2008

Reconstructing Images as Piecewise Smooth Functions

Ralf Juengling
Portland State University, ralf-juengling@gmail.com

Reconstructing Images as Piecewise Smooth Functions

by Ralf Juengling

Department of Computer Science, Portland State University
PO Box 751 Portland, OR 97207 USA

Email: juenglin@cs.pdx.edu

Abstract

Leclerc's approach to image reconstruction consists of finding the shortest description of the data (an image) as a model (reconstruction) plus noise [5]. The approach poses two design problems: 1. Define an appropriate description language for image models and noise, 2. Derive an objective function and conceive an optimization algorithm that finds good local minima. Leclerc proposed to model images as piecewise low order polynomials and to describe models in terms of region boundaries (discontinuity set) and polynomial coefficients.

In this report I describe Leclerc's methodology, and, adopting his image model and description language, derive an objective function within this methodology. I discuss the differences of my result with Leclerc's objective function, sketch the optimization algorithm and give expressions for the gradient of my objective function.

1 Introduction

1.1 The Reconstruction Problem

Like Leclerc [5] I want to find an explicit description of an image as a piecewise smooth function plus noise. The image is given as a set of gray values $\{z_i\}$, $z_i \in \mathcal{Z}$, assigned to nodes in a regular rectangular grid. Let the nodes be identified with labels from some label set $\mathcal{I}$. We associate coordinates $\boldsymbol{x}_i = (x_i, y_i)$ and a square domain $\mathcal{D}_i$, centered on $\boldsymbol{x}_i$, with each node $i \in \mathcal{I}$. The pair $(z_i, \mathcal{D}_i)$ is commonly called *pixel* (short for *picture element*), in line with the custom of reproducing an image as the piecewise constant function

$$z\colon \mathcal{D} \to \mathcal{Z}, \;\; z(\boldsymbol{x})|_{\mathcal{D}_i} = z_i, \text{ with } \mathcal{D} = \bigcup_{i \in \mathcal{I}} \mathcal{D}_i.$$

However, our intuition is that there is a *real* image, which is the limit of z with the number of nodes increasing, $|\mathcal{I}| \to \infty$, while the domain of z is held constant. Neglecting diffraction and sensor noise of the imaging system [4], the limit would be a piecewise smooth function. The task here is to define a method for finding a good estimate u of the real image (the problem is sometimes called image *reconstruction*, or *restoration*; see [3, 1] for other prominent approaches). Image reconstruction has been used for denoising images or for finding contours in images as the set of discontinuities in u.

1.2 Reconstruction by Optimization

Leclerc posed the reconstruction problem as an optimization problem [5]. As the space for u he chose the space of piecewise low-order polynomial functions over the image domain. More precisely, for any such function $u\colon \mathcal{D} \to \mathbb{R}$ there exists a finite number of regions $\{R_j\}_{j \in J}$, with $R_i \cap R_j = \varnothing$ when $i \neq j$, $\cup_{j \in J} R_j = \mathcal{D}$, and each R_j is a connected set of the form $R_j = \cup_{i \in I_j} \mathcal{D}_i$, such that $u|_{R_j}$ is a polynomial of order at most two. (The fact that a region R_j is the union of pixel domains is not a desirable property of u, but a technical constraint to limit the space of possible solutions.)

We may describe u by specifying the regions $\{R_j\}$ over which u is smooth plus a tuple of coefficients per region. I will refer to this format as a "natural description" of u and will say *the* natural description to refer to the shortest natural description. The natural description of a possible solution has a variable number of components. This makes it hard to define an efficient optimization procedure and so we need an alternative description. Since u is piecewise polynomial and the regions are unions of pixel domains, we may completely describe any possible function u by tuples of coefficients $\{\boldsymbol{u}_i\}$, one tuple for each pixel domain. That is, with $\boldsymbol{u}_i = (u_{i,0}, u_{i,1}, ..., u_{i,5})$, u over the domain $\mathcal{D}_i$ is

$$u = u_{i,0} + u_{i,1}(x - x_i) + u_{i,2}(y - y_i) + \frac{u_{i,3}}{2}(x - x_i)^2 + u_{i,4}(x - x_i)(y - y_i) + \frac{u_{i,5}}{2}(y - y_i)^2. \tag{1}$$

When $u_{i,k}, k = 1...5$ are all zero we say u is zero-order over $\mathcal{D}_i$. When $u_{i,3}$, $u_{i,4}$, and $u_{i,5}$ are zero and $u_{i,1}$ or $u_{i,2}$ non-zero we say u is first-order, and otherwise second-order over $\mathcal{D}_i$.

The description of u based on Eq. (1) is a vector $\boldsymbol{u}$ of length $6 \times |\mathcal{I}|$ holding all coefficients for all pixels. It is, of course, possible to derive the natural description from $\boldsymbol{u}$ by identifying all discontinuities in u and finding the partitioning $\{R_j\}$ of shortest description length such that no discontinuity is interior to any region R_j. In other words, natural descriptions and descriptions in terms of coefficient vectors $\boldsymbol{u}$ span the same function space. We will use the coefficient vector description to define an objective function on $U = \mathbb{R}^{6\times|\mathcal{I}|}$, but this objective function measures properties of u which are more easily expressed in terms of the natural description.

2 Objective Function

It is clear that the reconstruction u should exhibit some similarity with z. This property will be ensured by a "fidelity term" $\sim \sum_{i\in\mathcal{I}} \left(z_i - u(\boldsymbol{x}_i)\right)^2$ in the objective function. However, U contains many functions that are reasonably similar to z and we need to come up with more desirable properties of the solution to make a selection. Leclerc's idea is that the reconstruction should be the *simplest*, in a sense, of the reasonably similar ones and that simplicity could be formalized as length of the natural description of u (using minimum length of description in model selection and for defining prior probabilities is an established idea in statistics [6]). To balance all desirable properties of u in a principled way, we define an objective function L that approximately measures the length of a description of the data (i.e., of $\{z_i\}_{i\in\mathcal{I}}$) in terms of u. From a high-level point of view, L has two components:

$$L = |\text{description of } u| + |\text{description of } \{z_i - u(\boldsymbol{x}_i)\}|. \tag{2}$$

Since we do not require $u(\boldsymbol{x}_i) = z_i$ for all $i \in \mathcal{I}$, the differences in gray value between z and u at the nodes need to be accounted for by some noise model. Following Leclerc we assume zero-mean Gaussian noise with unknown, piecewise constant variance. (If sensor noise were the only source of noise, a spatially uniform noise model would be appropriate. However, another phenomenon we want to account for is *texture*, which, in Horn's words, is "detailed structure in an image that is too fine to be resolved, yet coarse enough to produce a noticeable fluctuation in the gray-levels of neighboring picture cells" [4], page 140.) We further assume that the variance is constant over the regions of the natural description and hold all variance values in a vector $\boldsymbol{\sigma}$, one value σ_i for every pixel. Thus, the objective function is a function of $\boldsymbol{u}$ and $\boldsymbol{\sigma}$, $L(\boldsymbol{u}, \boldsymbol{\sigma})$.

As $z_i - u(\boldsymbol{x}_i)$ is a known stochastic process we may use information-theoretic arguments to compute the length of $|\text{description of } \{z_i - u(\boldsymbol{x}_i)\}|$ in an encoding that is optimal in the sense of minimizing the expected value of $|\text{description of } z_i - u(\boldsymbol{x}_i)|$. From this follows [5]

$$|\text{description of } \{z_i - u(\boldsymbol{x}_i)\}| = \frac{1}{\log 2} \sum_{i\in\mathcal{I}} \left[\frac{1}{2} \left(\frac{z_i - u_{i,0}}{\sigma_i} \right)^2 + \log \sigma_i \right], \tag{3}$$

giving justification to the particular form of the fidelity term.

We want the other term, |description of u|, to be an approximation of the length of the natural description of u. If we knew the regions $\{R_j\}_{j\in J}$ of the natural description, this term could be decomposed as

$$|\text{description of } u| = \sum_{j\in J} |\text{description of } R_j| + \left|\text{description of } u|_{R_j}\right|. \tag{4}$$

A succinct way of describing a region R_j is by encoding its boundary with a chain-code [2]. This works here because R_j is the union of pixel domains, thus, at any point there is always a small, finite number of possible continuations of a contour. In a chain-code encoding of a contour we specify the first and last element, and the direction of continuation for each element in between. Thus we get

$$|\text{description of } R_j| = |\text{description of } \partial R_j| = 2\,(l_s - l_e) + l_e\,|\partial R_j|\,, \tag{5}$$

where l_s is the description length for the first (last) element, l_e is the description length per element, and ∂R_j is R_j's boundary.

The length of the other term, $\left|\text{description of } u|_{R_j}\right|$, depends on whether $u|_{R_j}$ is zero-order, first-order, or second-order. If $u|_{R_j}$ is zero-order, we need to encode one coefficient and the variance value (see Eq. (1)). A first-order (second-order) $u|_{R_j}$ requires two (three) more coefficients. Thus, if l_c is the number of bits required to encode a number and n_j the number of extra coefficients required to describe $u|_{R_j}$,

$$\left|\text{description of } u|_{R_j}\right| = l_c(2 + n_j). \tag{6}$$

Next I discuss how to approximately compute terms (5) and (6) from the coefficient vector $\boldsymbol{u}$ and using only "local" computations (each computation depends on a few entries in $\boldsymbol{u}$ only, corresponding to pixels close to each other).

2.1 Discontinuities

Different regions are separated by discontinuities. It makes sense to distinguish different kinds of discontinuities, for two adjacent regions may share lower-order coefficients but have different high-order coefficients. To be precise about discontinuities we introduce notation for "the k^{th} derivative of u around $\boldsymbol{x}$". Taking the k^{th} derivative of u around $\boldsymbol{x}$ means taking derivatives w.r.t. x and y of orders corresponding to coefficient $u_{i,k}$ in Eq. (1). Derivatives 1–5 are:

$$\begin{aligned}
D_1(\boldsymbol{u}_i, \boldsymbol{x}) = \left.\frac{\partial u}{\partial x}\right|_{\mathcal{D}_i} &= u_{i,1} + u_{i,3}(x - x_i) + u_{i,4}(y - y_i) \\
D_2(\boldsymbol{u}_i, \boldsymbol{x}) = \left.\frac{\partial u}{\partial y}\right|_{\mathcal{D}_i} &= u_{i,2} + u_{i,4}(x - x_i) + u_{i,5}(y - y_i) \\
D_3(\boldsymbol{u}_i, \boldsymbol{x}) = \left.\frac{\partial^2 u}{\partial x^2}\right|_{\mathcal{D}_i} &= u_{i,3} \\
D_4(\boldsymbol{u}_i, \boldsymbol{x}) = \left.\frac{\partial^2 u}{\partial x \partial y}\right|_{\mathcal{D}_i} &= u_{i,4} \\
D_5(\boldsymbol{u}_i, \boldsymbol{x}) = \left.\frac{\partial^2 u}{\partial y^2}\right|_{\mathcal{D}_i} &= u_{i,5}
\end{aligned}$$

and, for completeness, $D_0(\boldsymbol{u}_i, \boldsymbol{x}) = u(\boldsymbol{x})|_{\mathcal{D}_i}$.

Second, we need to be precise about the neighborhood relationship of pixels. On this point I deviate from Leclerc and use a topology in which every interior pixel has six neighbor pixels and write N_i^6 for the neighbors of pixel i (see the Appendix for a short discussion of different conventions used for pixel topologies; the advantage of this topology over the 8-connected neighborhood used by Leclerc will become clear later).

We say there is a discontinuity between two neighboring pixels i and $j \in N_i^6$ when

$$D_k(\boldsymbol{u}_i, \overline{\boldsymbol{x}_{i,j}}) \neq D_k(\boldsymbol{u}_j, \overline{\boldsymbol{x}_{i,j}}) \tag{7}$$

for any $k = 0...5$, where $\overline{\boldsymbol{x}_{i,j}} = \frac{1}{2}(\boldsymbol{x}_i + \boldsymbol{x}_j)$.[1] A *zero-order discontinuity* occurs when inequality (7) holds for $k = 0$ only. A *first-order discontinuity* occurs when Eq. (7) holds for at least one k, $0 < k \leq 2$, but for no $k > 2$, a *second-order discontinuity* occurs when Eq. (7) holds for at least one k, $2 < k \leq 5$. The following notational vehicle for expressing the order of a discontinuity will come in handy below

$$o_{i,j} = o_{i,j}(\boldsymbol{u}) = \begin{cases} 3, & \text{if } j \in N_i \text{ and } D_k(\boldsymbol{u}_i, \overline{\boldsymbol{x}_{i,j}}) \neq D_k(\boldsymbol{u}_j, \overline{\boldsymbol{x}_{i,j}}) \text{ for } k = 3,\ k = 4 \text{ or } k = 5 \\ 2, & \text{if } j \in N_i \text{ and } D_k(\boldsymbol{u}_i, \overline{\boldsymbol{x}_{i,j}}) \neq D_k(\boldsymbol{u}_j, \overline{\boldsymbol{x}_{i,j}}) \text{ for } k = 1 \text{ or } k = 2 \\ 1, & \text{if } j \in N_i \text{ and } D_0(\boldsymbol{u}_i, \overline{\boldsymbol{x}_{i,j}}) \neq D_0(\boldsymbol{u}_j, \overline{\boldsymbol{x}_{i,j}}) \\ 0, & \text{if } j \notin N_i \text{ or } D_k(\boldsymbol{u}_i, \overline{\boldsymbol{x}_{i,j}}) = D_k(\boldsymbol{u}_j, \overline{\boldsymbol{x}_{i,j}}) \text{ for all } k \end{cases}$$

2.2 Description Length of $\boldsymbol{u}$

We now assemble an expression that plays the role of term (5) in (4). With the new definitions from the preceding section we require that a contour is homogeneous in order (that is, there are three types of contours). It is easy to simply count all discontinuities,

$$\begin{aligned} \sum_{j \in J} |\partial R_j| &= \frac{l_e}{2} \sum_{i \in \mathcal{I}} \sum_{j \in N_i^6} \max_k \left(1 - \delta(D_k(\boldsymbol{u}_i, \overline{\boldsymbol{x}_{i,j}}) - D_k(\boldsymbol{u}_j, \overline{\boldsymbol{x}_{i,j}})\right) \\ &= \frac{l_e}{2} \sum_{i \in \mathcal{I}} \sum_{j \in N_i^6} \left(1 - \prod_{k=0,\ldots,5} \delta(\Delta_{i,j,k}(\boldsymbol{u}))\right) \end{aligned} \tag{8}$$

where $\delta(x) = 0$ for $x \neq 0$, $\delta(0) = 1$, $\Delta_{i,j,k}(\boldsymbol{u}) = D_k(\boldsymbol{u}_i, \overline{\boldsymbol{x}_{i,j}}) - D_k(\boldsymbol{u}_j, \overline{\boldsymbol{x}_{i,j}})$ and the factor $\frac{1}{2}$ accounts for counting all discontinuity elements twice on the right. Since a partial contour may only continue in one of two directions (see Appendix), the description length per contour element is 1 bit, $l_e = 1$.

To find ends of contours we need to look at triplets of discontinuity elements that correspond to pixel *cliques* in the neighborhood graph (see Appendix). A pixel clique is a set $\{h, i, j\} \subset \mathcal{I}$, such that $h, i \in N_j^6$, $i, j \in N_h^6$, and $h, j \in N_i^6$. At each such triplet we may observe up to three contour terminations. For instance, let $\{h, i, j\}$ be a pixel clique; if $(o_{i,j}, o_{h,i}, o_{j,h}) = (0, 1, 0)$, we count one contour termination; if $(o_{i,j}, o_{h,i}, o_{j,h}) = (0, 1, 1)$, we count zero terminations; if $(o_{i,j}, o_{h,i}, o_{j,h}) = (2, 0, 1)$ there are two contour terminations, and so on. There are $4^3 = 64$ combinations to consider and we denote t the function that maps any combination to the termination count, $t: T \times T \times T \to T$, where $T = \{0, 1, 2, 3\}$.

With the definition of a "clique indicator" $c_{h,i,j}$,

$$c_{h,i,j} = \begin{cases} 1, & \text{if } \{h, i, j\} \subset \mathcal{I} \text{ is a clique} \\ 0, & \text{otherwise} \end{cases}$$

1. Note that there are only six possible values $\boldsymbol{x}_j - \boldsymbol{x}_i$, and $D_k(\boldsymbol{u}_i, \overline{\boldsymbol{x}_{i,j}})$ may be written as a dot product of the coefficients $\boldsymbol{u}_i$ with a pre-computed vector $\boldsymbol{d}_{i,k,n}$, $k = 0...5$, $n = 0...5$. For instance, with $\boldsymbol{x}_j - \boldsymbol{x}_i = (-1, 1)$ we may write $D_2(\boldsymbol{u}_i, \overline{\boldsymbol{x}_{i,j}})$ as

$$D_2(\boldsymbol{u}_i, \overline{\boldsymbol{x}_{i,j}}) = u_{i,2} + u_{i,4}\left(\frac{x_j - x_i}{2}\right) + u_{i,5}\left(\frac{y_j - y_i}{2}\right) = \left(0, 0, 1, 0, -\frac{1}{2}, \frac{1}{2}\right) \cdot \boldsymbol{u}_i = \boldsymbol{d}_{i,2,3} \cdot \boldsymbol{u}_i,$$

where $n = 3$ in some arbitrary but fixed enumeration of the neighbors of i.

we may finally write an expression for the other part of term (5) in (4),

$$2\,(l_s - l_e)\,|J| \approx \frac{2\,(l_s - l_e)}{3!} \sum_{h,i,j \in \mathcal{I}} c_{h,i,j}\, t(o_{h,i}, o_{i,j}, o_{h,j}), \tag{9}$$

which completes the first part of (4) (the factor 1/3! accounts for the number of possible permutations of the three pixels in a clique).

To derive an approximation of (6) we make use of the fact that regions are unions of pixel domains, $R_j = \cup_{i \in \mathcal{I}_j} \mathcal{D}_i$. Any pixel $i \in \mathcal{I}_j$ may be used to compute the value $l_c(2 + n_j)$ because $u|_{\mathcal{D}_i}$ is of the same order across R_j,

$$l_c(2+n_j) = l_c \left[2 + 2 \left(1 - \prod_{k=1,\ldots,5} \delta(u_{i,k}) \right) + 3 \left(1 - \prod_{k=3,4,5} \delta(u_{i,k}) \right) \right]. \tag{10}$$

However, we do not know the regions nor their size. Using local computations we may only determine whether pixel i is interior to a region or not (it is interior when $o_{i,j} = 0$ for all $j \in N_i$). The idea now is to simply sum (10) over all pixels $i \in \mathcal{I}$ and to correct this sum for every pair of adjacent pixels that shares coefficients.

$$\begin{aligned} \sum_{j \in J} \left| \text{descr.}\, u|_{R_j} \right| \;\approx\; & l_c \sum_{i \in \mathcal{I}} \left[2 + 2 \left(1 - \prod_{k=1,\ldots,5} \delta(u_{i,k}) \right) + 3 \left(1 - \prod_{k=3,4,5} \delta(u_{i,k}) \right) \right] - \\ & \frac{l_c}{6} \sum_{i \in \mathcal{I}} \sum_{j \in N_i^6} \left(2\, \delta(\Delta_{i,j,0}) + 2 \prod_{k=0,1,2} \delta(\Delta_{i,j,k}) + 3 \prod_{k=0,\ldots,5} \delta(\Delta_{i,j,k}) \right) \end{aligned} \tag{11}$$

The second sum in (11) removes excess contributions in the first sum. For example, consider two neighboring pixels i, j with $u|_{\mathcal{D}_i}$ and $u|_{\mathcal{D}_j}$ second order and a second-order discontinuity between i and j. In this case pixel i contributes $7\,l_c$ to the first sum and the pair (i, j) contributes $\frac{4}{6} l_c$ to the second sum, which amounts to a sixth of the description length for the second-order coefficients that i and j share.

In general, an interior pixel i has six neighbors N_i^6. If $u|_{\mathcal{D}_i}$ is C^2-continuous with all neighbors, then all contributions by i to the second sum in (11) will annihilate the contribution of i to the first sum. The net contributions in (11) are therefore by pixels that are not interior to a region. Thus, what we are actually computing in (11) is, approximately,

$$\sum_{j \in J} \left| \text{description of}\, u|_{R_j} \right| \times |\partial R_j|. \tag{12}$$

While (12) is not exactly what we set out to compute, note that this expression does not run counter to our overall objective as it combines two desirable properties of u that we want to optimize (compare with (5)).

We need to come up with one more term that ensures the piecewise uniform noise property of the solution. To be more precise, we require that $\sigma_i = \sigma_j$ when $j \in N_i^6$ and $o_{i,j} = 0$. This constraint may be enforced by adding the term

$$\frac{g}{2} \sum_{i \in \mathcal{I}} \sum_{j \in N_i^6} \delta(\Delta_{i,j,0})\,(1 - \delta(\sigma_i - \sigma_j)), \tag{13}$$

with $g \gg l_c$. Note that the interpretation of (13) is different from all other terms, it is not measuring length of any description.

Combining all contributions to the objective function we finally arrive at

$$
\begin{aligned}
L(\boldsymbol{u},\boldsymbol{\sigma}) \;=\; \sum_{i\in\mathcal{I}} \Bigg\{ & \frac{1}{\log 2}\left[\frac{1}{2}\left(\frac{z_i-u_{i,0}}{\sigma_i}\right)^2+\log\sigma_i\right]+\frac{g}{2}\sum_{j\in N_i^6}\delta(\Delta_{i,j,0})(1-\delta(\sigma_i-\sigma_j)) \\
& +\frac{l_e}{2}\sum_{j\in N_i^6}\left(1-\prod_{k=0,\dots,5}\delta(\Delta_{i,j,k})\right)+\frac{2(l_s-l_e)}{3!}\sum_{h,j\in N_i^6} c_{h,i,j}\,t(o_{h,i},o_{i,j},o_{h,j}) \\
& +l_c\left[2+2\left(1-\prod_{k=1,\dots,5}\delta(u_{i,k})\right)+3\left(1-\prod_{k=3,4,5}\delta(u_{i,k})\right)\right] \\
& -\frac{l_c}{6}\sum_{j\in N_i^6}\left(2\;\delta(\Delta_{i,j,0})+2\prod_{k=0,1,2}\delta(\Delta_{i,j,k})+3\prod_{k=0,\dots,5}\delta(\Delta_{i,j,k})\right)\Bigg\} \qquad (14)
\end{aligned}
$$

3 Optimization

The objective function (14) is not only not convex over $U\times(\mathbb{R}^+)^{|\mathcal{I}|}$ but has many local minima and is even discontinuous. For instance, if $u|_{\mathcal{D}_i}$ is first-order with $u_{i,1}$ the largest non-zero coefficient, then $\lim_{u_{i,1}\to 0}\; L(\boldsymbol{u},\boldsymbol{\sigma})$ does not exist as L is discontinuous at $\boldsymbol{u}_i=(u_{i,0},\,0,\,0,\,0,\,0,\,0)$. Clearly, optimization methods utilizing derivatives of L are not applicable without further provisions. Leclerc's solution is to use a *continuation method* [5].

3.1 Continuation Method

The solution is to embed L in a family of objective functions, $L(\boldsymbol{u},\boldsymbol{\sigma},s)$, $s\in\mathbb{R}^+$, such that $L(\boldsymbol{u},\boldsymbol{\sigma},0)$ equals $L(\boldsymbol{u},\boldsymbol{\sigma})$, $L(\boldsymbol{u},\boldsymbol{\sigma},s)$, $s>0$ is smooth, and there is an s^0 where $L(\boldsymbol{u},\boldsymbol{\sigma},s^0)$ has a unique, known minimum $(\boldsymbol{u}^{*\,0},\boldsymbol{\sigma}^{*0})$. S is called the *scale* parameter. The idea is that $L(\boldsymbol{u},\boldsymbol{\sigma},s)$, $s>0$ is also smooth with respect to s, which makes it possible to *track* a local minimum across scales with the following algorithm: Starting with $t=0$, repeat the steps

1. Update s: $s^{t+1}=\rho\,s^t$,
2. Use $(\boldsymbol{u}^{*\,t-1},\boldsymbol{\sigma}^{*\,t-1})$ as starting point of a search for a minimum of $L(\boldsymbol{u},\boldsymbol{\sigma},s^t)$

until s^t is sufficiently close to zero (and thereby $L(\boldsymbol{u},\boldsymbol{\sigma},s^t)$ sufficiently similar to $L(\boldsymbol{u},\boldsymbol{\sigma})$). The parameter $\rho\in(0,\,1)$ determines the step size between different instances of $L(\boldsymbol{u},\boldsymbol{\sigma},s)$. Leclerc recommends $\rho=0.95$.

3.2 Engineering the Embedding

We need to find an embedding $L(\boldsymbol{u},\boldsymbol{\sigma},s)$ with the required properties. I use Leclerc's embedding recipe and replace all Kronecker deltas in the objective function by exponentials of the form $\exp\big(-x^2/(s\,\sigma)^2\big)$. This function assumes the properties of $\delta(x)$ for $s\to 0$. A parameter σ is included to adjust for different scales of x at different places. For instance, if x is the difference $u(\boldsymbol{x}_i)-u(\boldsymbol{x}_j)$, $j\in N_i$, then $\sigma=\overline{\sigma_{i,j}}=\frac{1}{2}\,(\sigma_i+\sigma_j)$ provides an appropriate scale.

Examining the objective function (14), however, we find that not all discontinuities may be traced back to a Kronecker delta. The exception is the expression inherited from (9), which counts contour terminations. To enable the continuation method I substitute a function $\bar{t}$ for t, $\bar{t}:\big[0,\,3\big]^3\to[0,\,3]$, which interpolates t and has easy to evaluate first derivatives (polynomial or trigonometric interpolation). Second, I rewrite $o_{i,j}(\boldsymbol{u})$ in terms of $\Delta_{i,j,k}$ as

$$
o_{i,j}(\boldsymbol{u})=6-\delta(\Delta_{i,j,0})-2\prod_{k=1,2}\delta(\Delta_{i,j,k})-3\prod_{k=3,4,5}\delta(\Delta_{i,j,k}).
$$

Now the embedding is obtained by substituting the exponentials

$$\begin{aligned} e^D_{i,j,k} = e^D_{i,j,k}(\boldsymbol{u}, s) &= \exp\left[-\left(\frac{2\,\Delta_{i,j,k}(\boldsymbol{u})}{s\left(\sigma_i^{*t-1} + \sigma_i^{*t-1}\right)} \right)^2 \right], \\ e^u_{i,k} = e^u_{i,k}(\boldsymbol{u}, s) &= \exp\left[-\left(\frac{u_{i,k}}{s\,\sigma_i^{*t-1}} \right)^2 \right], \text{and} \\ e^\sigma_{i,j} = e^\sigma_{i,j}(\boldsymbol{\sigma}, s) &= \exp\left[-\left(\frac{2\,(\sigma_i - \sigma_j)}{s\left(\sigma_i^{*t-1} + \sigma_i^{*t-1}\right)} \right)^2 \right] \end{aligned}$$

for $\delta(\Delta_{i,j,k})$, $\delta(u_{i,k})$, and $\delta(\sigma_i - \sigma_j)$, respectively:

$$\begin{aligned} L(\boldsymbol{u}, \boldsymbol{\sigma}, s) &= \sum_{i \in \mathcal{I}} \left\{ \frac{1}{\log 2} \left[\frac{1}{2} \left(\frac{z_i - u_{i,0}}{\sigma_i} \right)^2 + \log \sigma_i \right] + \frac{g}{2} \sum_{j \in N_i^6} e^D_{i,j,0} \times (1 - e^\sigma_{i,j}) \right. \\ &\quad + \frac{l_e}{2} \sum_{j \in N_i^6} \left(1 - \prod_{k=0,\ldots,5} e^D_{i,j,k} \right) + \frac{2(l_s - l_e)}{3!} \sum_{h,j \in N_i^6} c_{h,i,j}\, \bar{t}\,(o_{h,i}, o_{i,j}, o_{h,j}) \\ &\quad + l_c \left[2 + 2 \left(1 - \prod_{k=1,\ldots,5} e^u_{i,k} \right) + 3 \left(1 - \prod_{k=3,4,5} e^u_{i,k} \right) \right] \\ &\quad \left. - \frac{l_c}{6} \sum_{j \in N_i^6} \left(2\; e^D_{i,j,0} + 2 \prod_{k=0,1,2} e^D_{i,j,k} + 3 \prod_{k=0,\ldots,5} e^D_{i,j,k} \right) \right\}. \end{aligned} \tag{15}$$

Note that products of exponentials may be simplified. For example,

$$\prod_{k=1,\ldots,5} e^u_{i,k}(\boldsymbol{u}, \boldsymbol{\sigma}, s) = \exp\left[-\sum_{k=1,\ldots,5} \left(\frac{u_{i,k}}{s\,\sigma_i^{*t-1}} \right)^2 \right].$$

3.3 Minimization

According to Leclerc, simultaneous minimization of $L(\boldsymbol{u}, \boldsymbol{\sigma}, s^t)$ with respect to $(\boldsymbol{u}, \boldsymbol{\sigma})$ by iterative descent is not feasible. Instead he alternately minimizes with respect to $\boldsymbol{u}$ and $\boldsymbol{\sigma}$, respectively, holding $\boldsymbol{\sigma}$ fixed to the most recent values while minimizing $\boldsymbol{u}$, and vice versa.

The gradient of L with respect to $\boldsymbol{u}$ and $\boldsymbol{\sigma}$ is given in the Appendix. Setting the gradient $\partial L / \partial \boldsymbol{u}$ to zero results in a non-linear system of the form

$$\boldsymbol{A}(\boldsymbol{u}, \boldsymbol{\sigma}, s)\boldsymbol{u} + \boldsymbol{b}(\boldsymbol{u}, \boldsymbol{\sigma}, s) = 0.$$

The dependency on $\boldsymbol{u}$ aside, $\boldsymbol{A}$ has the properties of a stiffness matrix in FEM calculations (symmetric and sparse). Leclerc suggests to find a minimizer by linearizing the equations (i.e., fixing $\boldsymbol{u}$ in $\boldsymbol{A}(\boldsymbol{u}, \boldsymbol{\sigma}, s)$ and $\boldsymbol{b}(\boldsymbol{u}, \boldsymbol{\sigma}, s)$) and to solve for $\boldsymbol{u}$ by Gauss-Seidel iteration.

The gradient of L with respect to $\boldsymbol{\sigma}$ is not linearizable, so the minimization procedure for $\boldsymbol{u}$ is not applicable for $\boldsymbol{\sigma}$. Leclerc suggest the following modification to L to enable the linearization ([5], pages 90-91):

1. Replace the term $\sim \left(\frac{z_i - u_{i,0}}{\sigma_i} \right)^2$ in L by $\sim \left(\frac{z_i - u_{i,0}}{\sigma_i^{*t-1}} \right)^2$
2. Replace the term $\sim \log \sigma_i$ in L by a term $\sim \left(\frac{\sigma_i - \hat{\sigma_i}^{t-1}}{\sigma_i^{*t-1}} \right)^2$, where $\hat{\sigma_i}^{t-1}$ is an estimate of σ_i computed as $\hat{\sigma_i}^{t-1} = \sqrt{\frac{\sum_{j \in N_i} e^D_{i,j,0}(\boldsymbol{u}^{*t-1}, s^t)\,(z_j - D_0(\boldsymbol{u}^{*t-1}, \boldsymbol{x}_j))^2}{\sum_{j \in N_i} e^D_{i,j,0}(\boldsymbol{u}^{*t-1}, s^t)}}$

3.4 Leclerc's Objective Function

Leclerc derives the objective function (16) below ([5], page 91). The first two terms in the sum are the same as in (14) and are included for the same reasons (save for the term $\sum_{j\in S_i} K_{i,j}u_{j,0}$ in (16), which models the effect of a point-spread-function and which I left out).

$$\begin{aligned} L(\boldsymbol{u},\boldsymbol{\sigma}) \;=\; \sum_{i\in\mathcal{I}} \Bigg\{ & \frac{1}{\log 2}\left[\frac{1}{2}\left(\frac{z_i-\sum_{j\in S_i}K_{i,j}u_{j,0}}{\sigma_i}\right)^2+\log\sigma_i\right]+\frac{g}{2}\sum_{j\in N_i^8}\delta(\Delta_{i,j,0})(1-\delta(\sigma_i-\sigma_j)) \\ & +\frac{b}{2}\sum_{j\in N_i^8}\left[1-\delta(\Delta_{i,j,0})+2\left(1-\prod_{k=0,1,2}\delta(\Delta_{i,j,k})\right)+3\left(1-\prod_{k=0,\ldots,5}\delta(\Delta_{i,j,k})\right)\right] \\ & +d\left[2\left(1-\prod_{k=1,\ldots,5}\delta(u_{i,k})\right)+3\left(1-\prod_{k=3,4,5}\delta(u_{i,k})\right)\right]\Bigg\} \end{aligned} \tag{16}$$

The purpose of the term $\sim b$ is to approximate (5) ($|\text{description of } R_j|$). Note that this term is not really proportional to the number of discontinuities in u, but assigns a higher cost to higher order discontinuities. Further, Leclerc must assume a mean region boundary length to include the encoding cost for contour terminations. In contrast, we count contour terminations explicitly in (14). Similarly, the term $\sim d$ is to approximate (6) ($\left|\text{description of } u|_{R_j}\right|$). Again, one must assume an average region size to justify this term ($d \sim l_c/(\text{average region size})$). In contrast, (14) includes a contribution approximating $\left|\text{description of } u|_{R_j}\right| \times |\partial R_j|$ and is not tuned to a particular region size.

Another major difference is the use of the N_i^6 neighborhood in (14) versus the use of the N_i^8 neighborhood (16). With N_i^8 pixel cliques are of size 4, and a function corresponding to t above would be defined on T^4. Another advantage of N_i^6 is that a counterpart of the Jordan curve theorem holds in a topology generated with this neighborhood, and paradoxical answers to questions of connectedness do not occur [4].

While our objective function (14) is truer to the intended objective function (2), it also is more complex than (16). Thus, the design and implementation of a reconstruction procedure based on (14) will be more time consuming and its minimization computationally more expensive. It remains to be seen whether this increase in theoretical accuracy translates into more accurate image reconstructions.

4 Appendix

4.1 Pixel Neighborhoods

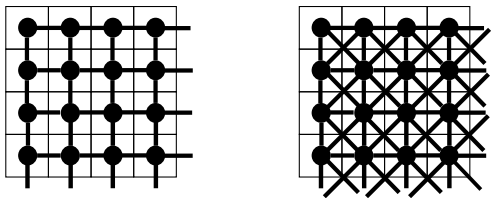

Figure 1. Pixel topology with 4- and 8-connected neighborhood, respectively.

Figure 1 shows the classical pixel topologies with 4- and 8-connected neighborhoods. Defining a topology like in Figure 2 (left) on a pixel-grid has some advantages over the more widely used 4- or 8-connected neighborhood [4]. Figure 2 (right) shows the possible elements of discontinuity that result with this topology and the places where we look for ends of contour chains, namely, where three discontinuity elements meet (shown as dots on the right). Note that the dots on the right correspond to cliques in the graph on the left.

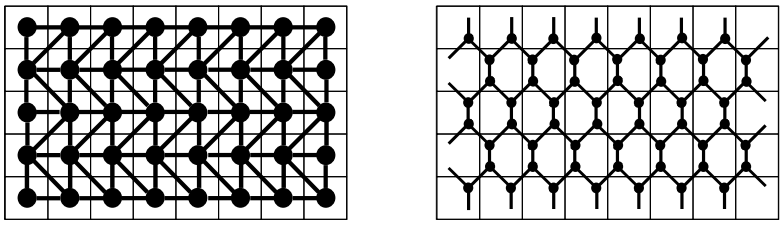

Figure 2. Left: Pixel topology (left) and corresponding discontinuity elements (right).

I am making use of different kinds of neighborhoods. To distinguish I write N_i^4 for all neighbors of pixel i in the 4-connected neighborhood, N_i^8 for all neighbors of i in the 8-connected neighborhood, and N_i^6 for all neighbors of i with the topology illustrated in Figure 2 (left). Note that $N_i^4 \subset N_i^6 \subset N_i^8$. In statements or definitions that make sense for any neighborhood, I sometimes write N_i.

4.2 Derivatives

$$\begin{aligned}\frac{\partial e^u_{i,k}}{\partial u_{i,l}} &= -\frac{2}{(s\,\sigma_i)^2} u_{i,l}\,\delta_{i,l}\,e^u_{i,k}\\ \frac{\partial e^\sigma_{i,j}}{\partial \sigma_i} &= -2\frac{\sigma_i-\sigma_j}{s^2\left(\sigma_i^{*t-1}+\sigma_j^{*t-1}\right)^2}\,e^\sigma_{i,j}\end{aligned}$$

	$e^D_{i,j,0}$	$e^D_{i,j,1}$	$e^D_{i,j,2}$	$e^D_{i,j,3}$	$e^D_{i,j,4}$	$e^D_{i,j,5}$
$\frac{1}{e^D_{i,j,k}}\frac{\partial}{\partial u_{i,0}}$	$2\frac{\Delta_{i,j,0}}{(s\,\overline{\sigma_{i,j}})^2}$	0	0	0	0	0
$\frac{1}{e^D_{i,j,k}}\frac{\partial}{\partial u_{i,1}}$	$2\frac{\Delta_{i,j,0}}{(s\,\overline{\sigma_{i,j}})^2}(x_j-x_i)$	$2\frac{\Delta_{i,j,1}}{(s\,\overline{\sigma_{i,j}})^2}$	0	0	0	0
$\frac{1}{e^D_{i,j,k}}\frac{\partial}{\partial u_{i,2}}$	$2\frac{\Delta_{i,j,0}}{(s\,\overline{\sigma_{i,j}})^2}(y_j-y_i)$	0	$2\frac{\Delta_{i,j,2}}{(s\,\overline{\sigma_{i,j}})^2}$	0	0	0
$\frac{1}{e^D_{i,j,k}}\frac{\partial}{\partial u_{i,3}}$	$\frac{\Delta_{i,j,0}}{(s\,\overline{\sigma_{i,j}})^2}(x_j-x_i)^2$	$2\frac{\Delta_{i,j,1}}{(s\,\overline{\sigma_{i,j}})^2}(x_j-x_i)$	0	$2\frac{\Delta_{i,j,3}}{(s\,\overline{\sigma_{i,j}})^2}$	0	0
$\frac{1}{e^D_{i,j,k}}\frac{\partial}{\partial u_{i,4}}$	$2\frac{\Delta_{i,j,0}}{(s\,\overline{\sigma_{i,j}})^2}(x_j-x_i)(y_j-y_i)$	$2\frac{\Delta_{i,j,1}}{(s\,\overline{\sigma_{i,j}})^2}(y_j-y_i)$	$2\frac{\Delta_{i,j,2}}{(s\,\overline{\sigma_{i,j}})^2}(x_j-x_i)$	0	$2\frac{\Delta_{i,j,4}}{(s\,\overline{\sigma_{i,j}})^2}$	0
$\frac{1}{e^D_{i,j,k}}\frac{\partial}{\partial u_{i,5}}$	$\frac{\Delta_{i,j,0}}{(s\,\overline{\sigma_{i,j}})^2}(y_j-y_i)^2$	0	$2\frac{\Delta_{i,j,2}}{(s\,\overline{\sigma_{i,j}})^2}(y_j-y_i)$	0	0	$2\frac{\Delta_{i,j,5}}{(s\,\overline{\sigma_{i,j}})^2}$

Table 1. Derivatives of $e^D_{i,j,k}$ with respect to $u_{i,l}$.

For readability I write $\epsilon^D_{i,j,k,l}$ for $\frac{\partial e^D_{i,j,k}}{\partial u_{i,l}}$, $\tilde{\epsilon}^D_{i,j,k,l}$ for $\frac{\partial e^D_{i,j,k}}{\partial u_{j,l}}$ and $\epsilon^u_{i,k,l}$ for $\frac{\partial e^u_{i,k}}{\partial u_{i,l}}$. Note that $\epsilon^D_{i,j,k,l}$ and $\epsilon^u_{i,k,l}$ are of the form $f(\boldsymbol{u})\,u_{i,l}$.

$$\begin{aligned}\frac{\partial o_{i,j}}{\partial u_{i,l}} &= -\epsilon^D_{i,j,0,l} - 2\left[\epsilon^D_{i,j,1,l}\times e^D_{i,j,2} + e^D_{i,j,1}\times\epsilon^D_{i,j,2,l}\right] - 3\left(\prod_{l=3,4,5} e^D_{i,j,l}\right)\sum_{k=3,4,5}\frac{\epsilon^D_{i,j,2,l}}{e^D_{i,j,k}}\cdot\\ \frac{\partial o_{i,j}}{\partial u_{j,l}} &= -\tilde{\epsilon}^D_{i,j,0,l} - 2\left[\tilde{\epsilon}^D_{i,j,1,l}\times e^D_{i,j,2} + e^D_{i,j,1}\times\tilde{\epsilon}^D_{i,j,2,l}\right] - 3\left(\prod_{l=3,4,5} e^D_{i,j,l}\right)\sum_{k=3,4,5}\frac{\tilde{\epsilon}^D_{i,j,2,l}}{e^D_{i,j,k}}\cdot\end{aligned}$$

$$
\begin{aligned}
\frac{\partial L}{\partial u_{i,0}} &= -\frac{1}{\log 2}\frac{z_i - u_{i,0}}{\sigma_i^2} + g\sum_{j\in N_i^6}\epsilon^D_{i,j,0,0}\times(1-e^\sigma_{i,j}) \\
&\quad - l_e\sum_{j\in N_i^6}\left(\prod_{k=0,\ldots,5} e^D_{i,j,k}\right)\frac{\epsilon^D_{i,j,0,0}}{e^D_{i,j,0}} \\
&\quad - 2\,(l_s - l_e)\sum_{h,j\in N_i^6} c_{h,i,j}\left(\frac{\partial \bar{t}\,(o_{h,i},o_{i,j},o_{h,j})}{\partial o_{h,i}}\frac{\partial o_{h,i}}{\partial u_{i,0}} + \frac{\partial \bar{t}\,(o_{h,i},o_{i,j},o_{h,j})}{\partial o_{i,j}}\frac{\partial o_{i,j}}{\partial u_{i,0}}\right) \\
&\quad + \frac{1}{3}\,l_c\sum_{j\in N_i^6}\frac{\epsilon^D_{i,j,0,0}}{e^D_{i,j,0}}\left(2\,e^D_{i,j,0} + 2\prod_{k=0,1,2} e^D_{i,j,k} + 3\prod_{k=0,\ldots,5} e^D_{i,j,k}\right) \\
\left.\frac{\partial L}{\partial u_{i,l}}\right|_{l=1,2} &= g\sum_{j\in N_i^6}\epsilon^D_{i,j,0,l}\times(1-e^\sigma_{i,j}) \\
&\quad - l_e\sum_{j\in N_i^6}\left(\prod_{k=0,\ldots,5} e^D_{i,j,k}\right)\sum_{k=0,1,2}\frac{\epsilon^D_{i,j,k,l}}{e^D_{i,j,k}} \\
&\quad - 2\,(l_s - l_e)\sum_{h,j\in N_i^6} c_{h,i,j}\left(\frac{\partial \bar{t}}{\partial o_{h,i}}\frac{\partial o_{h,i}}{\partial u_{i,l}} + \frac{\partial \bar{t}}{\partial o_{i,j}}\frac{\partial o_{i,j}}{\partial u_{i,l}}\right) \\
&\quad - 2\,l_c\frac{\epsilon^u_{i,l,l}}{e^u_{i,l}}\prod_{k=1,\ldots,5} e^u_{i,k} \\
&\quad + \frac{1}{3}\,l_c\sum_{j\in N_i^6}\frac{\epsilon^D_{i,j,0,l}}{e^D_{i,j,0}}\left(2\,e^D_{i,j,0} + 2\prod_{k=0,1,2} e^D_{i,j,k} + 3\prod_{k=0,\ldots,5} e^D_{i,j,k}\right) \\
&\quad + \frac{1}{3}\,l_c\sum_{j\in N_i^6}\frac{\epsilon^D_{i,j,l,l}}{e^D_{i,j,l}}\left(2\prod_{k=0,1,2} e^D_{i,j,k} + 3\prod_{k=0,\ldots,5} e^D_{i,j,k}\right) \\
\left.\frac{\partial L}{\partial u_{i,l}}\right|_{l=3,4,5} &= g\sum_{j\in N_i^6}\epsilon^D_{i,j,0,l}\times(1-e^\sigma_{i,j}) \\
&\quad - l_e\sum_{j\in N_i^6}\left\{\left(\prod_{k=0,\ldots,5} e^D_{i,j,k}\right)\sum_{k=0,1,2}\frac{\epsilon^D_{i,j,k,l}}{e^D_{i,j,k}} + \left(\prod_{k\neq l} e^D_{i,j,k}\right)\epsilon^D_{i,j,l,l}\right\} \\
&\quad - 2\,(l_s - l_e)\sum_{h,j\in N_i^6} c_{h,i,j}\left(\frac{\partial \bar{t}}{\partial o_{h,i}}\frac{\partial o_{h,i}}{\partial u_{i,l}} + \frac{\partial \bar{t}}{\partial o_{i,j}}\frac{\partial o_{i,j}}{\partial u_{i,l}}\right) \\
&\quad - 2\,l_c\frac{\epsilon^u_{i,l,l}}{e^u_{i,l}}\prod_{k=1,\ldots,5} e^u_{i,k} - 3\,l_c\frac{\epsilon^u_{i,l,l}}{e^u_{i,l}}\prod_{k=3,4,5} e^u_{i,k} \\
&\quad + \frac{1}{3}\,l_c\sum_{j\in N_i^6}\frac{\epsilon^D_{i,j,0,l}}{e^D_{i,j,0}}\left(2\,e^D_{i,j,0} + 2\prod_{k=0,1,2} e^D_{i,j,k} + 3\prod_{k=0,\ldots,5} e^D_{i,j,k}\right) \\
&\quad + \frac{1}{3}\,l_c\sum_{j\in N_i^6}\left(\frac{\epsilon^D_{i,j,1,l}}{e^D_{i,j,1}} + \frac{\epsilon^D_{i,j,2,l}}{e^D_{i,j,2}} + \frac{\epsilon^D_{i,j,l,l}}{e^D_{i,j,l}}\right)\left(2\prod_{k=0,1,2} e^D_{i,j,k} + 3\prod_{k=0,\ldots,5} e^D_{i,j,k}\right) \\
\frac{\partial L}{\partial \sigma_i} &= \frac{-1}{\log 2}\frac{(z_i - u_{i,0})^2}{\sigma_i^3} + \frac{1}{\sigma_i} - g\sum_{j\in N_i^6}\frac{\partial e^\sigma_{i,j}}{\partial \sigma_i}\times e^D_{i,j,0}
\end{aligned}
$$

Bibliography

[1] A. Blake and A. Zisserman. *Visual Reconstruction*. MIT Press, Cambridge, MA, 1987.

[2] H. Freeman. Computer processing of line-drawing images. *Computing Surveys*, 6:57–97, March 1974.

[3] S. Geman and D. Geman. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(6):721–741, November 1984.

[4] B. K. P. Horn. *Robot Vision*. MIT Press, Cambridge, MA, 1986.

[5] Y. G. Leclerc. Constructing simple stable descriptions for image partitioning. *International Journal of Computer Vision*, 3:73–102, May 1989.

[6] J. Rissanen. Minimum description length principle. In S. Kotz and N. L. Johnson, editors, *Encyclopedia of Statistical Sciences, 5*, pages 523–527. Wiley, New York, 1985.