Derivative-Free Superiorization Using the Facet Model

Ran Davidi, Robert M. Haralick, Gabor T. Herman

Seminar on Image Processing and Computer Vision

September 16, 2010

Outline

Introduction

Facets and facet model

The facet model iterative procedure

Procedure overview

Mathematical development

Example: 3×3 neighborhoods

Block error calculation

Superiorization and the facet model

Convex feasibility problems

Superiorization using the facet model

Outline

Introduction

- Facets and facet model

The facet model iterative procedure

- Procedure overview
- Mathematical development
- Example: 3×3 neighborhoods
- Block error calculation

Superiorization and the facet model

- Convex feasibility problems
- Superiorization using the facet model

Facets and facet model

- This material is based on the paper:
  - ROBERT M. HARALICK AND LAYNE WATSON
  - A FACET MODEL FOR IMAGE DATA
  - *COMPUTER GRAPHICS AND IMAGE PROCESSING* 15, 113-129 (1981)
- Abstract:
  - Image processing algorithms implicitly or explicitly assume an idealized form for the image data on which they operate. The degree to which the observed data meets the assumed idealized form is typically not examined or accounted for. This causes processing errors often attributed to noise. In this paper we discuss a facet model for image data which has the potential for fitting the form of the real idealized image, and for describing how the observed image differs from the idealized form. It is also an appropriate form for a variety of image processing algorithms. We give a relaxation procedure, and prove its convergence, for determining an estimate of the ideal image from observed image data.

Facets and facet model

- The facet model assumes that the image is everywhere "simple."
- This means that the spatial domain of the image can be partitioned into connected regions called *facets* each of which satisfy certain gray level and shape constraints.
- The *gray level constraints* that are used are simple: The gray levels in each facet must be a polynomial function of the row and column coordinates of the pixels in the facet.
- In the illustrations of the Haralick and Watson paper the polynomial function is of degree zero, one, or two.

Facets and facet model

Example

- If we consider the gray levels as composing a surface above the pixels of the facet, then (for an ideal/error-free image) having a degree-one polynomial function, the surface is a sloped plane.
- This *sloped facet model* is the one on which we concentrate in this talk.

Facets and facet model

- The *shape constraints* are also simple: Each facet must be "smooth" in shape.
- Specifically, it is assumed that each facet is a union of $K \times K$ blocks of pixels, where K is an arbitrary but fixed positive odd integer. (Note: The oddness of K is assumed only for notational convenience. Everything that is discussed below can also be described for an even K, at the cost of some loss of simplicity in the presentation.)
- We now illustrate this for $K = 3$ on the blackboard.

Facets and facet model

- Let Z_r and Z_c be the row and column index set for the spatial domain of an image and let $K \times K$ be the selected block size.
- For any $(r,c) \in Z_r \times Z_c$ let $I(r,c)$ be the gray value of pixel (r,c).
- An image is said to be *ideal* in the *sloped facet model* if there is a partition $\pi = \{\pi_1, \ldots, \pi_T\}$ of the spatial domain of $Z_r \times Z_c$ into *facets* (unions of $K \times K$ blocks of pixels) together with real numbers α_t, β_t and γ_t, for $1 \leq t \leq T$, such that

$$I(r,c) = \alpha_t r + \beta_t c + \gamma_t, \tag{1}$$

  for every pixel $(r,c) \in \pi_t$.
- We assume that any *observed image* J is the corrupted version of an ideal image I in the sense that

$$J(r,c) = I(r,c) + \eta(r,c), \tag{2}$$

  where η is the *error* made in the observation. Rephrasing this in the terminology of the previous talks, being ideal is the assumed "common attribute" of images in our application area.

Outline

Introduction
- Facets and facet model

The facet model iterative procedure
- Procedure overview
- Mathematical development
- Example: 3×3 neighborhoods
- Block error calculation

Superiorization and the facet model
- Convex feasibility problems
- Superiorization using the facet model

Overview of the iterative FM procedure

The facet model suggests an iterative procedure, which we name *FM*, whose aim is to get from the observed image to the underlying ideal image. (Haralick and Watson provide a proof of convergence of the procedure; however, the mathematical sense in which the image to which the procedure converges satisfies the stated aim is not clear to me. Fortunately, this does not actually matter for the purpose of this talk, which is the development of a derivative-free superiorization algorithm for image reconstruction from projections based on the facet model.) The *iterative step* of FM is as follows.

For each pixel in the input image,

- for each of the at most K^2 different $K \times K$ blocks containing that pixel, fit the gray values in the block by a sloped plane,
- identify one of these blocks for which the error of the fit by the sloped plane is minimal,
- set the output gray value to be that gray value assigned by the sloped plane of the block of minimal error of the fit.

Now show illustrative images from Haralick and Watson.

Mathematical development of FM

- Consider now a $(2L+1)\times(2L+1)$ block with the upper left-hand corner pixel having local row-column coordinates $(-L,-L)$ and the lower right-hand corner pixel having local row-column coordinates (L,L). Let $J(r^*,c^*)$ be the gray value at row r^* and column c^* of the block in the input image of the iterative step of FM. **Warning:** These coordinates (r^*,c^*) are *local* relative to the central pixel of the block. (Previously we have been using *global* coordinates (r,c) relative to the whole image.)

- For any block entirely contained in a facet, we assume (based on the sloped facet model) that

$$J(r^*,c^*) = \alpha r^* + \beta c^* + \gamma + \eta(r^*,c^*), \tag{3}$$

for some appropriate α, β and γ, where $\eta(r^*,c^*)$ is the error at pixel with local coordinates (r^*,c^*).

Mathematical development of FM

- A least-squares procedure is used to determine estimates for α, β and γ.

- Let

$$f(\alpha,\beta,\gamma) = \sum_{r^*=-L}^{L} \sum_{c^*=-L}^{L} (\alpha r^* + \beta c^* + \gamma - J(r^*, c^*))^2 . \qquad (4)$$

- The least-squares estimates for α, β and γ are those that minimize f.

Mathematical development of FM

- To determine these values, we take the partial derivatives of f with respect to α, β and γ:

$$\frac{\partial f}{\partial \alpha} = 2 \sum_{r^*=-L}^{L} \sum_{c^*=-L}^{L} (\alpha r^* + \beta c^* + \gamma - J(r^*, c^*)) \, r^*, \tag{5}$$

$$\frac{\partial f}{\partial \beta} = 2 \sum_{r^*=-L}^{L} \sum_{c^*=-L}^{L} (\alpha r^* + \beta c^* + \gamma - J(r^*, c^*)) \, c^*, \tag{6}$$

$$\frac{\partial f}{\partial \gamma} = 2 \sum_{r^*=-L}^{L} \sum_{c^*=-L}^{L} (\alpha r^* + \beta c^* + \gamma - J(r^*, c^*)). \tag{7}$$

Mathematical development of FM

- We set these to zero and then solve for α, β and γ:

$$\sum_{r^*=-L}^{L} \sum_{c^*=-L}^{L} \left(\alpha (r^*)^2 + \beta r c^* + \gamma r^* - J(r^*, c^*) r^* \right) = 0, \tag{8}$$

$$\sum_{r^*=-L}^{L} \sum_{c^*=-L}^{L} \left(\alpha r^* c^* + \beta (c^*)^2 + \gamma c^* - J(r^*, c^*) c^* \right) = 0, \tag{9}$$

$$\sum_{r^*=-L}^{L} \sum_{c^*=-L}^{L} (\alpha r^* + \beta c^* + \gamma - J(r^*, c^*)) = 0. \tag{10}$$

Mathematical development of FM

- Known facts:

$$\sum_{i=-K}^{K} i = 0, \tag{11}$$

.

$$\sum_{i=-K}^{K} i^2 = \frac{1}{3} K(K+1)(2K+1). \tag{12}$$

- Using (8)-(12) we obtain:

$$\frac{1}{3} L(L+1)(2L+1)^2 \alpha - \sum_{r^*=-L}^{L} r^* \sum_{c^*=-L}^{L} J(r^*, c^*) = 0, \tag{13}$$

$$\frac{1}{3} L(L+1)(2L+1)^2 \beta - \sum_{c^*=-L}^{L} c^* \sum_{r^*=-L}^{L} J(r^*, c^*) = 0, \tag{14}$$

$$(2L+1)^2 \gamma - \sum_{r^*=-L}^{L} \sum_{c^*=-L}^{L} J(r^*, c^*) = 0. \tag{15}$$

Mathematical development of FM

- Solving for α, β and γ:

$$\alpha = \frac{3}{L(L+1)(2L+1)^2} \sum_{r^*=-L}^{L} r^* \sum_{c^*=-L}^{L} J(r^*, c^*), \tag{16}$$

$$\beta = \frac{3}{L(L+1)(2L+1)^2} \sum_{c^*=-L}^{L} c^* \sum_{r^*=-L}^{L} J(r^*, c^*), \tag{17}$$

$$\gamma = \frac{1}{(2L+1)^2} \sum_{r^*=-L}^{L} \sum_{c^*=-L}^{L} J(r^*, c^*). \tag{18}$$

Example $K = 3$

- When $K = 3$, $L = 1$.

- Then

$$\alpha = \frac{1}{6}\left[J(1,\cdot) - J(-1,\cdot)\right], \tag{19}$$

$$\beta = \frac{1}{6}\left[J(\cdot,1) - J(\cdot,-1)\right], \tag{20}$$

$$\gamma = \frac{1}{9}J(\cdot,\cdot). \tag{21}$$

where an argument of J taking the value dot means that J is summed from $-L$ to L in that argument position.

- Interpretation:
  - α is proportional to the slope down the row dimension,
  - β is proportional to the slope across the column dimension, and
  - γ is the simple gray value average over the block.

Example $K = 3$ (continued)

- The fitted gray level for any pixel (r^*, c^*) in the block is given by

$$\hat{J}(r^*, c^*) = \alpha r^* + \beta c^* + \gamma. \tag{22}$$

- Then

$$\begin{aligned} \hat{J}(r^*, c^*) &= \frac{1}{6}\left[J(1,\cdot) - J(-1,\cdot)\right] r^* \\ &\quad + \frac{1}{6}\left[J(\cdot,1) - J(\cdot,-1)\right] c^* \\ &\quad + \frac{1}{9} J(\cdot,\cdot). \end{aligned} \tag{23}$$

Example $K = 3$ (continued)

- Writing this expression in full:

$$
\begin{aligned}
\hat{J}(r^*, c^*) = \frac{1}{18} \quad & \{J(-1,-1)(-3r^* - 3c^* + 2) \\
& +J(-1,0)(-3r^* + 2) \\
& +J(-1,1)(-3r^* + 3c^* + 2) \\
& +J(0,-1)(-3c^* + 2) \\
& +J(0,0)(2) \\
& +J(0,1)(3c^* + 2) \\
& +J(1,-1)(3r^* - 3c^* + 2) \\
& +J(1,0)(3r^* + 2) \\
& +J(1,1)(3r^* + 3c^* + 2)\} \qquad (24)
\end{aligned}
$$

Example $K = 3$ (continued)

- This leads to the set of linear filter masks shown below for fitting each pixel position in the 3 X 3 block (each block should be normalized by dividing by 18):

$\hat{J}(-1,-1)$

8	5	2
5	2	-1
2	-1	-4

$\hat{J}(-1,0)$

5	5	5
2	2	2
-1	-1	-1

$\hat{J}(-1,1)$

2	5	8
-1	2	5
-4	-1	2

$\hat{J}(0,-1)$

5	2	-1
5	2	-1
5	2	-1

$\hat{J}(0,0)$

2	2	2
2	2	2
2	2	2

$\hat{J}(0,1)$

-1	2	5
-1	2	5
-1	2	5

$\hat{J}(1,-1)$

2	-1	-4
5	2	-1
8	5	2

$\hat{J}(1,0)$

-1	-1	-1
2	2	2
5	5	5

$\hat{J}(1,1)$

-4	-1	2
-1	2	5
2	5	8

Block error calculation

- The FM iterative step examines, for each image pixel, each of the at most K^2 blocks to which that pixel belongs.
- For each block, the *error of fit* is

$$\sum_{r^*=-L}^{L} \sum_{c^*=-L}^{L} \left(\hat{J}(r^*, c^*) - J(r^*, c^*)\right)^2. \tag{25}$$

- There is a block with a minimal error of fit among those that contain the pixel whose global coordinates are (r, c).
- Let (r^*, c^*) be the coordinates of the pixel with global coordinates (r, c) in the local coordinate system of the block having smallest error of fit.
- The output gray value of the iterative step of FM at pixel with global coordinates (r, c) is then given by $\hat{J}(r^*, c^*)$, which is the fitted gray value at local coordinates (r^*, c^*) for the block having smallest error of fit, see (22).

Outline

Introduction
Facets and facet model

The facet model iterative procedure
Procedure overview
Mathematical development
Example: 3×3 neighborhoods
Block error calculation

Superiorization and the facet model
Convex feasibility problems
Superiorization using the facet model

Convex feasibility problems

- Let n be an arbitrary but fixed positive integer.
- Let $\mathfrak{C}$ be a nonempty collection of closed convex subsets of the vector space $\mathbb{R}^n$.
- A *convex feasibility problem* $T_{\mathfrak{C}}$ has the form:

$$\text{find an } \boldsymbol{x}^* \in \bigcap_{C \in \mathfrak{C}} C. \tag{26}$$

- Such points $\boldsymbol{x}^*$ are called *solutions* of $T_{\mathfrak{C}}$. They do not necessarily exist.

Convex feasibility problems

- An *algorithm* $\mathbf{P}$ for convex feasibility assigns to each $T_{\mathfrak{C}}$ an algorithmic operator $\mathbf{P}_{T_{\mathfrak{C}}} : \mathbb{R}^n \to \mathbb{R}^n$.
- Such an algorithm $\mathbf{P}$ is said to be *bounded perturbations resilient* if, for all $T_{\mathfrak{C}}$, the following is the case: if the sequence $\left(\left(\mathbf{P}_{T_{\mathfrak{C}}}\right)^k \boldsymbol{x}\right)_{k=0}^{\infty}$ converges to a solution of $T_{\mathfrak{C}}$ for all $\boldsymbol{x} \in \mathbb{R}^n$, then any sequence $\left(\boldsymbol{x}^k\right)_{k=0}^{\infty}$ of points in $\mathbb{R}^n$ also converges to a solution of $T_{\mathfrak{C}}$ provided that, for all $k \geq 0$,

$$\boldsymbol{x}^{k+1} = \mathbf{P}_{T_{\mathfrak{C}}}\left(\boldsymbol{x}^k + \beta_k \boldsymbol{v}^k\right), \tag{27}$$

  where $\beta_k \boldsymbol{v}^k$ are bounded perturbations, meaning that β_k are real nonnegative numbers such that $\sum_{k=0}^{\infty} \beta_k < \infty$ and the sequence $\left(\boldsymbol{v}^k\right)_{k=0}^{\infty}$ is bounded.
- There are bounded perturbations resilient algorithms for convex feasibility that have the property that, for any $\mathfrak{C}$ such that $\bigcap_{C \in \mathfrak{C}} C$ is nonempty, the sequence $\left(\left(\mathbf{P}_{T_{\mathfrak{C}}}\right)^k \boldsymbol{x}\right)_{k=0}^{\infty}$ converges to a solution of $T_{\mathfrak{C}}$ for all $\boldsymbol{x} \in \mathbb{R}^n$. We call such algorithms *good*.

Convex feasibility problems

- For any $\mathfrak{C}$ and any $\boldsymbol{x} \in \mathbb{R}^n$, we define

$$Pr_{\mathfrak{C}}(\boldsymbol{x}) = \sqrt{\sum_{C \in \mathfrak{C}} (d(\boldsymbol{x}, C))^2}, \tag{28}$$

  where $d(\boldsymbol{x}, C)$ is the Euclidean distance of $\boldsymbol{x}$ from the set C.
- Clearly, $\boldsymbol{x}$ is a solution of $T_{\mathfrak{C}}$ if, and only if, $Pr_{\mathfrak{C}}(\boldsymbol{x}) = 0$.
- There are good algorithms **P** for convex feasibility such that, for all $T_{\mathfrak{C}}$,
  - $\mathbf{P}_{T_{\mathfrak{C}}}$ is a continuous operator and
  - either $T_{\mathfrak{C}}$ does not have a solution or, for any $\boldsymbol{x} \in \mathbb{R}^n$ that is not a solution of $T_{\mathfrak{C}}$, $Pr_{\mathfrak{C}}\left(\mathbf{P}_{T_{\mathfrak{C}}} \boldsymbol{x}\right) < Pr_{\mathfrak{C}}(\boldsymbol{x})$.
- We call such algorithms *very good.*

Superiorization using the facet model

- Superiorization uses perturbations to steer the iterates towards a superior solution (rather than just any solution) of a convex feasibility problem.
- The superiority of the solution is measured by the smallness of an objective function $\phi : \mathbb{R}^n \to \mathbb{R}$.
- In order to apply this to images, we map the gray values of a $Z_r \times Z_c$ image into the components of a vector $x \in \mathbb{R}^n$, with $n = Z_r \times Z_c$. The measured data and prior information (such as the estimated line integrals and nonnegativity of the x-ray linear attenuation coefficients in image reconstruction from projections) give rise to the closed convex set of a convex feasibility problem.
- Let $\mathbf{H} : \mathbb{R}^n \to \mathbb{R}^n$ be the operator that corresponds to a single iterative step of FM.
- We define our objective function to be

$$\phi(\boldsymbol{x}) = \|\mathbf{H}\boldsymbol{x} - \boldsymbol{x}\|. \tag{29}$$

Superiorization using the facet model

- For any $\boldsymbol{x} \in \mathbb{R}^n$, we define

$$\boldsymbol{v}(\boldsymbol{x}) = \begin{cases} \min\left\{1, \frac{1}{\|\mathbf{H}\boldsymbol{x}-\boldsymbol{x}\|}\right\}(\mathbf{H}x - x), & \text{if } \phi(\boldsymbol{x}) \neq 0, \\ 0, & \text{otherwise.} \end{cases} \tag{30}$$

- Note that it is always the case that $\|\boldsymbol{v}(\boldsymbol{x})\| \leq 1$.

Conjecture 1

For all $\boldsymbol{x} \in \mathbb{R}^n$ and for $0 \leq \beta \leq 1$, then

$$\phi(\boldsymbol{x} + \beta \boldsymbol{v}(\boldsymbol{x})) \leq \phi(\boldsymbol{x}). \tag{31}$$

- In addition to the already described concepts, the Superiorization Algorithm makes use of an initial vector $\bar{\boldsymbol{x}}$, a real number parameter a, $0 < a < 1$, and a positive real number parameter ε. The output of the algorithm is denoted by $\boldsymbol{x}^*$. It also makes use of a very good algorithm $\mathbf{P}$ for convex feasibility.

Superiorization using the facet model

The *Superiorization Algorithm for the convex feasibility problem* $T_{\mathfrak{C}}$*:*

$k = 0$

$\beta = 1$

$\boldsymbol{x}^k = \bar{\boldsymbol{x}}$

WHILE $Pr_{\mathfrak{C}}(\boldsymbol{x}^k) > \varepsilon$ **DO**

$\boldsymbol{v}^k = \boldsymbol{v}(\boldsymbol{x}^k)$

logic=true

WHILE logic

$\boldsymbol{x}^{k+1} = \mathbf{P}_{T_{\mathfrak{C}}}\left(\boldsymbol{x}^k + \beta \boldsymbol{v}^k\right)$

IF $Pr_{\mathfrak{C}}\left(\boldsymbol{x}^{k+1}\right) < Pr_{\mathfrak{C}}\left(\boldsymbol{x}^k\right)$ **THEN** logic=false

$\beta = a \times \beta$

$k = k + 1$

$\boldsymbol{x}^* = \boldsymbol{x}^k$

Superiorization using the facet model

- If this algorithm terminates, then its output $\boldsymbol{x}^*$ is clearly such that $Pr_{\mathfrak{C}}(\boldsymbol{x}^*) \leq \varepsilon$.
- From the fact that **P** is a very good algorithm for convex feasibility, we see that the Superiorization Algorithm cannot get stuck in the **WHILE** logic loop.
- This together with the fact that **P** is a good algorithm for convex feasibility, implies that the Superiorization Algorithm will terminate in a finite number of steps for any $\mathfrak{C}$ such that $\bigcap_{C \in \mathfrak{C}} C$ is nonempty.
- That the Superiorization Algorithm steers the sequence of the $\boldsymbol{x}^k$ in a direction that tends to reduce the value of the objective function ϕ follows from Conjecture 1.
- Implementation and evaluation of performance on real convex feasibility problems is work in progress.