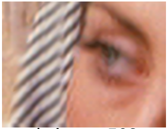
time: ≈ 500 *ms*

vector magic [vectorMagic10]

time: ≈ 2000 *ms*

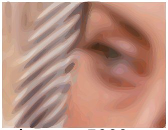
time: ≈ 2000 *ms*

Roussos-Maragos [Roussos-Maragios]



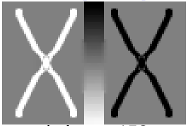
total time : 503 *ms*

Potrace [Potrace]



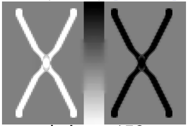
time: ≈ 5000 *ms*

Hq4x [Stepin]



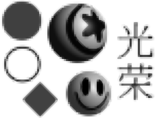
total time : 158 *ms*

Hq4x(*Hq4x*) [Stepin]



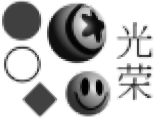
total time: 152 *ms*

Hq4x [Stepin]



total time : 142 *ms*

Hq4x(*Hq4x*) [Stepin]



total time: 320 *ms*

Pixel art / quantified images

Play yourself: online demonstrators

- Our code with GitHub Repository:

  https://github.com/kerautret/GTVimageVect

- Our online demonstration :
  https://ipolcore.ipol.im/demo/clientApp/demo.html?id=280

Play yourself: online demonstrators

- Our code with GitHub Repository:

  https://github.com/kerautret/GTVimageVect

- Our online demonstration :
  https://ipolcore.ipol.im/demo/clientApp/demo.html?id=280

- Other demonstrations of compared works can be reproduced online:
  - Convolutional Neural Network for Subpixel Super-Resolution [Shi et al. 2016] : https://ipolcore.ipol.im/demo/clientApp/demo.html?id=77777000078
  - Super resolution with HQx Algorithm [Stepin] : https://ipolcore.ipol.im/demo/clientApp/demo.html?id=77777000079
  - Image Interpolation with Geometric Contour Stencils [Getreuer 2011] : http://demo.ipol.im/demo/g_interpolation_geometric_contour_stencils
  - Vector-valued image interpolation by an anisotropic diffusion-projection PDE [Roussos-Maragios] : http://demo.ipol.im/demo/g_roussos_diffusion_interpolation
  - [Vector Magic Inc] : http://vectormagic.com.
  - Depixelizing [Kopf, Lischinsky] and Potrace [Potrace] through Inkscape: https://inkscape.org/

Conclusion

- A sound variational model for image zooming, image vectorization and pixel art depixelizing
- Reproducible research, online demos
- Future works include: quantitative error analysis, improved energy model, GPU implementation for real-time

Visit https://github.com/kerautret/GTVimageVect !

Conclusion

- A sound variational model for image zooming, image vectorization and pixel art depixelizing
- Reproducible research, online demos
- Future works include: quantitative error analysis, improved energy model, GPU implementation for real-time

Visit https://github.com/kerautret/GTVimageVect !

Thank you for your attention !
Any questions ?

References I

Pascal Getreuer.
Contour Stencils: Total Variation along Curves for Adaptive Image Interpolation.
SIAM J. on Imaging Sciences, 4(3):954–979, January 2011.

Pascal Getreuer.
Image Interpolation with Geometric Contour Stencils.
Image Processing On Line, 1:98–116, September 2011.

Pascal Getreuer.
Roussos-Maragos Tensor-Driven Diffusion for Image Interpolation.
Image Processing On Line, 1:178–186, September 2011.

Vector Magic Inc.
Vector magic, 2010.
http://vectormagic.com.

Johannes Kopf and Dani Lischinski.
Depixelizing Pixel Art.
In *ACM SIGGRAPH 2011 Papers*, pages 99:1–99:8, 2011.
Vancouver, British Columbia, Canada.

Anastasios Roussos and Petros Maragos.
Vector-valued image interpolation by an anisotropic diffusion-projection PDE.
In *International Conference on Scale Space and Variational Methods in Computer Vision*, pages 104–115. Springer, 2007.

References II

B. Kerautret and J.-O. Lachaud.
Github repository, 2019.
https://github.com/kerautret/GTVimageVect.

Peter Selinger.
Potrace, 2001-2017.
http://potrace.sourceforge.net.

Wenzhe Shi, Jose Caballero, Ferenc Huszár, Johannes Totz, Andrew P Aitken, Rob Bishop, Daniel Rueckert, and Zehan Wang.
Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network.
In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1874–1883, 2016.

M. Stepin.
Hqx magnification filter, 2003.
http://web.archive.org/web/20070717064839/www.hiend3d.com/hq4x.html.