Optimization techniques on pixel neighborhood graphs for image processing

V.V. Mottl, A.B. Blinov, A.V. Kopylov, A.A. Kostin

Tula State University, Lenin Ave. 92, 300600 Tula, Russia
E-mail: mottl@atm.tsu.tula.ru

Abstract. A class of image processing problems is considered from the standpoint of treating them as those of co-ordinating the local image-dependent information and a priori smoothness constraints. Such a generalized problem is set as the formal problem of minimization of a separable objective function defined on an appropriate pixel neighborhood graph. For attaining a higher computation speed, the full pixel lattice is replaced by a succession of partial identical neighborhood trees. Two versions of a high-speed minimization procedure are proposed for, respectively, discretely defined and quadratic objective functions.

1 Introduction

Despite the dramatic variety of application problems of image analysis, it is possible to set off some single subclasses of problems which allow for treating them in the unified terms of respective standard mathematical optimization problems for which there exist effective methods of solving. One of such classes of image processing problems is considered in this paper.

We will consider an image as a numerical function $y_{\mathbf{t}}, \mathbf{t} = (t_1, t_2)$, usually, brightness, which takes values from an appropriate set $y \in \mathcal{Y}$, and is defined on the discrete image pixel grid $\mathbf{t} \in T$.

In this work, we restrict our concern to only those of image processing problems which can be represented as problems of transforming the original image $Y = (y_{\mathbf{t}}, \mathbf{t} \in T)$, $y \in \mathcal{Y}$, into another function $X = (x_{\mathbf{t}}, \mathbf{t} \in T)$, that would be defined on the same argument set $\mathbf{t} \in T$ and take values $x \in \mathcal{X}$ from a set specific for each particular problem.

As samples of such problems, at least, four of them should be mentioned, namely, the problems of smoothing, segmentation, image matching, and building the map of local texture orientation. These problems were considered in [1] from the statistical viewpoint as those of estimating the hidden Markov component of a two-component random field, in which the original image acted as the observable component. The assumed Markov property of the hidden field $X = (x_{\mathbf{t}}, \mathbf{t} \in T)$ was meant to express the *a priori* smoothness constraints of that or other kind, which usually follow from the nature of the respective practical problem.

However, the essence of both advantages and difficulties of such an approach lies, actually, not in its probabilistic aspect, but in the fact that the procedures of estimating the realization of the hidden random field $\hat{X}(Y)$ are those of solving a

Graph-Based Representations in Pattern Recognition (J.-M. Jolion and W.G. Kropatsch, ed.). Computing, Supplement 12. Springer-Verlag/Wien, 1998, pp. 135-145.

specific optimization problem $\hat{X}(Y) = \arg\min(or\max)J(X|Y)$. The specificity of the problem is that the variables constituting the vector to be evaluated $X = (x_{\mathbf{t}}, \mathbf{t} \in T)$ are associated with nodes of a neighborhood graph defined on the image pixel set, and that, due to the principal Markov assumption, the objective function $J(X|Y)$ is separable in accordance with this graph. This fact itself does not contribute essentially to an improvement of the computational solvability of the problem [2], until the neighborhood graph is assumed to have no cycles, i.e. to be a tree. It is just the latter assumption which leads to highly effective resulting algorithms and is systematically exploited within the bounds of the statistical approach to image analysis.

In particular, in [3] the usual lattice-like neighborhood graph is approximated by a combination of "horizontal" trees lying immediately in the image plane, and in [4] the neighborhood tree has the form of a "vertical" multiscale pyramid.

In this work, we consider the "horizontal" version of the generalized image processing problem specified above as that of finding the minimum of a separable objective function $J(X|Y)$ defined on the pixel neighborhood graph, without reference to any probabilistic interpretation. Such a function does not lend itself to an easy numerical minimization in the general case, therefore, we decompose it into a combination of partial objective functions each defined on a tree-like pixel-neighborhood subgraph that rests on all the pixels as its nodes.

The global minimization of each of the tree-supported separable objective functions is provided by a generalized dynamic programming procedure. Besides, we consider here two particular versions of such a procedure, when the set $x_{\mathbf{t}} \in \mathcal{X}$ the hidden variables take values from is finite $x_{\mathbf{t}} \in \mathcal{X} = \{1, \ldots m\}$, and when it is the n-dimensional real space $x_{\mathbf{t}} \in \mathcal{X} = \mathsf{R}^n$ with quadratic constituents of the separable objective functions. These versions are illustrated, respectively, by the problems of texture image segmentation and image matching.

2 The generalized image processing problem as that of minimization of a separable objective function defined on the pixel-neighborhood graph

As a rule, the essence of the particular image processing problem suggests a way to preliminarily judge, in a fuzzy form, about the value of the hidden field $x_{\mathbf{t}} \in \mathcal{X}$ at a single pixel $\mathbf{t}$ (node of the pixel-neighborhood graph) from some vicinity $Y_{\mathbf{t}}$ of this point in the original image Y. It is convenient to express such a judgment in the form of a function $\psi^o_{\mathbf{t}}(x_{\mathbf{t}}|Y)$ which is defined over the set $\mathcal{X}$ and takes the greater value the more evident is the contradiction between the hypothesis that $x_{\mathbf{t}} \in \mathcal{X}$ is just the "correct" local value we are seeking and the current image fragment $Y_{\mathbf{t}}$. Let such a function $\psi^o_{\mathbf{t}}(x_{\mathbf{t}}|Y)$ be called image-dependent node function, with the upper index "o" meant as *observation-based.*

However, practically inevitable interferences of various kinds usually prevent from taking the totality of locally optimal values $\hat{x}_{\mathbf{t}}(Y_{\mathbf{t}}) = \arg\min \psi^o_{\mathbf{t}}(x|Y_{\mathbf{t}})$, $x \in \mathcal{X}$, as the final decision on the sought-for hidden field $\hat{X}(Y) = [\hat{x}_{\mathbf{t}}(Y_{\mathbf{t}}), \mathbf{t} \in T]$.

On the other hand, the same essence of the particular application problem suggests, as a rule, also a fuzzy *a priori* knowledge on the expected form of the

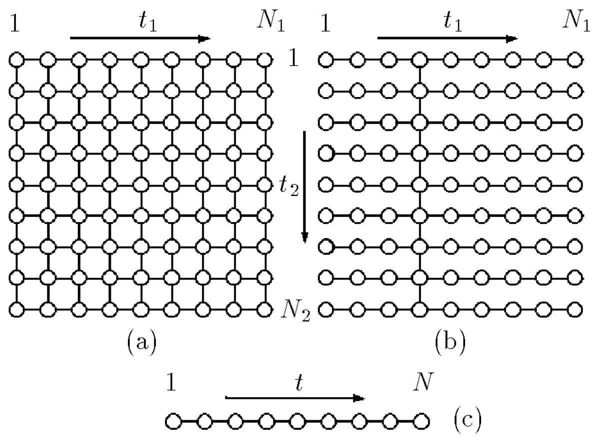

Figure 1: Neighborhood graphs on the pixel grid: a rectangular lattice (a), the simplest tree (b), and a chain (c).

hidden field. This knowledge often can be expressed in local terms as a function $\gamma^m_{\mathbf{t}',\mathbf{t}''}(x_{\mathbf{t}'}, x_{\mathbf{t}''})$ which is defined over $\mathcal{X} \times \mathcal{X}$ at each pair of neighboring pixels and takes the greater value the greater is the discrepancy between the respective hidden values. Let function $\gamma^m_{\mathbf{t}',\mathbf{t}''}(x_{\mathbf{t}'}, x_{\mathbf{t}''})$ be called *model-based* edge function. The pixel neighborhood depends on the particular problem, and is expressed, in the general case, by an undirected graph $G \in T \times T$.

Given an image Y, the decision on the hidden field $X = (x_{\mathbf{t}},\, \mathbf{t} \in T)$ should be sought for as a compromise between the mutually contradictory requirements to minimize both node and edge functions. It is natural to take the sum

$$J(X|Y) = \sum_{\mathbf{t}\in T} \psi^o_{\mathbf{t}}(x_{\mathbf{t}}|Y_{\mathbf{t}}) + \sum_{(\mathbf{t}',\mathbf{t}'')\in G} \gamma^m_{\mathbf{t}',\mathbf{t}''}(x_{\mathbf{t}'}, x_{\mathbf{t}''}) \tag{1}$$

as a combined criterion of decision making. Such a criterion is called here G-supported separable objective function. The number of its variables is equal to the number $|T|$ of nodes in G, i.e. the number of pixels.

Unfortunately, the intent to create a non-iterative procedure of finding the minimum point $\hat{X}$ of a separable function (1) on a supporting graph of general kind comes up against a fundamental barrier. The simplest form of the pixel neighborhood relation usually considered to be appropriate for images is lattice (Fig. 1,a). Such a graph does not possess any specific properties from the viewpoint of minimizing a separable function in comparison with the general case.

But in the particular case, when the graph has no cycles, i.e. is a tree, there exists a highly effective global optimization procedure, based on a recurrent decomposition of the initial problem of minimizing function (1) of $|T|$ variables into a succession of $|T|$ partial problems, each of which consists in minimizing a function of only one variable $x \in \mathcal{X}$.

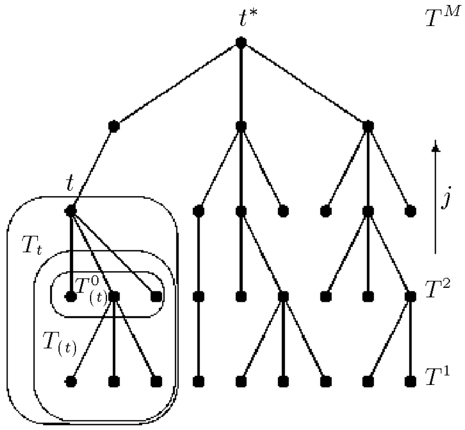

Figure 2: Structure of an arbitrary tree.

Below, in Section 5, we replace the neighborhood lattice by a combination of trees, each of which has no additional branching nodes except those of the stem, and differs from the others only by the position of this stem (Fig. 1,b).
In a still more particular case, when tree G has no branching nodes, i.e. is a chain (Fig. 1,c), such a procedure is nothing else than a version of the famous dynamic programming procedure.

3 The basic optimization procedure for a separable function supported by a tree

In this and the next Section, we don't associate the variables of the objective function, i.e. the nodes of the supporting tree, with variables in any certain application problem. Let G be a tree on the set of nodes $T = \{t\}$ (Fig. 2), and

$$J(X) = \sum_{t \in \mathsf{T}} \psi_t^o(x_t) + \sum_{(t',t'') \in G} \gamma^m_{t',t''}(x_{t'}, x_{t''}), \quad x_t \in \mathcal{X}, \tag{2}$$

be a G-supported separable objective function formed by, respectively, node $\psi_t^o(x_t)$ and edge functions $\gamma^m_{t',t''}(x_{t'}, x_{t''})$.
It is out of significance which of the nodes t^* is assigned the root of the tree, but once it is done, the node set gets partitioned into a number of subsets each consisting of the nodes of a certain level (Fig. 2):

$$T = \bigcup_{j=1}^{M} T^j, \quad T^j \bigcap T^k = \emptyset, \quad j \neq k.$$

We will denote the set of the nodes of the lowest level by T^1, and the set containing the only node of the highest level, namely, the root, by T^M.

Let, further, the tree $G_t \subset G$ formed by a node t and its descendants be called descendant tree of this node. It is clear that $G_t = \emptyset$ for the leaves, in particular, for all $t \in T^1$, and that $G_t = G$ for the root $t \in T^M$. The set of all the nodes in the descendant tree of a node t will be denoted by T_t, symbol $T_{(t)}$ will mean the same set without this node itself, i.e. the set of its descendants, and $T^0_{(t)} \subseteq T_{(t)}$ will denote the subset of its immediate (next) descendants. So, $T_t = \{t\}$ and $T^0_{(t)} = T_{(t)} = \emptyset$ for the leaves, and $T_{t*} = T$ for the root.
Analogously, symbols $X_t = (x_s,\ s \in T_t)$ and $X_{(t)} = (x_s,\ s \in T_{(t)})$ will mean the respective partial vectors of the node variables.
Finally, let us associate with each node t a partial objective function

$$J_t(X_t) = J_t(x_t, X_{(t)}) = \sum_{s \in T_t} \psi^o_s(x_s) + \sum_{(s',s'') \in G_t} \gamma^m_{s',s''}(x_{s'}, x_{s''})$$

which has absolutely the same structure as the full objective function (2) but is defined on the respective partial descendant tree. It is easy to see that

$$J_t(X_t) = J_t(x_t, X_{(t)}) = \psi^o_t(x_t) + \sum_{s \in T_{(t)}} \left\{\gamma^m_{t,s}(x_t, x_s) + J_s(x_s, X_{(s)})\right\}.$$

In particular, for the nodes of the lowest level $t \in T^1$, as well as for all the leaves, the partial objective function gets trivial $J_t(X_t) = J_t(x_t) = \psi^o_t(x_t)$, and for the root the partial function is nothing else than the full objective function $J_{t^*}(X_{t^*}) = J(X)$ (2).
The principal idea of the optimization procedure for separable objective functions supported by trees is based, like the classical dynamic programming procedure for chains, on the notion of the Bellman function

$$\tilde{J}_t(x_t) = \min_{X_{(t)}} J(x_t, X_{(t)}) = \psi^o_t(x_t) + \min_{X_s,\ s \in T^0_{(t)}} \sum_{s \in T^0_{(t)}} \left\{\gamma^m_{t,s}(x_t, x_s) + J_s(X_s)\right\},\ x_t \in \mathcal{X}.$$

The fundamental property of the Bellman function

$$\tilde{J}_t(x_t) = \psi^o_t(x_t) + \sum_{s \in T^0_{(t)}} \min_{x_s \in \mathcal{X}} \left\{\gamma^m_{t,s}(x_t, x_s) + \tilde{J}_s(x_s)\right\} \tag{3}$$

will be called the upward recurrent relation. The inverted form of this relation

$$\tilde{x}_s(x_t) = \arg\min_{x_s \in \mathcal{X}} \left\{\gamma^m_{t,s}(x_t, x_s) + \tilde{J}_s(x_s)\right\},\quad s \in T^0_{(t)}, \tag{4}$$

will be referred to as the downward recurrent relation.
The optimization procedure is based on the assumption that there exists a sufficiently effective way of solving the partial optimization problems occurring in (3), and storing, in a compact form, the Bellman functions or downward relations.
The procedure runs twice through all the nodes of the graph in order of their membership in the hierarchy levels T^j, first upwards $j = 1, \ldots, M$, and then downwards $j = M, \ldots, 2$.

It starts with the nodes of the lowest (first) hierarchy level $t \in T^1$, $j = 1$. As $T^0_{(t)} = \emptyset$ for these nodes,

$$\tilde{J}_t(x_t) = \psi^o_t(x_t), \quad x_t \in \mathcal{X}, \quad t \in T^1.$$

Thereupon, the upward recurrent relation (3) is applied sequentially to the nodes of the higher levels $j = 2, \ldots, M$. The Bellman functions $\tilde{J}_t(x_t)$ or downward relations $\tilde{x}_s(x_t)$, $x_t \in \mathcal{X}$, are to be stored for all the nodes of the tree except those of the first level.

The Bellman function at the root $\tilde{J}_{t^*}(x_{t^*})$, $x_{t^*} \in \mathcal{X}$, that is obtained at the last step of the upward pass, immediately gives the optimal value of the root variable:

$$\hat{x}_{t^*} = \arg \min_{x_{t^*} \in \mathcal{X}} \tilde{J}_{t^*}(x_{t^*}).$$

On the downward pass, as the procedure descends the hierarchy levels $j = M, \ldots, 2$, the already found optimal values of the current level determine optimal values of the variables at the immediately underlying level in accordance with the downward recurrent rules (4), stored as the result of the upward pass or determined by the stored Bellman functions (3)

$$\hat{x}_s = \tilde{x}_s(\hat{x}_t), \quad s \in T^0_{(t)}.$$

The degenerate particular case of a tree is tree without branching nodes, i.e. a chain. In this case, the above described optimization procedure becomes the classical dynamic programming procedure.

We introduce here an additional denotation $\hat{J}_t(x_t)$ for the so-called marginal node functions each of which shows how the full objective function depends on the value of a single variable at the respective node t, if other node variables take the conditionally optimal values in accordance with x_t:

$$\hat{J}_t(x_t) = \min_{x_s, s \in T, s \neq t} J(X).$$

At the root, the marginal node function is found at the last step of the upward pass $\hat{J}_{t^*}(x_{t^*})$. It can be shown that the marginal node functions at other nodes, each of which inevitably is an immediate descendant $s \in T^0_{(t)}$ of a node t of the higher level, satisfy the downward recurrent relation

$$\hat{J}_s(x_s) = \min_{x_t \in \mathcal{X}} \left\{ \hat{J}_t(x_t) + \left[\tilde{J}_s(x_s) + \gamma^m_{t,s}(x_t, x_s) \right] - \min_{x'_s \in \mathcal{X}} \left[\tilde{J}_s(x'_s) + \gamma^m_{t,s}(x_t, x'_s) \right] \right\}. \quad (5)$$

4 Numerical realization of the basic procedure for discretely defined and quadratic node and edge functions

A numerical realization of the basic procedure for a separable function supported by a tree, just like the classical dynamic programming procedure, is possible only in case there exists a finitely parametrized function family $\tilde{J}(x; \mathbf{a})$ concordant with node functions $\psi^o_t(x)$ and edge functions $\gamma^m_{t',t''}(x', x'')$ in the sense that the

Bellman functions (3) belong to this family at each step. In this case, the upward pass of the procedure consists in a recurrent re-evaluation of parameters $\tilde{\mathbf{a}}_t$ that completely represent the Bellman functions. We consider here two kinds of such a parametrization which present, in some sense, two extremes of the range of possible ways to define node and edge functions.

The evident case of a trivial parametrization is the case of a discrete range of node variables $\mathcal{X} = \{1, \ldots, m\}$. In this case, Bellman functions (3) are m-dimensional vectors, downward relations (4) are vectors, too, and the elementary enumeration gives an easy solution of the partial optimization problems.

Another extreme case is the assumption that the range of node variables is n-dimensional real space $\mathbf{x} \in \mathcal{X} = \mathsf{R}^n$, and both node and edge functions are quadratic with positive definite Hessian matrices:

$$\begin{aligned} &\psi_t^o(\mathbf{x}) = (\mathbf{x} - \mathbf{x}_t^0)^T \mathbf{Q}_t^0 (\mathbf{x} - \mathbf{x}_t^0), \\ &\gamma_{t',t''}^m(\mathbf{x}', \mathbf{x}'') = (\mathbf{x}' - \mathbf{x}'')^T \mathbf{U}_{t',t''} (\mathbf{x}' - \mathbf{x}''). \end{aligned} \tag{6}$$

It can be shown that, in accordance with (3), the Bellman functions will be likewise quadratic

$$\tilde{J}_t(\mathbf{x}_t) = \text{const} + (\mathbf{x} - \tilde{\mathbf{x}}_t)^T \tilde{\mathbf{Q}}_t (\mathbf{x} - \tilde{\mathbf{x}}_t), \quad \mathbf{x} \in \mathsf{R}^n,$$

and, so, completely represented by their minimum points $\tilde{\mathbf{x}}_t \in \mathsf{R}^n$ and positive definite Hessian matrices $\tilde{\mathbf{Q}}_t$ $(n \times n)$ to be found on the upward pass:

$$\tilde{\mathbf{Q}}_t = \mathbf{Q}_t^0 + \sum_{s \in T_{(t)}^0} \left[\tilde{\mathbf{Q}}_s^{-1} + \mathbf{U}_{t,s}^{-1}\right]^{-1}, \; \tilde{\mathbf{x}}_t = \tilde{\mathbf{Q}}_t^{-1} \left\{\mathbf{Q}_t^0 \mathbf{x}_t^0 + \sum_{s \in T_{(t)}^0} \left[\tilde{\mathbf{Q}}_s^{-1} + \mathbf{U}_{t,s}^{-1}\right]^{-1} \tilde{\mathbf{x}}_s\right\}. \tag{7}$$

Analogously, it can be derived from (4), (6) and (7), that the downward recurrent relations will be linear

$$\begin{aligned} \tilde{\mathbf{x}}_s(\mathbf{x}_t) &= (\mathbf{I} - \tilde{\mathbf{H}}_s)\tilde{\mathbf{x}}_s + \tilde{\mathbf{H}}_s \mathbf{x}_t = \tilde{\mathbf{x}}_s + \tilde{\mathbf{H}}_s(\mathbf{x}_t - \tilde{\mathbf{x}}_s), \quad s \in T_{(t)}^0, \\ \tilde{\mathbf{H}}_s &= (\tilde{\mathbf{Q}}_s + \mathbf{U}_{t,s})^{-1} \mathbf{U}_{t,s} = \left[\mathbf{U}_{t,s}^{-1} \tilde{\mathbf{Q}}_s + \mathbf{I}\right]^{-1}. \end{aligned}$$

Finally, all the marginal functions will be quadratic functions

$$\hat{J}_t(\mathbf{x}_t) = \text{const} + (\mathbf{x}_t - \hat{\mathbf{x}}_t)^T \hat{\mathbf{Q}}_t (\mathbf{x}_t - \hat{\mathbf{x}}_t)$$

defined by the final optimal values of the node variables and marginal Hessian matrices, both to be found on the downward pass in accordance with (5):

$$\hat{\mathbf{x}}_s = \tilde{\mathbf{x}}_s(\hat{\mathbf{x}}_t), \hat{\mathbf{Q}}_s = \left[\tilde{\mathbf{H}}_s \hat{\mathbf{Q}}_t^{-1} \tilde{\mathbf{H}}_s^T + \left(\tilde{\mathbf{Q}}_s + \mathbf{U}_{t,s}\right)^{-1}\right]^{-1}, \quad s \in T_{(t)}^0.$$

5 Tree approximation of the image pixel lattice and the generalized image processing technique

It is clear that there is no way to replace the lattice-like neighborhood graph (Fig. 1,a) by a tree-like one without loss in the ability to ensure smoothness of the secondary data field in all directions from each point of the image plane.

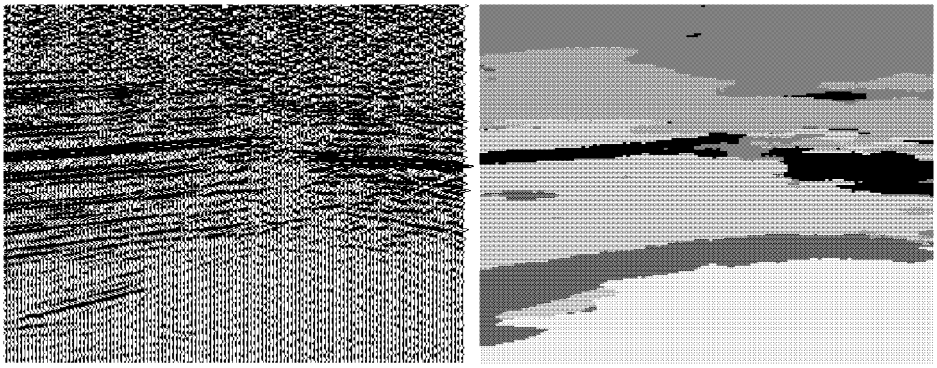

Figure 3: A vertical seismic section and the result of its segmentation into rock layers of six classes.

To avoid this obstacle, for finding the values of the hidden field at each vertical row of the picture, we use a separate pixel neighborhood tree which is defined, nevertheless, on the whole pixel set and has the same horizontal branches as the others (Fig. 1,b). The resulting image processing procedure is aimed at finding the optimal values only for the hidden variables at the stem nodes in each tree, and boils down to a combination of two usual dynamic programming procedures, each dealing with single image rows, respectively, the horizontal and vertical ones. First, such a one-dimensional procedure is applied to the horizontal rows $t_1 = 1, \ldots, N_1$ independently for each $t_2 = 1, \ldots, N_2$. The resulting marginal node functions $\hat{J}_{t_1,t_2}(x)$ have to be stored in the memory. Then, the procedure is applied to the vertical rows $t_2 = 1, \ldots, N_2$ independently for each $t_1 = 1, \ldots, N_1$ with the only alteration: the respective marginal node functions $\hat{J}_t(x)$, obtained at the first step, are taken instead of the image-dependent node functions $\psi_{\mathbf{t}}^o(x)$.

6 Discrete optimization: Segmentation of seismic sections considered as texture images

In the course of gas and oil reserves exploration, the so-called seismic sections are analyzed, that consist of synchronous records of reflected seismic signals registered by a large number of geophones (seismic sensors) placed along a straight line on the earth surface (Fig. 3). As the source of the initial seismic pulse, usually serves a series of explosions, responses to which are averaged in a special manner. Although the initial argument of the seismic signals is time, a relevant preprocessing allows to identify the vertical axis of the picture with depth under the respective sensor. Segmentation of a seismic section as a specific texture image is aimed at building a geological model of the layered rock mass.

In the problem of segmentation, the class allocation of the image pixels will be naturally expressed by a variable taking values from a finite set of class-membership indices $\mathcal{X} = \{1, \ldots, m\}$. So, the segmentation of the original image

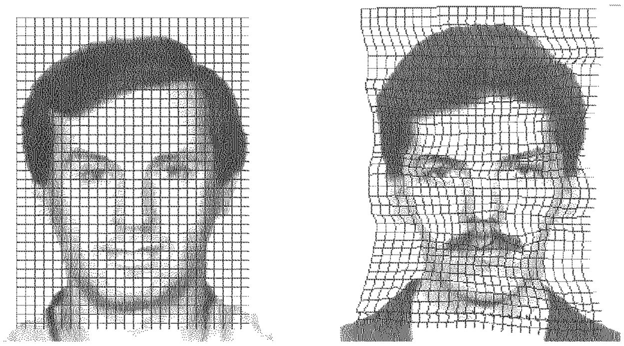

Figure 4: A pair of human portraits and disparity map of their identical points.

can be sought for in the form of a class allocation map $X = (x_{\mathbf{t}}, \mathbf{t} \in T)$.
Let us assume that the texture models are already chosen for all the classes of the sought-for homogeneous regions $x \in \mathcal{X} = \{1, \ldots, m\}$ and expressed each in the form of a linear autoregression model $y_{\mathbf{t}} \cong \sum a_{\mathbf{i}}^x y_{\mathbf{t}+\mathbf{i}}$, $\mathbf{i} \in I^0$, $\mathbf{i} \neq \mathbf{0}$, where the set of shifts I^0 determines the current image fragment $Y_{\mathbf{t}}$. Then, the image-dependent node functions, discrete in this case, can be taken in the form

$$\psi_{\mathbf{t}}^o(x|Y_{\mathbf{t}}) = \Big(y_{\mathbf{t}} - \sum_{\mathbf{i} \in I^0, \mathbf{i} \neq \mathbf{0}} a_{\mathbf{i}}^x y_{\mathbf{t}+\mathbf{i}}\Big)^2, \; x \in \mathcal{X} = \{1, \ldots, m\}.$$

As the simplest version of the model-based edge functions, meant to express the discrepancy between two neighboring values of the discrete hidden class-membership index, we used coincidence functions $\gamma_{\mathbf{t}',\mathbf{t}''}(x_{\mathbf{t}'}, x_{\mathbf{t}'}) = 0$ if $x_{\mathbf{t}'} = x_{\mathbf{t}''}$, and $\gamma_{\mathbf{t}',\mathbf{t}''}(x_{\mathbf{t}'}, x_{\mathbf{t}'}) = \alpha_{\mathbf{t}',\mathbf{t}''}$ if $x_{\mathbf{t}'} \neq x_{\mathbf{t}''}$ with much greater value of $\alpha_{\mathbf{t}',\mathbf{t}''}$ at horizontal edges to emphasize the expected horizontally stretched form of the sought-for rock layers. The result of segmentation is shown in Fig. 3.

7 Quadratic optimization: Matching of two images of similar structure

Let two images $Y^l = (y_{\mathbf{t}}^l, \mathbf{t} \in T)$, $Y^r = (y_{\mathbf{t}}^r, \mathbf{t} \in T)$ have the same range of definition T and form an entire image pair $Y = (Y^l, Y^r)$ to be analyzed as a whole, for instance, the left and the right image of a stereo pair, or photos of two human faces taken on the same scale (Fig. 4), and the like. Whatever objective might be pursued by the analysis of a pair of similar images, the first and algorithmically most difficult stage is matching, i.e. establishing a correspondence in the two images between the point pairs which could be treated as identical. The result of such an operation can be represented in the form of the so-called disparity map which, for each point $\mathbf{t}$ of the one of the images (conventionally taken as reference image), determines vector $\mathbf{x}_{\mathbf{t}} = (x_{\mathbf{t}}^1, x_{\mathbf{t}}^2) \in \mathsf{R}^2$ indicating the spatial shift between $\mathbf{t}$ and its match $\mathbf{t}' = \mathbf{t} + \mathbf{x}_{\mathbf{t}}$ in the second (search) image with eventual interpixel interpolation [5]. Thus, the result of matching of two images

$Y = (Y^l, Y^r)$ should be represented as a secondary vector data field (disparity map) $\mathbf{X} = (\mathbf{x_t}, \mathbf{t} \in T)$, $\mathbf{x} \in \mathcal{X} = \mathsf{R}^2$, with the same range of definition as the reference image.
As universally adopted, a reference (left) picture element $\mathbf{t}$, represented by some vector of its local features $\mathbf{z}^l_{\mathbf{t}}$, is to be compared with the set of analogously represented search (right) picture elements $\mathbf{z}^r_{\mathbf{t+x}}$ accessible from $\mathbf{t}$ by admissible shifts $\mathbf{x}$ forming the elementary fragment $Y_{\mathbf{t}}$ of the image pair. The quadratic approximation of the squared Euclidean norm of the vector feature difference within the admissible shift set

$$\psi^o_{\mathbf{t}}(\mathbf{x}) = c + (\mathbf{x} - \mathbf{x}^o_{\mathbf{t}})^T \mathbf{Q}^o_{\mathbf{t}} (\mathbf{x} - \mathbf{x}^o_{\mathbf{t}}) \cong ||\mathbf{z}^l_{\mathbf{t}} - \mathbf{z}^r_{\mathbf{t+x}}||^2$$

will be an appropriate image-dependent node function in which the eigenvectors and eigenvalues of $\mathbf{Q}^o_{\mathbf{t}}$ carry an information on the plausibility of deflections of the local decision $\mathbf{x_t}$ in different directions from the most likely value $\mathbf{x}^o_{\mathbf{t}}$.

8 Conclusions

Many image processing problems may be represented as those of transformation of the original image into a secondary data field on the same pixel grid. The essence of the problem consists in coordinating the mutually contradictory image-dependent information and model-based smoothness constraints both defined on the pixel neighborhood graph. At the same time, it is just this aspect of the image processing techniques which implies the major part of the amount of processing operations. We have shown that such a generalized problem can be solved by a high-speed procedure of minimizing a separable function defined on a succession of pixel neighborhood trees. The specificity of each particular image processing problem falls only on the way of extracting single units of the local information on the hidden field immediately from the original image, i.e. on the forming of an appropriate image-dependent node function.

References

[1] Mottl, V.V., Muchnik, I.B., Blinov, A.B., Kopylov, A.V.: Hidden tree-like quasi-Markov model and generalized technique for a class of image processing problems. 13th International Conference on Pattern Recognition. Vienna, Austria, August 25-29, 1996. Track B, pp. 715-719.

[2] Geman, S., Geman, D.: Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. IEEE Trans. on PAMI, 6, pp. 721-741 (1984).

[3] Wu, C.-H., Doershuk, P.C.: Tree approximation to Markov random fields. IEEE Trans. Patt. Anal. and Mach. Intel., 17, No. 4, pp. 391-402 (1995).

[4] Luettgen, M.R., Karl, W.C., Willsky, A.S., Tenney, R.R.: Multiscale representation of Markov random fields. IEEE Trans. Sign. Processing, 41, No. 12, pp. 3377-3396 (1993).

[5] Mottl, V.V., Kopylov, A.V., Blinov, A.B., Zheltov, S.Yu.: Quasi-statistical approach to the problem of stereo image matching. Proceedings SPIE, 2363, pp. 50-61 (1994).