

Approximations of the The Mumford-Shah Functional for Unit Vector Fields

Jonas Haehnle

University of Tübingen

Warwick, Jan 2009

イロン 不得 とくほ とくほとう

3

Introduction The Functional

Motivation

Algorithms

Splitting & Projection Splitting & Penalisation Penalisation without Splitting

Examples

Comparison: Projection vs. Penalisation Comparison: CB vs. RGB

ヘロト 人間 ト ヘヨト ヘヨト

ъ

The Functional Motivation

Vectorial Mumford-Shah Functional (sphere valued functions)

For $\Omega \subseteq \mathbb{R}^d$, $d \ge 2$, find $(\mathbf{v}, \Gamma) := \underset{\mathbf{u} \in H^1(\Omega \setminus K, \mathbb{S}^2)}{\operatorname{argmin}} E(\mathbf{u}, K),$ $E(\mathbf{u}, K) := \frac{\alpha}{2} \int_{\Omega \setminus K} |\nabla \mathbf{u}|^2 d\mathbf{x} + \frac{\gamma}{2} \int_{\Omega} |\mathbf{u} - \mathbf{g}|^2 d\mathbf{x} + \beta \mathscr{H}^{d-1}(K),$

 $\mathbf{g} \in L^{\infty}(\Omega, \mathbb{S}^2)$: original image, K: edge set (closed).

・ロト・日本・日本・日本・日本・日本・日本

The Functional Motivation

◆□▶ ◆□▶ ◆三▶ ◆三▶ ● ● ●

Vectorial Mumford-Shah Functional (sphere valued functions)

For $\Omega \subseteq \mathbb{R}^d$, $d \ge 2$, find $(\mathbf{v}, \Gamma) := \underset{\mathbf{u} \in H^1(\Omega \setminus K, \mathbb{S}^2)}{\operatorname{argmin}} E(\mathbf{u}, K),$ $E(\mathbf{u}, K) := \frac{\alpha}{2} \int_{\Omega \setminus K} |\nabla \mathbf{u}|^2 d\mathbf{x} + \frac{\gamma}{2} \int_{\Omega} |\mathbf{u} - \mathbf{g}|^2 d\mathbf{x} + \beta \mathscr{H}^{d-1}(K),$ $\mathbf{g} \in L^{\infty}(\Omega, \mathbb{S}^2)$: original image, K: edge set (closed).

3 Terms: smoothing, fidelity, edge length.

The Functional Motivation

イロト イポト イヨト イヨト

3

Weak Formulation

(sphere valued functions)

For $\mathbf{g} \in L^{\infty}(\Omega; \mathbb{S}^2)$, $\mathbf{u} \in SBV(\Omega, \mathbb{S}^2)$, minimise

$$\overline{E}(\mathbf{u}) := \frac{\alpha}{2} \int_{\Omega} |\nabla \mathbf{u}|^2 d\mathbf{x} + \frac{\gamma}{2} \int_{\Omega} |\mathbf{u} - \mathbf{g}|^2 d\mathbf{x} + \beta \mathscr{H}^{d-1}(S_{\mathbf{u}} \cap \Omega).$$

The Functional Motivation

イロン イボン イヨン イヨン

æ

Weak Formulation

(sphere valued functions)

For $\mathbf{g} \in L^{\infty}(\Omega; \mathbb{S}^2)$, $\mathbf{u} \in SBV(\Omega, \mathbb{S}^2)$, minimise

$$\overline{E}(\mathbf{u}) := \frac{\alpha}{2} \int_{\Omega} |\nabla \mathbf{u}|^2 \mathrm{d}\mathbf{x} + \frac{\gamma}{2} \int_{\Omega} |\mathbf{u} - \mathbf{g}|^2 \mathrm{d}\mathbf{x} + \beta \mathscr{H}^{d-1}(S_{\mathbf{u}} \cap \Omega).$$

Thm: (Carriero-Leaci, 91): \exists solution **u**, and $(\mathbf{u}, \overline{S}_{\mathbf{u}})$ solves original problem.

The Functional Motivation

Vectorial Ambrosio-Tortorelli Functional (sphere valued functions)

For $\mathbf{u}, \mathbf{g} \in H^1(\Omega; \mathbb{S}^2)$, $s \in H^1(\Omega, [0, 1])$, minimise

$$\begin{array}{ll} \mathcal{A} \mathcal{T}_{\varepsilon}(\mathbf{u},s) &:= & \displaystyle \frac{\alpha}{2} \int_{\Omega} (s^2 + k_{\varepsilon}) |\nabla \mathbf{u}|^2 \mathrm{d} \mathbf{x} + \frac{\gamma}{2} \int_{\Omega} |\mathbf{u} - \mathbf{g}|^2 \mathrm{d} \mathbf{x} \\ & + \beta \int_{\Omega} \varepsilon |\nabla s|^2 + \frac{1}{4\varepsilon} (1-s)^2 \mathrm{d} \mathbf{x}. \end{array}$$

(ロ) (四) (三) (三) (三) (三) (○)

The Functional Motivation

Vectorial Ambrosio-Tortorelli Functional (sphere valued functions)

For $\mathbf{u}, \mathbf{g} \in H^1(\Omega; \mathbb{S}^2)$, $s \in H^1(\Omega, [0, 1])$, minimise $\begin{aligned} AT_{\varepsilon}(\mathbf{u}, s) &:= \quad \frac{\alpha}{2} \int_{\Omega} (s^2 + k_{\varepsilon}) |\nabla \mathbf{u}|^2 d\mathbf{x} + \frac{\gamma}{2} \int_{\Omega} |\mathbf{u} - \mathbf{g}|^2 d\mathbf{x} \\ &+ \beta \int_{\Omega} \varepsilon |\nabla s|^2 + \frac{1}{4\varepsilon} (1 - s)^2 d\mathbf{x}. \end{aligned}$

 $\textbf{\textit{s}}: \text{``phase function'', i.e. } \textbf{\textit{K}} \approx \{\textbf{\textit{s}} \approx \textbf{0}\}, \ \Omega \setminus \textbf{\textit{K}} \approx \{\textbf{\textit{s}} \approx \textbf{1}\}.$

▲□▶ ▲□▶ ▲臣▶ ▲臣▶ ―臣 - のへで

The Functional Motivation

Vectorial Ambrosio-Tortorelli Functional (sphere valued functions)

For $\mathbf{u}, \mathbf{g} \in H^1(\Omega; \mathbb{S}^2)$, $s \in H^1(\Omega, [0, 1])$, minimise $AT_{\varepsilon}(\mathbf{u}, s) := \frac{\alpha}{2} \int_{\Omega} (s^2 + k_{\varepsilon}) |\nabla \mathbf{u}|^2 d\mathbf{x} + \frac{\gamma}{2} \int_{\Omega} |\mathbf{u} - \mathbf{g}|^2 d\mathbf{x} + \beta \int_{\Omega} \varepsilon |\nabla s|^2 + \frac{1}{4\varepsilon} (1 - s)^2 d\mathbf{x}.$

 $\textbf{\textit{s}}: \text{``phase function'', i.e. } \textbf{\textit{K}} \approx \{\textbf{\textit{s}} \approx \textbf{0}\}, \ \Omega \setminus \textbf{\textit{K}} \approx \{\textbf{\textit{s}} \approx \textbf{1}\}.$

Thm (Ambrosio-Tortorelli, 90/92): $0 < k_{\varepsilon} = o(\varepsilon) \Longrightarrow AT_{\varepsilon}(\mathbf{u}, s) \xrightarrow{\Gamma} \overline{E}(\mathbf{u}, K).$

(ロ) (四) (三) (三) (三) (○)

The Functiona Motivation

Prototype Problem

Non-convex Functional (Mumford-Shah).



The Functional Motivation

・ロン・西方・ ・ ヨン・

3

Prototype Problem

- Non-convex Functional (Mumford-Shah).
- ▶ Non-convex constraint ($\mathbf{u} \in \mathbb{S}^2$).

イロト イポト イヨト イヨト

3

Prototype Problem

- Non-convex Functional (Mumford-Shah).
- ▶ Non-convex constraint ($\mathbf{u} \in \mathbb{S}^2$).
- ▶ With AT: Strong nonlinear coupling between **u** and *s*.

The Functional Motivation

ヘロト ヘアト ヘビト ヘビト

Prototype Problem

- Non-convex Functional (Mumford-Shah).
- ▶ Non-convex constraint ($\mathbf{u} \in \mathbb{S}^2$).
- ▶ With AT: Strong nonlinear coupling between **u** and *s*.
- Extending work on harmonic maps to S² (convex functional, non-convex constraint): Alouges (97), Bartels (05), Bartels-Prohl (07).

Colour Image Segmentation

Chromaticity & Brightness (CB) Colour Model:



イロト イポト イヨト イヨト

3

Colour Image Segmentation

► Chromaticity & Brightness (CB) Colour Model: **RGB** image $\mathbf{g} : \Omega \rightarrow [0, 255]^3 \subset \mathbb{N}^3$:

•
$$\mathbf{C} := \mathbf{g}/|\mathbf{g}| = \mathbf{g}/B : \Omega \to \mathbb{S}^2.$$

Colour Image Segmentation

- Chromaticity & Brightness (CB) Colour Model: RGB image g : Ω → [0,255]³ ⊂ N³:
 - ► *B* := |**g**|, and
 - $\mathbf{C} := \mathbf{g}/|\mathbf{g}| = \mathbf{g}/B : \Omega \to \mathbb{S}^2.$

 Sources: Mumford-Shah (89) Chan-Kang-Shen (01): TV, CB Tang-Sapiro-Caselles (01): Harmonic flow, CB Osher-Vese (04): *p*-harmonic flow, CB Bartels-Prohl (06): PM, CB Bellettini-Coscia (94), Bourdin (99): MS, grayscale

The Functional Motivation

Nematic Liquid Crystals

Simplified Ericksen's energy

$$\int_{\Omega} \frac{1}{2} s^2 |\nabla \mathbf{n}|^2 + |\nabla s|^2 + W_0(s) \,\mathrm{d}\mathbf{x},$$

2

Nematic Liquid Crystals

Simplified Ericksen's energy

$$\int_{\Omega} \frac{1}{2} s^2 |\nabla \mathbf{n}|^2 + |\nabla s|^2 + W_0(s) d\mathbf{x},$$

s ∈ [-1/2, 1]: degree of orientation (often s ≥ 0),
 n ∈ S²: director.

Nematic Liquid Crystals

Simplified Ericksen's energy

$$\int_{\Omega} \frac{1}{2} s^2 |\nabla \mathbf{n}|^2 + |\nabla s|^2 + W_0(s) d\mathbf{x},$$

・ロト ・聞 ト ・ ヨト ・ ヨト … ヨ

- s ∈ [-1/2,1]: degree of orientation (often s ≥ 0),
 n ∈ S²: director.
- Sources: Lin (89), Lin-Luskin (89)
 Virga (94), Alouges (97), Bartels-Prohl (07)

Algorithm 1: Splitting & Projection

FE Discr.
$$\rightarrow 0$$
Alouges-Splitting $?$ $h \rightarrow 0$ $h \rightarrow 0$ $n \rightarrow \infty$ AT on \mathbb{S}^2 Γ $MS \text{ on } \mathbb{S}^2$

▲口▶▲圖▶▲臣▶▲臣▶ 臣 のなぐ

Algorithm 1: Splitting & Projection

Easy to solve numerically. Exact sphere constraint.

・ロト ・聞 ト ・ ヨト ・ ヨト

ъ

Introduction Splitt Algorithms Splitt Examples Pena

Splitting & Projection Splitting & Penalisation Penalisation without Splitting

ъ

Algorithm 1: Splitting & Projection

Splitting: In every iteration, minimise first for **u**, then for *s*.

イロト イポト イヨト イヨト

- Splitting: In every iteration, minimise first for **u**, then for *s*.
- Ist step, based on Alouges (97): Look for update w s.t. AT_ε(u − w, s) ≤ AT_ε(u, s).

イロト イポト イヨト イヨト

- Splitting: In every iteration, minimise first for **u**, then for *s*.
- Ist step, based on Alouges (97): Look for update w s.t. AT_ε(u − w, s) ≤ AT_ε(u, s).
- Only take **w** s.t. $\mathbf{w} \perp \mathbf{u}$ a.e.

・ロト ・ 一下・ ・ ヨト・

- Splitting: In every iteration, minimise first for **u**, then for *s*.
- Ist step, based on Alouges (97): Look for update w s.t. AT_ε(u − w, s) ≤ AT_ε(u, s).
- Only take **w** s.t. $\mathbf{w} \perp \mathbf{u}$ a.e. \implies
 - projecting u w never increases energy
 - existence (**u** · **w** = 0 is linear constraint)

◆□▶ ◆□▶ ◆臣▶ ◆臣▶

- Splitting: In every iteration, minimise first for **u**, then for *s*.
- Ist step, based on Alouges (97): Look for update w s.t. AT_ε(u − w, s) ≤ AT_ε(u, s).
- Only take **w** s.t. $\mathbf{w} \perp \mathbf{u}$ a.e. \implies
 - projecting u w never increases energy
 - existence ($\mathbf{u} \cdot \mathbf{w} = 0$ is linear constraint)
- 2nd step: ∃ by convexity & coercivity, 0 ≤ s ≤ 1 by max-principle (or testing with cutoff functions).

Properties

Energy decreasing.



・ロト ・聞 ト ・ ヨト ・ ヨトー

3

Properties

- Energy decreasing.
- $(\mathbf{u}_n, s_n) \rightarrow (\mathbf{u}, s)$ in H^1 (up to subseq).

イロト イポト イヨト イヨト

æ

Properties

- Energy decreasing.
- $(\mathbf{u}_n, s_n) \rightarrow (\mathbf{u}, s)$ in H^1 (up to subseq).
- $\mathbf{u}_n \rightarrow \mathbf{u}$ in L^2 , and $\mathbf{w}_n \rightarrow 0$ in H^1 .

イロト イポト イヨト イヨト

æ

Properties

- Energy decreasing.
- $(\mathbf{u}_n, s_n) \rightarrow (\mathbf{u}, s)$ in H^1 (up to subseq).
- $\mathbf{u}_n \rightarrow \mathbf{u}$ in L^2 , and $\mathbf{w}_n \rightarrow 0$ in H^1 .
- Limit fulfils constraints.

Properties

- Energy decreasing.
- $(\mathbf{u}_n, s_n) \rightarrow (\mathbf{u}, s)$ in H^1 (up to subseq).
- $\mathbf{u}_n \rightarrow \mathbf{u}$ in L^2 , and $\mathbf{w}_n \rightarrow 0$ in H^1 .
- Limit fulfils constraints.
- Problem: Cannot identify limit

$$\lim_{n\to\infty} \left(|\nabla \mathbf{u}_{n+1}|^2 s_n, \varphi \right),\,$$

イロト イヨト イヨト イ

- ⊒ →

so cannot show (**u**, *s*) stationary point of $AT_{\varepsilon}(\cdot, \cdot)$.

3

Discretisation

 Polygonal Lipschitz domain, uniform triangulation, continuous, piecewise affine FE.

イロト イポト イヨト イヨト

æ

- Polygonal Lipschitz domain, uniform triangulation, continuous, piecewise affine FE.
- Only linear eqns in every iteration.

(日) (四) (日) (日) (日)

- Polygonal Lipschitz domain, uniform triangulation, continuous, piecewise affine FE.
- Only linear eqns in every iteration.
- Energy decreasing, if all int. angles $\leq \pi/2$ (AC).

イロト イヨト イヨト イ

- ⊒ →

- Polygonal Lipschitz domain, uniform triangulation, continuous, piecewise affine FE.
- Only linear eqns in every iteration.
- Energy decreasing, if all int. angles $\leq \pi/2$ (AC).
- Sphere constraint exactly fulfilled on nodal points.

ヘロト ヘアト ヘヨト ヘ

- Polygonal Lipschitz domain, uniform triangulation, continuous, piecewise affine FE.
- Only linear eqns in every iteration.
- Energy decreasing, if all int. angles $\leq \pi/2$ (AC).
- Sphere constraint exactly fulfilled on nodal points.
- Limit fulfils constraints.
Introduction Splitting & Projection Algorithms Splitting & Penalisation Examples Penalisation without Splitting

ヘロト ヘアト ヘヨト ヘ

- ⊒ →

Discretisation

- Polygonal Lipschitz domain, uniform triangulation, continuous, piecewise affine FE.
- Only linear eqns in every iteration.
- Energy decreasing, if all int. angles $\leq \pi/2$ (AC).
- Sphere constraint exactly fulfilled on nodal points.
- Limit fulfils constraints.
- Problem: As before.

Algorithm 2: Splitting & Penalisation

$$\begin{array}{c} \mbox{FE Discr.} & \xrightarrow[h \to 0]{} \mbox{Splitting} & \xrightarrow[n \to \infty]{} \\ \hline \mbox{Reg. GL-AT} & \xrightarrow[\varepsilon \to 0]{} \mbox{MS on } \mbox{S}^2 \end{array}$$

◆□ > ◆□ > ◆臣 > ◆臣 > ─臣 ─のへで

Algorithm 2: Splitting & Penalisation

$$\begin{array}{c} \mbox{FE Discr.} & \xrightarrow[h \to 0]{} \mbox{Splitting} & \xrightarrow[n \to \infty]{} \\ \mbox{Reg. GL-AT} & \xrightarrow[\epsilon \to 0]{} \mbox{MS on } \mathbb{S}^2 \end{array}$$

Harder to solve numerically. Approx. sphere constraint.

ъ

Algorithm 2: Splitting & Penalisation

Splitting again.



Algorithm 2: Splitting & Penalisation

- Splitting again.
- No projection, but Ginzburg-Landau penalisation: add

$$\frac{1}{\delta_{\varepsilon}}\int_{\Omega}\left(|\mathbf{u}|^2-\mathbf{1}
ight)^2\mathrm{d}\mathbf{x}.$$

æ

Algorithm 2: Splitting & Penalisation

- Splitting again.
- No projection, but Ginzburg-Landau penalisation: add

$$\frac{1}{\delta_{\varepsilon}}\int_{\Omega}\left(|\mathbf{u}|^2-\mathbf{1}\right)^2\mathrm{d}\mathbf{x}.$$

Identifying limits needs higher regularity of s, so add

$$rac{\eta_{arepsilon}}{
ho}\int_{\Omega}| riangle s|^{
ho}\mathsf{d}\mathbf{x}.$$

▲□▶ ▲□▶ ▲目▶ ▲目▶ 三目 - 釣A@

イロト イポト イヨト イヨト

ъ

Properties

► Γ-convergence for proper scalings of δ_ε and η_ε (δ_ε slower than ε, η_ε much faster).

イロト イポト イヨト イヨト

- Γ-convergence for proper scalings of δ_ε and η_ε
 (δ_ε slower than ε, η_ε much faster).
- Solutions exist in each step.

Introduction Splitting & Projection Algorithms Splitting & Penalisation Examples Penalisation without Splitting

イロト イポト イヨト イヨト

- ► Γ-convergence for proper scalings of δ_ε and η_ε (δ_ε slower than ε, η_ε much faster).
- Solutions exist in each step.
- Energy decreasing.

Introduction Splitting & Projection Algorithms Splitting & Penalisation Examples Penalisation without Splitting

イロト イポト イヨト イヨト

- Γ-convergence for proper scalings of δ_ε and η_ε
 (δ_ε slower than ε, η_ε much faster).
- Solutions exist in each step.
- Energy decreasing.
- $(\mathbf{u}_n, s_n) \rightharpoonup (\mathbf{u}, s)$ in $H^2 \times W^{2,p}$ (up to subseq).

(日) (四) (日) (日) (日)

- Γ-convergence for proper scalings of δ_ε and η_ε
 (δ_ε slower than ε, η_ε much faster).
- Solutions exist in each step.
- Energy decreasing.
- $(\mathbf{u}_n, s_n) \rightharpoonup (\mathbf{u}, s)$ in $H^2 \times W^{2,p}$ (up to subseq).
- For p > d, u_n → u in H¹, and s_n → s in L[∞], so identifying limits works.

Discretisation/Disadvantages

► *s_n* now needs hermite elements (but...).



イロト イポト イヨト イヨト

ъ

Discretisation/Disadvantages

- ► *s*_n now needs hermite elements (but...).
- Nonlinear eqns need to be solved.

Introduction Splitting & Projection Algorithms Splitting & Penalisation Examples Penalisation without Splittin

イロト イ理ト イヨト イヨト

Discretisation/Disadvantages

- ► *s_n* now needs hermite elements (but...).
- Nonlinear eqns need to be solved.
- Sphere constraint only approximated.

Introduction Splitting & Projection Algorithms Splitting & Penalisation Examples Penalisation without Splittin

ヘロト ヘアト ヘビト ヘ

- ⊒ →

Discretisation/Disadvantages

- ► *s*_n now needs hermite elements (but...).
- Nonlinear eqns need to be solved.
- Sphere constraint only approximated.
- Two additional parameters.

・ロト ・聞 ト ・ ヨト ・ ヨトー

2

Algorithm 3: Penalisation without Splitting

FE Discr.
$$\xrightarrow[h\to 0]{\Gamma}$$
 MS on \mathbb{S}^2

Algorithm 3: Penalisation without Splitting

FE Discr.
$$\xrightarrow[h\to 0]{\Gamma}$$
 MS on \mathbb{S}^2

Hard to solve numerically. Approx. sphere constraint.

イロト イポト イヨト イヨト

Penalisation without Splitting

No splitting.



・ロト ・聞 ト ・ ヨト ・ ヨト

ъ

Penalisation without Splitting

- No splitting.
- No projection, but Ginzburg-Landau penalisation.

イロト イポト イヨト イヨト

æ

Penalisation without Splitting

- No splitting.
- No projection, but Ginzburg-Landau penalisation.
- Direct Γ-convergence of 1st-order FE-functional (as in Bellettini-Coscia, 94).

Introduction Splitting & Projection Algorithms Splitting & Penalisation Examples Penalisation without Splitting

Penalisation without Splitting

- No splitting.
- No projection, but Ginzburg-Landau penalisation.
- Direct Γ-convergence of 1st-order FE-functional (as in Bellettini-Coscia, 94).
- Advantage: No additional regularisation term needed.

イロト イポト イヨト イヨト

Introduction Splitting & Projection Algorithms Splitting & Penalisation Examples Penalisation without Splitting

Penalisation without Splitting

- No splitting.
- No projection, but Ginzburg-Landau penalisation.
- Direct Γ-convergence of 1st-order FE-functional (as in Bellettini-Coscia, 94).
- Advantage: No additional regularisation term needed.

イロト イポト イヨト イヨト

Disadvantage: No convergent algorithm known.

Comparison: Projection vs. Penalisation Comparison: CB vs. RGB

ヘロト 人間 とくほとくほとう

æ

Projection



Comparison: Projection vs. Penalisation Comparison: CB vs. RGB

イロト イポト イヨト イヨト

ъ

Projection



Figure: $\mathbf{U}_0 \equiv \mathbf{g}$ (left) and \mathbf{U}_{10} (right).

Comparison: Projection vs. Penalisation Comparison: CB vs. RGB

Projection



Comparison: Projection vs. Penalisation Comparison: CB vs. RGB

Projection



Figure: Section at x = 0.5 through $U_0 \equiv g$ and U_{10} .

◆□ > ◆□ > ◆三 > ◆三 > ・三 ・ のへぐ

Comparison: Projection vs. Penalisation Comparison: CB vs. RGB

Projection vs. Penalisation



Comparison: Projection vs. Penalisation Comparison: CB vs. RGB

Projection vs. Penalisation



Figure: S_{10} , projection (left) and penalisation (right).

|▲□▶▲圖▶▲≣▶▲≣▶ = 三 つの(

Comparison: Projection vs. Penalisation Comparison: CB vs. RGB

Projection vs. Penalisation



Comparison: Projection vs. Penalisation Comparison: CB vs. RGB

Projection vs. Penalisation



Figure: Section at y = 0.375 through S_{10} , projection (left) and penalisation (right).

▲ロト▲御▶▲臣▶▲臣▶ 臣 のの()

Comparison: Projection vs. Penalisation Comparison: CB vs. RGB

Penalisation



Figure: Min and max of $\|\mathbf{U}\|$.

◆□ → ◆□ → ◆三 → ◆三 → ● ◆ ● ◆ ● ◆

Comparison: Projection vs. Penalisation Comparison: CB vs. RGB

CB Model

$$\begin{array}{lll} \mathsf{AT}_{\varepsilon}(\mathbf{u},v,s) &:= & \displaystyle\frac{\alpha}{2} \int_{\Omega} (s^2 + k_{\varepsilon}) |\nabla \mathbf{u}|^2 \mathrm{d}\mathbf{x} + \displaystyle\frac{\gamma}{2} \int_{\Omega} |\mathbf{u} - \mathbf{g}|^2 \mathrm{d}\mathbf{x} \\ & + \displaystyle\frac{\alpha_1}{2} \int_{\Omega} (s^2 + k_{\varepsilon}) |\nabla v|^2 \mathrm{d}\mathbf{x} + \displaystyle\frac{\gamma_1}{2} \int_{\Omega} |v - b|^2 \mathrm{d}\mathbf{x} \\ & + \displaystyle\beta \int_{\Omega} \varepsilon |\nabla s|^2 + \displaystyle\frac{1}{4\varepsilon} (1 - s)^2 \mathrm{d}\mathbf{x}, \end{array}$$

◆□▶ ◆□▶ ◆目▶ ◆目▶ ○目・ のへぐ

Comparison: Projection vs. Penalisation Comparison: CB vs. RGB

イロン 不得 とくほ とくほとう

ъ

CB Model

$$\begin{array}{lll} \mathsf{AT}_{\varepsilon}(\mathbf{u},v,s) &:= & \displaystyle\frac{\alpha}{2} \int_{\Omega} (s^2 + k_{\varepsilon}) |\nabla \mathbf{u}|^2 \mathrm{d}\mathbf{x} + \displaystyle\frac{\gamma}{2} \int_{\Omega} |\mathbf{u} - \mathbf{g}|^2 \mathrm{d}\mathbf{x} \\ & + \displaystyle\frac{\alpha_1}{2} \int_{\Omega} (s^2 + k_{\varepsilon}) |\nabla v|^2 \mathrm{d}\mathbf{x} + \displaystyle\frac{\gamma_1}{2} \int_{\Omega} |v - b|^2 \mathrm{d}\mathbf{x} \\ & + \displaystyle\beta \int_{\Omega} \varepsilon |\nabla s|^2 + \displaystyle\frac{1}{4\varepsilon} (1 - s)^2 \mathrm{d}\mathbf{x}, \end{array}$$

- **g**, **u**: Original and iterate chromaticity,
- b, v: Original and iterate brightness,
- s: Joint edge-indicator.

Comparison: Projection vs. Penalisation Comparison: CB vs. RGB

CB



Comparison: Projection vs. Penalisation Comparison: CB vs. RGB

・ロト ・聞ト ・ヨト ・ヨト

æ

CB



Figure: $\mathbf{U}_0 \equiv \mathbf{G}$ (left) and \mathbf{U}_{10} (right).

Comparison: Projection vs. Penalisation Comparison: CB vs. RGB

CB vs. RGB


Comparison: Projection vs. Penalisation Comparison: CB vs. RGB

CB vs. RGB



Figure: Crop of U_{10} , CB (left) and RGB (right).

Comparison: Projection vs. Penalisation Comparison: CB vs. RGB

CB vs. RGB



Comparison: Projection vs. Penalisation Comparison: CB vs. RGB

CB vs. RGB



Figure: S_{10} , CB (left) and RGB (right).

Comparison: Projection vs. Penalisation Comparison: CB vs. RGB

Thank You