Numer. Math. Theor. Meth. Appl. Vol. **8**, No. 2, pp. 283-312
doi: 10.4208/nmtma March 2015

Review of Methods Inspired by Algebraic-Multigrid for Data and Image Analysis Applications

Meirav Galun[1],[*], Ronen Basri[1] and Irad Yavneh[2]

[1] *Weizmann Institute of Science, Rehovot, Israel.*
[2] *Technion, Israel Institute of Technology, Haifa, Israel.*

Received 05 June 2014; Accepted 05 February 2015

Abstract. Algebraic Multigrid (AMG) methods were developed originally for numerically solving Partial Differential Equations (PDE), not necessarily on structured grids. In the last two decades solvers inspired by the AMG approach, were developed for non PDE problems, including data and image analysis problems, such as clustering, segmentation, quantization and others. These solvers share a common principle in that there is a crosstalk between fine and coarse representations of the problems, with flow of information in both directions, fine-to-coarse and coarse-to-fine. This paper surveys some of these problems and the AMG-inspired algorithms for their solution.

AMS subject classifications: 65F10, 65F50, 65H10, 68U10
Key words: Algebraic Multigrid, Image Analysis, Data Analysis, Multiscale Algorithms, Multilevel Optimization.

1. Introduction

Multigrid methods were introduced in the 1960s and developed extensively in the 1970s [11], [12], [28], [29]. Originally devised for elliptic boundary value problems on structured grids, variants were introduced during these early years that were adapted to handling more and more complicated problems, including nonlinear problems, constraints, discontinuous coefficients and eigenvalue problems. The early 1980s saw the development of a new multigrid approach with far-reaching implications, namely, Algebraic Multigrid (AMG) ([13], [56], [50], [22], [14]). With this approach, the solvers were no longer restricted to problems defined on structured grids, nor necessarily discretized PDEs.

The fundamental idea of AMG was to choose a basic iterative method, the relaxation, and then devise a coarse-grid correction process that will be effective for reducing all error that was not efficiently eliminated by the relaxation. To this end, a

[*] Corresponding author. *Email address:* meirav.galun@weizmann.ac.il (Meirav Galun), ronen.basri@weizmann.ac.il (Ronen Basri), irad@cs.technion.ac.il (Irad Yavneh)

heuristic algorithm was then developed for selecting a subset of the variables to be designated as the coarse-grid variables, along with a suitable prolongation matrix, and a Galerkin procedure was employed to project the problem onto the subspace spanned by the coarse-grid variables. This was applied recursively, resulting in a complete multilevel algorithm, quite similar in spirit to the classical multigrid approach except that all operations were defined algebraically without the need to consider an underlying PDE or even a computational grid. In the time that followed, many AMG variants were developed for various problems, but the large majority of these still aimed at solving linear systems of equations [58], or, less often, eigenproblems [9], [39].

In more recent years, multilevel methods have been developed in an ad-hoc manner for problems that are quite different in nature from those for which classical AMG can be applied, and yet these methods have often been inspired by AMG. In this paper we survey such algorithms for data analysis problems, mostly associated with Machine Learning and Image and Signal Processing. We focus on the problems of image segmentation, clustering and quantization along with the closely related problem of Voronoi tessellations, Markov Random Field (MRF) energy minimization, and Multi-Dimensional Scaling (MDS). Common to all the methods we survey is the multilevel structure and the cross-talk between the different scales associated with the problems.

The problems are reviewed in Section 2. Section 3 begins with a brief description of the classical AMG algorithm, followed by a generic description of variational coarsening for general convex functionals, leading to the classical FAS nonlinear multigrid algorithm [11]. The methods inspired by AMG for data and image analysis applications are described in Section 4. First, a FAS-like multigrid solver for MDS is reviewed, together with an application. After this, we describe multilevel methods for scalar and vector quantization and centroidal Voronoi tessellations, moving on to a multiscale algorithm for MRF energy minimization and finally clustering and segmentation by multilevel weighted aggregation. Each method is accompanied with a pseudo-code algorithm.

2. Applications and goals

2.1. Multidimensional scaling (MDS)

Multidimensional scaling is a generic name for a family of algorithms that embed points in target metric space from approximate inter-point distances, measured in some other metric space. MDS is widely used in dimensionality reduction, data analysis and visualization applications such as representing complicated high-dimensional data structures by low-dimensional ones [8], [40].

Problem definition. Let $\triangle$ be a symmetric $n \times n$ matrix of geodesic distances δ_{ij} measured between n points of a Riemannian manifold. The goal is to find a set of points $x_1, ...x_n$ in $\mathbb{R}^m$, such that the embedding error is minimal. A functional commonly used in MDS literature is the stress function

$$s(X; \triangle, W) = \sum_{i<j} w_{ij}(d_{ij}(X) - \delta_{ij})^2, \tag{2.1}$$

where $X = \begin{bmatrix} x_1^T \\ ... \\ x_n^T \end{bmatrix}$ is an $n \times m$ matrix of coordinates in $\mathbb{R}^m$ and $d_{ij} = ||x_i - x_j||$. The symmetric $n \times n$ matrix of weights $W = (w_{ij})$ determines the relative contribution of distances to the embedding error criterion. It is quite challenging to minimize the stress functional as it is non-convex and its structured Hessian is dense.

2.2. Quantization

Quantization is the process of representing continuum space with only a finite number of representatives or representing an initially rich amount of discrete data with smaller amount of representatives. Rounding real numbers to the nearest integer is a simple form of *scalar* quantization. Representing color images with a smaller amount of colors is an example of *vector* quantization. For quantization in a variety of applications see [42], [27], [45], [26].

Problem definition. Let Ω be some input domain and let $p : \Omega \to \mathbb{R}$ be the probability density function of some random process over Ω. Let n be a known positive integer. A quantizer $q : \Omega \to \Omega$ is defined by n representatives $R = \{\vec{r}_i\}_{i=1}^n$ and n decision regions $\{D_i\}_{i=1}^n$, where for all i, $\vec{r}_i \in D_i$, and $\cup_i D_i = \Omega$. That is, $q(\Omega)$ is a piecewise constant approximation of Ω, where all $\vec{x} \in D_i$ are represented by $\vec{r}_i$.

In the continuum case, an optimal quantizer achieves minimal distortion, defined as the expectation of the quantization error squared,

$$\mathfrak{D}(q) = \mathbb{E}[\|\vec{x} - q(\vec{x})\|_2^2] = \sum_{i=1}^{n} \int_{D_i} \|\vec{x} \ - \vec{r}_i\|_2^2 \, p(\vec{x}) d\vec{x}, \tag{2.2}$$

where $\| \cdot \|_2$ denotes the L_2 norm. Note that the functional $\mathfrak{D}$ depends on the representatives $\{\vec{r}_i\}_{i=1}^n$ and the decision regions $\{D_i\}_{i=1.}^n$

In the discrete case, where $p(\vec{x})$ is nonzero at only a finite number of points $\{\vec{x}_i\}_{i=1}^n$, a similar functional is minimized

$$\mathfrak{D}(q) = \mathbb{E}[\|\vec{x} - q(\vec{x})\|_2^2] = \sum_{i=1}^{n} \sum_{\vec{x}_j \in D_i} \|\vec{x}_j - \vec{r}_i\|_2^2 \, p(\vec{x}_j). \tag{2.3}$$

Quantization and Voronoi tessellation ([18], [60], [41]) are closely related. Given an open set $\Omega \subseteq \mathbb{R}^n$, the set $\{V_i\}_{i=1}^n$ is called a *tessellation* of Ω if $V_i \cap V_j = \emptyset$ for $i \neq j$ and $\cup_{i=1}^n V_i = \bar{\Omega}$. Given a set of points$\{\vec{z}_i\}_{i=1}^n$ belonging to $\bar{\Omega}$, the *Voronoi* region $\hat{V}_i$ corresponding to the point $\vec{z}_i$ is defined by

$$\left(\hat{V}_i\right) = \{\vec{x} \in \Omega \mid \ \|\vec{x} - \vec{z}_i\|_2 < \|\vec{x} - \vec{z}_j\|_2 \ \text{for} \ j = 1, ..., n, \ \ j \neq i\}. \tag{2.4}$$

The points $\{\vec{z}_i\}_{i=1}^n$ are called *generators*. The set $\{\hat{V}_i\}_{i=1}^n$ is a *Voronoi tessellation*, and each $\hat{V}_i$ is referred to as the *Voronoi region* corresponding to z_i. Given the regions

$\{\hat{V}_i\}_{i=1}^n$ we can define the mass centroids

$$\vec{z}_i^* = \frac{\int_{\hat{V}_i} \vec{y}p(\vec{y})d\vec{y}}{\int_{\hat{V}_i} p(\vec{y})d\vec{y}}. \tag{2.5}$$

Then, a tessellation for which the points z_i that serve as generators for the Voronoi regions $\hat{V}_i$ are themselves the mass centroids of those regions is called a *centroidal Voronoi tessellation*. As proved in [18] a necessary condition for $\mathfrak{D}$ (2.2) to be minimized is that the $D_i's$ are the Voronoi regions corresponding to the $r_i's$ in the sense of (2.4) and simultaneously, the $r_i's$ are the centroids of the corresponding $V_i's$ in the sense of (2.5). This determines the relation between quantization and centroidal Voronoi tessellation.

2.3. Minimization of Markov Random Field (MRF) energy

Discrete energy minimization problems, in the form of factor graphs, or equivalently Markov or Conditional Random Field models (MRF/CRF) are ubiquitous in the field of image analysis. Their applications are diverse and range from image denoising, segmentation, motion estimation, and stereo, to object recognition and image editing. Szeliski *et al.* [57] conducted a comparative study on a specific class: MRF models with *smoothness-based priors*. Since then, the field has made rapid progress. Modern vision problems involve more complex models, and use machine learning techniques to train the model parameters and energies. These changes give rise to hard energy minimization problems that are fundamentally different from the ones considered by Szeliski *et al.* In particular, the pairwise interaction can be arbitrary, including *contrast-enhancing priors*. A comparative study of modern discrete energy minimization problems is reviewed by [31].

Problem formulation. We consider discrete *pairwise* minimization problems defined over a graph $(\mathcal{V}, \mathcal{E})$ of the form

$$\mathcal{F}(L) = \sum_{i \in \mathcal{V}} \phi_i(l_i) + \sum_{(i,j) \in \mathcal{E}} \phi_{ij}(l_i, l_j), \tag{2.6}$$

where $\mathcal{V}$ is the set of variables and $\mathcal{E}$ is the set of edges and L is a discrete assignment, $L \in \{1, ..., l\}^n$ with $n = |\mathcal{V}|$ variables taking l possible labels. The main goal of discrete optimization is finding an assignment L^* which minimizes $\mathcal{F}(L)$ over all discrete assignments. Generally speaking, when dealing with *smoothness-based priors*, the sum over the pairwise terms, ϕ_{ij}, measures the extent to which the labeling of the variables is not piecewise smooth, while the sum over the unary terms, ϕ_i, measures the disagreement between x and the observed data. A classical example of a smoothness-based prior problem, is the stereo problem. Given a rectified pair of images, i.e., images which were transformed onto a common image plane, the goal is to find the disparity of each pixel in the reference (left) image, i.e., the difference in the location of a pixel seen by the left and right images. Each variable can take one of l discrete states, which represent the possible disparities at that point. For each possible disparity value, there is a cost associated with matching the pixel in the reference image to the corresponding pixel in the other (right) image at that disparity value. Typically, the cost of this unary term is based on the intensity difference between the two pixels. It relates how compatible a disparity value is with the observed intensity difference. The cost of the pairwise term expresses the disparity compatibility between neighboring pixels, typically yielding to the sought piecewise smooth disparity assignment. As mentioned above, in modern discrete energy minimization problems the model of smoothness-based prior is not necessarily appropriate. The pairwise interactions can be arbitrary, including contrast-enhancing priors.

Two main factors affect the difficulty of the optimization problem: the underlying graph structure $\mathcal{E}$ (i.e., tree or cyclic) and the properties of the pairwise interactions ϕ_{ij}. When the underlying graph is a tree, belief propagation [48] recovers a global optimum in a polynomial time. For graphs with cycles, recovery of global optimum can be achieved when the pairwise potentials satisfy the submodularity property [51]. For graphs with cycles and general pairwise potentials the problem is NP hard. However, when the pairwise potentials are smoothness-preserving there are approximation algorithms with theoretical guarantees on the quality of the approximation. A review about the hardness of the problem can be found in [5]. Details about existing methods can be found in Sec. 4.3 and in [10], [33], [57] and [31].

2.4. Data clustering and image segmentation

Clustering algorithms are useful in many fields, including machine learning, pattern recognition, image analysis, information retrieval, bioinformatics and others [16], [61], [59]. Generally, clustering is applied to a dataset, which is a collection of n $d-$dimensional vectors (data points) representing d measured features per sample. Given a data set, clustering algorithms seek a partition of the data to coherent groups, in a sense that data points in the same group (cluster) are more similar to each other than to those in other groups (clusters). Many approaches try to solve the clustering problem by optimizing a global functional, expressed in terms of the local similarities between data points. Spectral clustering methods start from the formulation of the normalized cut functional and attempt to solve a related generalized eigenvector problem [55], [46], [59]. In other words, spectral clustering algorithms typically start from *local* information encoded in a weighted graph on the data and cluster according to the *global* eigenvectors of the corresponding similarity matrix. Nadler *et al.* [43] explore the fundamental limitations of spectral clustering, showing that the first few eigenvectors of such similarity matrices cannot successfully cluster datasets that contain structures at different scales of size and density.

Problem formulation. The clustering problem can be formulated as seeking coherent groups in a weighted graph. The exact mathematical goal of the clustering problem is usually determined as part of the proposed algorithm, e.g., k-means, average linkage, hierarchical clustering, etc. Therefore, the problem formulation given below is quite related to the multilevel clustering algorithms which are described in this review.

Given a dataset of n points $\{x_i\}_{i=1}^n \subset R^d$ and an adjacency matrix W which encodes the similarity weights between the data points, a weighted graph $G = (V, W)$ is constructed as follows. Each data point i is represented by a graph node $i \in V$ where $V = \{1, ..., n\}$. For every two adjacent nodes i and j a weighted edge is constructed with the corresponding weight w_{ij} (w_{ii} is set to zero). To evaluate the coherence of the clusters, every cluster $S \subseteq V$ is associated with a state vector $u = (u_1, ..., u_n)$ representing the assignments of data points to it

$$u_i = \begin{cases} 1 & i \in S \\ 0 & i \notin S. \end{cases} \tag{2.7}$$

The coherence measure associated with S is defined by the normalized-cut like measure

$$\Gamma(S) = \frac{\sum_{i>j} w_{ij}(u_i - u_j)^2}{\sum_{i>j} w_{ij} u_i u_j}, \tag{2.8}$$

i.e., it sums the weights along the boundaries of S divided by the total sum of the internal weights. In matrix notation

$$\Gamma(S) = \frac{2u^T L u}{u^T W u}, \tag{2.9}$$

where L is the graph Laplacian matrix whose elements are

$$l_{ij} = \begin{cases} -w_{ij} & i \neq j \\ \sum_{k\ (k \neq i)} w_{ik} & i = j. \end{cases} \tag{2.10}$$

Clusters with small values of $\Gamma(S)$ are considered coherent. Therefore, the aim in a clustering task is to reveal groups of data points having small values of $\Gamma(S)$. Note that we are interested in clusters at different scales. In particular, clusters which are not too big, e.g., ($\leq |V|/2$). Otherwise, the problem becomes relatively easy. Finally, image segmentation is a special case of clustering, where the geometric layout, e.g., $4-$connected or $8-$connected graph, is simple and known. Prominent works in image segmentation are for example [55], [23], [2].

3. Methods

3.1. Classical algebraic multigrid (AMG)

"Algebraic multigrid (AMG) solves linear systems based on multigrid principles, but in a way that depends only on the coefficients in the underlying matrix" [22].

Historically, Classical AMG was introduced by Brandt, McCormick and Ruge in 1982 [13]. It was explored early on by Stueben in 1983 [56], and popularized by Ruge and Stueben in 1987 [50].

Algebraic multigrid is a method for solving linear systems based on multigrid principles, that requires no explicit knowledge of the problem geometry. AMG determines coarse grids, inter-grid transfer operators, and coarse grid equations based solely on the matrix entries.

Derivation by variational coarsening

We are interested in solving the linear system

$$Au = f, \tag{3.1}$$

where A is a real symmetric positive definite $n \times n$ matrix and $u, f \in \mathbb{R}^n$. Clearly, we need to answer the question how to choose the operators of the multigrid scheme, i.e., the interpolation (prolongation) $P \in \mathbb{R}^{n\times N}$, which maps the coarse grid to the fine grid (with N denoting the number of coarse-grid variables), the restriction $R \in \mathbb{R}^{N\times n}$ which maps the fine grid to the coarse grid, and the coarse-grid operator A_c.

Let us recast the problem (3.1) as the following equivalent convex optimization problem

$$u = \text{argmin}_{v\in\mathbb{R}^n}(\frac{1}{2}v^TAv - v^Tf). \tag{3.2}$$

Assume that a current approximation $\tilde{u}$ is obtained by a few local relaxation sweeps over the original system (3.1). Then, the aim is to add a correction Pe_c that will reduce the fine-grid functional as much as possible, where $e_c \in \mathbb{R}^N$ denotes the realization of the coarse grid variable v_c. Note that the set of possible corrections is the space spanned by the columns of P. Plugging $v = \tilde{u} + Pe_c$ into the functional (3.2) yields

$$e_c = \text{argmin}_{v_c\in\mathbb{R}^N}\frac{1}{2}(\tilde{u} + Pv_c)^TA(\tilde{u} + Pv_c) - (\tilde{u} + Pv_c)^Tf \tag{3.3}$$

$$\Longleftrightarrow$$

$$P^TAPe_c + P^T(A\tilde{u} - f) = 0. \tag{3.4}$$

Therefore, given an interpolation matrix P, we can conclude that the coarse-grid operator is $A_c = P^TAP$, known as Galerkin coarsening, the restriction operator is $R = P^T$, and the coarse-grid problem is

$$A_ce_c = P^Tr, \tag{3.5}$$

where $r = f - A\tilde{u}$ is the residual. Therefore, the update on the fine grid will be $\tilde{u} \leftarrow \tilde{u} + Pe_c$. This multigrid scheme is known as the *correction scheme*. We still need to choose the important component, which is the interpolation operator P, and to determine the coarse grid. These details can be found in the book [14], and in the paper [22].

3.2. Variational coarsening for general convex functionals

In the previous section we considered the derivation of an AMG scheme for solving a linear system of equations. This is analogous to the minimization of a quadratic functional. In this section we consider the minimization of a general convex problem. While the variational coarsening derivation for the minimization of a (convex) quadratic functional yields the correction scheme (CS), the derivation of the variational coarsening for general convex functionals yields the well-known full approximation scheme (FAS). Below, we describe the derivation.

Our aim is to solve

$$u = \text{argmin}_{v \in \mathbb{R}^n} (F(v) - v^T f) \tag{3.6}$$

$$\Longleftrightarrow$$

$$\nabla F = f, \tag{3.7}$$

where $F : \mathbb{R}^n \to \mathbb{R}$ is a convex function. First, we introduce the coarse variable

$$u_c = R\tilde{u} + e_c \tag{3.8}$$

where $R \in \mathbb{R}^{N \times n}$ is the restriction operator and $\tilde{u}$ is the current approximation on the fine level. Note that now the coarse variables approximate the full solution rather than just the correction. Therefore, the update on the fine grid will be $\tilde{u} \leftarrow \tilde{u} + P(u_c - R\tilde{u})$.

The optimal coarse functional is now

$$\begin{aligned} F_c(u_c) &\triangleq F(\tilde{u} + Pe_c) \\ &= F(\tilde{u} + P(u_c - R\tilde{u})) \\ &= F((I - PR)\tilde{u} + Pu_c). \end{aligned} \tag{3.9}$$

Hence, on the coarse grid one aims to solve

$$\nabla F_c(u_c) = P^T f$$

$$\Longleftrightarrow$$

$$\nabla_{u_c} F((I - PR)\tilde{u} + Pu_c) = P^T f. \tag{3.10}$$

If the structure of F_c is similar to the structure of the original functional F (as in the quadratic case), then a recursion can be employed. Otherwise, an approximation of F_c is employed by using a simple convex approximation $\hat{F}_c$ to F_c. According to (3.9) and (3.8)

$$\begin{aligned} F_c(u_c) &= \hat{F}_c(u_c) + F_c(u_c) - \hat{F}_c(u_c) \\ &= \hat{F}_c(u_c) + F((I - PR)\tilde{u} + Pu_c) - \hat{F}_c(u_c) \\ &= \hat{F}_c(u_c) + F(\tilde{u} + Pe_c) - \hat{F}_c(R\tilde{u} + e_c). \end{aligned} \tag{3.11}$$

The derivation proceeds by expanding (3.11) to a first order approximation in e_c, as follows. The second term and the third term are approximated by a truncated Taylor series around $\tilde{u}$ and $R\tilde{u}$, respectively, yielding the convex approximate functional $F_c^1(u_c)$

$$F_c^1(u_c) \triangleq \hat{F}_c(u_c) + F(\tilde{u}) - \hat{F}_c(R\tilde{u}) + e_c^T(P^T\nabla F(\tilde{u}) - \nabla\hat{F}_c(R\tilde{u})). \tag{3.12}$$

Recall that $u_c = R\tilde{u} + e_c$, therefore

$$\nabla F_c^1(u_c) = \nabla\hat{F}_c(u_c) + P^T\nabla F(\tilde{u}) - \nabla\hat{F}_c(R\tilde{u}). \tag{3.13}$$

Then, the problem on the coarse grid becomes $\nabla F_c^1(u_c) = P^T f$ which yields

$$\nabla\hat{F}_c(u_c) = P^T(f - \nabla F(\tilde{u})) + \nabla\hat{F}_c(R\tilde{u}). \tag{3.14}$$

Essentially, this is the classical full approximation scheme (FAS) [11].

4. Methods inspired by AMG for data and image analysis applications

This section reviews papers that developed algorithms inspired by AMG, aiming at solving the problems described in Sec. 2.

4.1. Multidimensional scaling (MDS)

A common method for large-scale MDS problem is SMACOF (Scaling by Majorizing a Complicated Function), see Chapter 8 in [8]. SMACOF generally suffers from slow convergence. However, a substantial performance improvement can be achieved by embedding the SMACOF relaxation into a multigrid scheme. In this section, we first sketch the basic optimization algorithm and then describe the multigrid multidimensional scaling algorithm, presented in [15].

4.1.1. Scaling by majorizing a complicated function - SMACOF

SMACOF is a basic optimization algorithm for solving the problem of MDS, i.e., minimizing the stress function (2.1). It has dual representations, one from the majorization-minimization perspective and the other from the gradient-descent perspective.

The key idea of majorization methods is to replace iteratively the original complicated non-convex function $S(X)$ by an auxiliary function $\tau(X, Z)$, where Z is some value that is fixed during each iteration. The function $\tau(X, Z)$ is called a majorizing function of $S(X)$ if it satisfies the following requirements.

- The auxiliary function $\tau(X, Z)$ should be simpler to minimize than $S(X)$.
- The original function must always be smaller than or equal to the auxiliary function, i.e., $S(X) \leq \tau(X, Z)$.

- The auxiliary function should touch the surface at the supporting point Z, i.e., $S(Z) = \tau(Z, Z)$.

To understand the principle of minimizing a function by majorization, consider the following. Start with initialization at Z, derive $\tau(X, Z)$ and let the minimum of $\tau(X, Z)$ over X be attained at X^*. According to the majorization requirements above, the following chain of inequalities is obtained, $S(X^*) \leq \tau(X^*, Z) \leq \tau(Z, Z) = S(Z)$.

Therefore, an attractive property of the majorization algorithm is the generation of a non-increasing sequence of the original function values.

By some algebraic manipulations it can be shown that the stress function (2.1) satisfies

$$\begin{aligned} S(X) &= \sum_{i<j} w_{ij}(d_{ij}(X) - \delta_{ij})^2 \\ &= \sum_{i<j} w_{ij}\delta_{ij}^2 + \sum_{i<j} w_{ij} d_{ij}^2(X) - 2\sum_{i<j} w_{ij}\delta_{ij}d_{ij} \\ &= \sum_{i<j} w_{ij}\delta_{ij}^2 + Tr(X^T V X) - 2Tr(X^T B(X) X) \\ &\leq \sum_{i<j} w_{ij}\delta_{ij}^2 + Tr(X^T V X) - 2Tr(X^T B(Z) Z) \triangleq \tau(X, Z), \end{aligned} \tag{4.1}$$

where the elements of V are

$$v_{ij} = \begin{cases} -w_{ij} & i \neq j \\ \sum_{k\ (k \neq i)} w_{ik} & i = j, \end{cases} \tag{4.2}$$

and the elements of $B(Z)$ are

$$b_{ij} = \begin{cases} -w_{ij}\delta_{ij}d_{ij}^{-1}(Z) & i \neq j \quad \text{and} \quad d_{ij}(Z) \neq 0 \\ 0 & i \neq j \quad \text{and} \quad d_{ij}(Z) = 0 \\ -\sum_{k\ (k\neq i)} b_{ik} & i = j. \end{cases} \tag{4.3}$$

Therefore, the function $\tau(X, Z)$ is a simple majorizing function of the stress function $S(X)$. $\tau(X, Z)$ is a convex function which is quadratic in X. Its minimum can be obtained analytically by setting the gradient of $\tau(X, Z)$ to zero, that is

$$2VX - 2B(Z)Z = 0. \tag{4.4}$$

The iterative scheme which follows this relation,

$$X^{(k+1)} = V^{\dagger} B(X^{(k)}) X^{(k)}, \tag{4.5}$$

converges to a local minimum of $S(X)$ and is called SMACOF. Since, $\nabla S(X) = 2VX - 2B(X)X$, it can be shown that a SMACOF iteration is equivalent to weighted gradient descent with constant step size

$$X^{(k+1)} = X^{(k)} - \frac{1}{2}V^{\dagger}\nabla S(X^{(k)}). \tag{4.6}$$

4.1.2. Multigrid MDS (MG-MDS)

The optimization problem (2.1) $\min_X S(X)$ is equivalent to the solution of the non-linear equation $\nabla S(X) = 0$. The concept of the multigrid algorithm in this case is to solve the non-linear equation $\nabla S(X) = 0$ using a sequence of approximate solutions to non-linear problems of the form $\nabla S(X) = F$, solved on coarse grids. The term F arises from the residual transferred from finer levels. In the form of the optimization problem, the minimization is a function of the form

$$S(X) - Tr(X^T F) \tag{4.7}$$

whose gradient equals $\nabla S(X) - F$. It turns out that the linear term makes the function $S(X) - Tr(X^T F)$ unbounded from below. Therefore, in order to overcome this problem, a localization term is introduced to the original stress functional yielding the modified stress

$$\hat{S}(X; \triangle, W) = \sum_{i<j} w_{ij}(d_{ij}(X) - \delta_{ij})^2 + \lambda \sum_{j=1}^{m} \left(\sum_{i=1}^{n} x_{ij} \right)^2, \quad \lambda > 0. \tag{4.8}$$

Then, the function $\hat{S}(X) - Tr(X^T F)$ is bounded from below. Note that the localization term resolves the translation ambiguity by restricting the center of mass to be at the origin, but has no other effect on the solution.

The multigrid method described in this review paper is suitable for MDS problems where the underlying geometry is explicitly available in addition to the matrix of distances $\triangle$. However, this is not the general case. In general, the MDS problem can be applied to the case in which only the matrix $\triangle$ is given and the geometry of the original points is not available. For further generalization of the multigrid MDS algorithm, one can refer to the paper [15].

Hierarchy of grids. A hierarchy of grids $\Omega_1 \supset \ldots \supset \Omega_R$ is constructed, where R is the coarsest level. Denote by N_r the number of grid points at the r^{th} level ($N_1 = n$). The selection of the grid points depends on the form in which the metric space is given. For example, if the surface is given by a triangulation representation, a hierarchy of grids can be constructed using the farthest point sampling algorithm [21].

Coarsening strategy. The restriction operator which transfers the information from level r to the coarser level $r+1$ is an $N_{r+1} \times N_r$ matrix, denoted by P_r^{r+1}. The interpolation operator from level $r+1$ to the finer level r is an $N_r \times N_{r+1}$ matrix, denoted

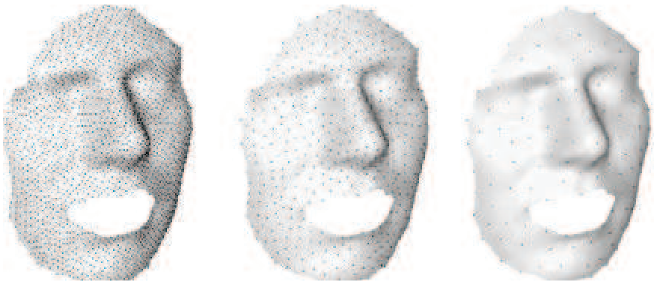

Figure 1: A grid hierarchy construction in the facial surface embedding problem, using the farthest point sampling algorithm. Retrieved from [15].

by P^r_{r+1}. These matrices are sparse and often chosen to satisfy $P^{r+1}_r = (P^r_{r+1})^T$. In the case of polyhedral representation, the triangulation is employed to determine the neighbors of each grid point and construct P^{r+1}_r and P^r_{r+1}.

The optimization problem is transferred to the next coarser level by applying the classical Galerkin coarsening (see Sec. 3.1) to the matrices $\triangle$ and W, via the restriction operator $\tilde{P}^{r+1}_r$ (not necessarily equal to P^{r+1}_r)

$$\begin{aligned} \triangle_{r+1} &= \tilde{P}^{r+1}_r \triangle_r (\tilde{P}^{r+1}_r)^T \\ W_{r+1} &= \tilde{P}^{r+1}_r W_r (\tilde{P}^{r+1}_r)^T. \end{aligned} \tag{4.9}$$

This coarsening strategy generates a hierarchy of problems of the form

$$\hat{S}_r(X_r, F_r) \triangleq \hat{S}(X_r; \triangle_r, W_r) - Tr(X_r^T F_r) \tag{4.10}$$

that need to be solved approximately at each level.

Relaxation. SMACOF-type iterations are used at the relaxation stage of the MG-MDS algorithm. The gradient of a typical function (4.10) has two extra terms, attributed to the quadratic localization term in the modified stress function and the linear residual term. In matrix notation, the gradient of $\hat{S}_r$ has the form

$$2VX - 2B(X)X - F + \lambda Z(X) \tag{4.11}$$

where the elements of the matrix $Z(X)$ defined as $z_{ij} = \sum_k x_{kj}$. As in (4.6) an additive update form is used.

See Algorithm 1 for a pseudo-code for the multigrid MDS algorithm.

Experiments - Embedding of a facial surface

The facial surface contains 5263 points. A hierarchy of grids with three resolution levels (containing 1977, 492 and 128 points) was constructed by the farthest point sampling algorithm, see Fig. 1.

The resulting embedding of a facial surface and a comparison of the stress function convergence over different algorithms are shown in Figs. 2 and 3, respectively.

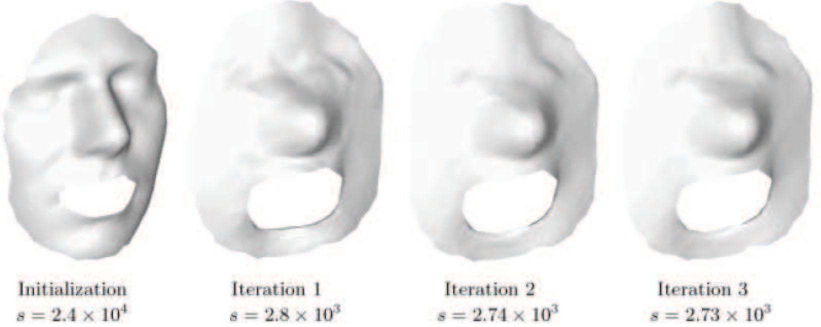

Figure 2: Embedding of a facial surface: as few as three MG-MDS iterations are sufficient in order to obtain a good expression-invariant representation. Retrieved from [15].

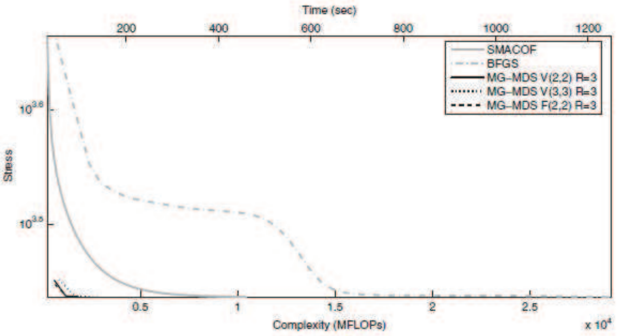

Figure 3: Convergence of the stress function of different algorithms in the facial surface embedding problem, initialization with the original points in $\mathbb{R}^3$. Retrieved from [15].

```
Algorithm 1: Multigrid MDS (MG-MDS) (FAS V-cycle), [15]
 Input: $\triangle, W$
 Output: $X$
 Init $r \leftarrow 1$// fine scale
 Vcycle $(X_r, F_r, \triangle_r, K_r, K_r')$
 if $r$=$R$ then
   solve $min_{X_R}(X_R, F_R)$ and return // coarsest level
else
     Relaxation: Apply $K_r$ SMACOF iterations to $S_r(X_r, F_r)$ and return $X_r'$
     Compute:
     $G_r' = \nabla S_r(X_r')$
     $X_{r+1}' = P_r^{r+1} X_r'$
     $G_{r+1}' = \nabla S_{r+1}(X_{r+1}')$
     $F_{r+1} = G_{r+1}' - P_r^{r+1} G_r'$
     Apply the Vcycle on level $r+1$:
    $X_{r+1}'' \leftarrow$ Vcycle$(X_r, F_r, \triangle_r, K_r, K_r')$ // Apply V-cycle on a coarser level
     Correction:
     $E_r = P_{r+1}^r (X_{r+1}'' - X_{r+1}')$
     $X_r'' \leftarrow X_r' + \alpha E_r$
     Relaxation: Apply $K_r'$ SMACOF iterations to $S_r(X_r'', F_r)$ and return $X_r'''$
```

Conclusion

The acceleration of the SMACOF relaxation by embedding it in a multiscale framework is impressive but not as significant as in the case of multiscale PDE solvers. Unlike PDEs, where the operators are local and therefore associated with sparse matrices, the MDS operators are of a global nature associated with dense matrices. Consequently, the smoothing properties of the SMACOF procedure is less effective than what we commonly see in PDE solvers.

4.2. Quantization and Centroidal Voronoi Tessellation (CVT)

The development of multilevel algorithms for quantization and the closely related (CVT) problems, emerged about a decade ago. This section reviews algorithms inspired by AMG for solving the quantization problem, which are presented [19], [20], [17], [36] and [37]. In Koren *et al.* [36], a non-linear multigrid algorithm was developed for the scalar quantization problem. The approach is based on the FAS algorithm described in Sec. 3.2, employing the classical Lloyd-Max iteration as a relaxation. In Du *et al.* [19], new algorithms were developed for the CVT problem, including a Lloyd-Newton iteration and a multilevel optimization approach.

4.2.1. Lloyd-Max (LM) iteration

A necessary condition for achieving minimal distortion (2.2) is the vanishing of the gradient, i.e.,

$$\begin{aligned} &\frac{\partial}{\partial \vec{r_i}}\mathfrak{D}(q) = 0 \qquad i = 1, ..., n \\ &\frac{\partial}{\partial D_i}\mathfrak{D}(q) = 0 \qquad i = 1, ..., n. \end{aligned} \tag{4.12}$$

These conditions yield

$$\begin{aligned} &\vec{r_i} = \frac{\int_{D_i} \vec{x} p(\vec{x}) d\vec{x}}{\int_{D_i} p(\vec{x}) d\vec{x}} \qquad i = 1, ..., n \\ &D_i = V(\vec{r_i}) \qquad\qquad i = 1, ..., n, \end{aligned} \tag{4.13}$$

where $V(\vec{r_i})$ is the Voronoi cell corresponding to r_i (the set of all points in the domain Ω that are closer to the representative $\vec{r_i}$ than to the other representatives). Lloyd-Max iteration [37] is an alternating process between the two optimality equations,

$$D_i^l = V((\vec{r_i})^l), \tag{4.14}$$

$$(\vec{r_i})^{l+1} = \frac{\int_{D_i^l} \vec{x} p(\vec{x}) d\vec{x}}{\int_{D_i^l} p(\vec{x}) d\vec{x}}, \tag{4.15}$$

that is, computing the Voronoi cells D_i's according to the representatives $\vec{r_i}$'s followed by the centers of mass $\vec{r_i}$'s according to the new D_i's, and so on. This process converges monotonically to a solution that satisfies (4.13).

4.2.2. The multilevel algorithm for the scalar quantization problem

In Koren *et al.* [36] the representative variables are eliminated from the Lloyd-Max equations, resulting in a non-linear optimization problem for the decision regions. The LM iteration, with suitable over relaxation, turns out to be very similar to an optimally damped Jacobi relaxation for a non-linear Poisson-like equation. Therefore, it is an extremely efficient smoother and can be incorporated successfully in a fairly standard FAS scheme.

4.2.3. Vector quantization and CVT

In Du *et al.* [19] the Lloyd-Newton algorithm is introduced for the CVT problem. In one approach the authors employ the classical AMG algorithm for solving the linear systems obtained by a block Gauss-Seidel pre-conditioner for the linearized problem. The authors refer to this approach as a global linearization method with an inner multigrid scheme. In a second approach the global linearization is avoided. Instead, a space decomposition technique is applied along with a successive subspace correction algorithm. Each step in this algorithm involves solving approximately a system of non-linear equations employing the Newton iteration as in the first approach, and updating the overall approximation successively. In Du *et al.* [20], the authors analyze a similar method for the scalar quantization problem and a rigorous uniform convergence theory. In Di *et al.* [17] the authors employ the MG/OPT approach [44]. In this approach a prescribed optimization method (opt) is used as a relaxation within a multilevel optimization algorithm. The coarse scale problem supplies a descent direction which is interpolated to the fine grid, and supplemented by a line search. This ensures a monotonic reduction of the fine grid functional, provided that opt itself is monotonically convergent.

An alternative multiscale approach was investigated in Koren *et al.* [37]. The authors define a wealth measure, which should asymptotically tend to a uniform function in an optimal quantizer. This measure is used to transfer representatives from dense regions to sparse regions. The density is defined in a multiscale manner: the region is considered dense if the wealth function is relatively high at all scales. The wealth function is a global measure, and thus it enables this approach to seek solutions that are better than those yielded by classical methods in problems with many local minima.

See Algorithm 2 for a pseudo-code for the MG/OPT algorithm to the CVT. The specific instantiation of the MG/OPT algorithm of [17] features simple injection as the restriction of the solution, standard full-weighting as the restriction of the gradient (residuals), linear interpolation as the prolongation and a truncated-Newton algorithm as the relaxation (whereby the Newton step direction is computed approximately using

Table 1: One dimensional example. Average residual reduction rate per iteration for the density function $p(x) = 6x^2e^{-2x^3}$. Retrieved from [36], and [17]. The convergence rate of MG/OPT is better, but the cost per iteration is higher than that of FAS, primarily due to the use of a much more elaborate relaxation.

number of representatives	Lloyd-Max	FAS [36]	MG/Opt [17]
16	0.9900	0.1725	0.0023
32	0.9973	0.1782	0.0177
64	0.9993	0.1847	0.0551
128	0.9998	0.1962	0.0673

Table 2: One dimensional example. Average convergence factor per iteration for the uniform density function, where OPT is the truncated-Newton method. Retrieved from [17].

number of representatives	OPT	MG/Opt [17]
16	0.1803	0.0234
32	0.2094	0.0440
64	0.2897	0.0952
128	0.4839	0.0604
256	0.6474	0.1174
512	0.8129	0.0932
1024	0.9076	0.0796
2048	0.9403	0.0728
4096	0.9679	0.0757

several Conjugate Gradient iterations).

Numerical experiments

Results of numerical experiments are given in the tables 1, 2 and 3.

Conclusion

The multilevel framework strategy is shown to be effective for the quantization problems considered in this section, in the sense that the convergence to a local minimum is fast. Indeed, the rate of multilevel convergence is comparable to that of multigrid PDE solvers.

4.3. Minimization of Markov Random Field (MRF) energies

In this section we first describe common energy minimization methods. We then review [35], [3] and [4] that developed algorithms which integrate these methods into a multiscale scheme, gaining improved performance.

Table 3: Two-dimensional example. Comparing convergence factor of 2-dimensional triangular domain with uniform density. Retrieved from [17].

number of representatives	Lloyd-Max	OPT	MG/Opt [17]
10	0.8323	0.0871	0.0127
55	0.9554	0.1378	0.0231
253	0.9891	0.1957	0.0121
1081	0.9973	0.4055	0.0092

```
Algorithm 2: MG/OPT to the CVT (V-cycle), [17]
 Input: $K, \Phi_0(y), k_1, k_2$
 Output: $z_0$ // The $K$ representatives
 Init // fine scale
 $r \leftarrow 0$   $v_r \leftarrow 0$ // initial estimate
$z_r^0$
 $z_r^{j+1} \leftarrow$MG/Opt $(z_r^j, v_r, k_1, k_2)$
 if $r$=$R$ then
    // solve the optimization problem on the coarsest level
     $z_r^{j+1} \leftarrow$ Opt$(\Phi_r, v_r, z_r^j)$
else
     Pre-smoothing: $\tilde{z}_r \leftarrow$ Opt$(\Phi_r, v_r, z_r^j, k_1)$
     Compute:
     $\tilde{z}_{r+1} = I_r^{r+1}\tilde{z}_r$
     $\tilde{v} = \nabla\Phi_{r+1}(\tilde{z}_{r+1}) + \hat{I}_r^{r+1}(v_r - \nabla\Phi_r(\tilde{z}_r))$
     // Apply MG/Opt to the surrogate model at level $r+1$
     $z_{r+1}^{+} \leftarrow$ MG/Opt$(\Phi_{r+1}, \tilde{v}, \tilde{z}_{r+1})$
     // Compute the search directions
     $e_{r+1} = z_{r+1}^{+} - \tilde{z}_{r+1}$ and $e_r = I_{r+1}^{r} e_{r+1}$
     // Use a line search to determine $\alpha$
     $z_r^{+} = \tilde{z}_r + \alpha e_r$ satisfying $\Phi_r(z_r^{+}) \leq \Phi_r(\tilde{z}_r)$
     Post-smoothing:  $z_r^{j+1} \leftarrow$ Opt$(\Phi_r, v_r, z_r^{+}, k_2)$;
```

where **Opt** is a convergent optimization algorithm in the following sense, $\lim_{j\to\infty} \|\nabla\Phi(z^j)\| = 0$ where $\{z^j\}$ are the iterates computed by **Opt**. $z^{+} \leftarrow$ **Opt**$(\Phi, v, \tilde{z}, k)$ applies k iterations of the convergent optimization algorithm to the problem $\min_z \Phi(z) - v^T z$ with initial guess $\tilde{z}$ to obtain z^{+}.

4.3.1. Iterated Conditional Modes (ICM)

Iterated conditional modes [7] uses a deterministic "greedy" strategy to find a local minimum. It starts with an estimate of the labeling, and then, for each variable in turn, it chooses the label giving the largest decrease of the energy function. This process is repeated until convergence, which is guaranteed to occur, and, in practice, is very rapid. However, the process is very sensitive to the initial estimate, cannot employ large moves and therefore tends to get stuck at local minima.

4.3.2. Large moves by graph cuts

The two most popular graph-cut algorithms, called the swap-move algorithm and the expansion-move algorithm, were introduced in [10]. Both algorithms work by repeatedly computing the global minimum of a binary labeling sub-problem in their inner loops. Solving each binary sub-problem lies in computing the minimum cut, which is done via max flow. The graph-cuts algorithms converge rapidly to a strong local minimum, in the sense that no "permitted move" will produce a labeling with lower energy.

The expand and swap moves are considered large scale operations because they allow the label of more than one variable to be changed at each step. It turns out that these algorithms perform large-scale moves and therefore can be considered as very large neighborhood search techniques. The $\alpha-$expand move can be described in the following way: the set of labels is traversed and at each step a label α is selected. Then all variables in the model are allowed to either keep their current label or change their label to α. The $\alpha - \beta$ swap algorithm operates similarly: at each step selecting two labels, α and β and allowing variables with these labels to either keep their current assignment or swap to the other label.

Essentially, the swap-move and expansion-move algorithms differ by the binary sub-problem they solve. For a pair of labels α, β, the binary step of the swap-move considers whether to retain the current label of a variable or switch between the labels. An expansion-move for a label α considers for each variable whether to retain its current label or flip it to label α. In the original work by Boykov *et al.* [10], the expansion-move algorithm was shown to be applicable to any energy where the pairwise term is a metric, and the swap-move algorithm, to any energy where the pairwise term is a semi-metric (a metric except that the triangle inequality may not hold). The work by Kolmogorov *et al.* [33] subsequently relaxed these conditions and showed that the expansion-move algorithm can be used if for all labels α, β and γ

$$\Phi_{ij}(\alpha,\alpha) + \Phi_{ij}(\beta,\gamma) \leq \Phi_{ij}(\beta,\alpha) + \Phi_{ij}(\alpha,\gamma) \tag{4.16}$$

and the swap-move algorithm can be used if for all labels α and β

$$\Phi_{ij}(\alpha,\alpha) + \Phi_{ij}(\beta,\beta) \leq \Phi_{ij}(\alpha,\beta) + \Phi_{ij}(\beta,\alpha). \tag{4.17}$$

It is shown in [33] that under these conditions the pairwise terms which are constructed by the binary assignment sub-problems satisfy the submodularity property, i.e.,

$$\theta_{pq}(0,0) + \theta_{pq}(1,1) \leq \theta_{pq}(0,1) + \theta_{pq}(1,0). \tag{4.18}$$

Then, a global minimum of each binary sub-problem can be computed in polynomial time by the minimum cut.

However, in many applications, none of these conditions is satisfied ((4.16) or (4.17) in a multilabel problem or (4.18) in a binary problem). Minimizing functions with arbitrary pairwise term is a very challenging task (in general, NP-hard). Rother *et al.* [49] suggested a successful method, called **QPBO(I)**, which can be applicable to non-submodular binary problems.

The multiscale algorithm, which is described in the next section, can handle arbitrary pairwise terms, either multilabel or binary.

4.3.3. Multiscale algebraic MRF coarsening

Here we consider a pairwise MRF of the form

$$\mathcal{F}(L) = \sum_{i \in \mathcal{V}} \phi_i(l_i) + \sum_{(i,j) \in \mathcal{E}} w_{ij} \cdot \phi(l_i, l_j). \tag{4.19}$$

The multiscale scheme for MRF coarsening suggested in [3] and [4] relies on an algebraic representation which is equivalent to the pairwise MRF. This equivalent algebraic representation forms the basis of the principled algorithm presented below, inspired by AMG. A related work by Komodakis [35] provides an algebraic multigrid formulation for MRF minimization in the dual space. However, despite the presentation of general formulation the paper provides examples using regular diadic grids of smoothness-preserving energies, i.e., energies which satisfy one of conditions ((4.16), (4.17), (4.18)).

The algebraic representation starts with the substitution of the assignment vector L in (4.19) by an equivalent binary matrix representation $U \in \{0,1\}^{n \times l}$. The rows of U correspond to the variables, and the columns correspond to labels: $U_{i\alpha} = 1$ iff $l_i = \alpha$. Expressing the energy (4.19) using U yields

$$\begin{gathered} E(U) = Tr(DU^T + WUVU^T) \\ \text{s. t. } U \in \{0,1\}^{n \times l}, \sum_{\alpha=1}^{L} U_{i\alpha} = 1, \end{gathered} \tag{4.20}$$

where $W = \{w_{ij}\}$, $D \in \mathbb{R}^{n \times l}$ with $D_{i\alpha} \triangleq \phi_i(\alpha)$, and $V \in \mathbb{R}^{l \times l}$ with $V_{\alpha\beta} \triangleq \phi_{\alpha\beta},\ \alpha, \beta \in \{1, ..., l\}$. Let (n^f, l, D^f, W^f, V) be the parametrization of the fine energy scale. We wish to generate a coarser representation (n^c, l, D^c, W^c, V) with $n^c < n^f$. This representation aims to approximate $E(u^f)$ with fewer variables, i.e., U^c is composed of n^c rows and l columns. An interpolation matrix $P \in [0,1]^{n^f \times n^c}$ with $\sum_j P_{ij} = 1\ \forall i$ maps coarse assignment U^c to fine assignment PU^c. Assume for now that the interpolation matrix P is known. Then, for any fine assignment U^f that can be approximated by a coarse assignment U^c, i.e., $U^f \approx PU^c$, the energy U^f can be approximated as follows.

$$\begin{aligned} E(U^f) &= Tr(D^f U^{f^T} + W^f U^f V U^{f^T}) \\ &\approx Tr(D^f U^{c^T} P^T + W^f P U^c V U^{c^T} P^T) \\ &= Tr((P^T D^f) U^{c^T} + (P^T W^f P) U^c V U^{c^T}) \qquad (4.21) \\ &= Tr(D^c U^{c^T} + W^c U^c V U^{c^T}) \triangleq E(U^c) \end{aligned}$$

where $D^c \triangleq P^T D^c$ and $W^c = P^T W^f P$.

Thus, the coarse energy $E(U^c)$ parametrized by (n^c, l, D^c, W^c, V) approximates the fine energy $E(U^f)$. This coarse energy is of the same form as the original energy, thus allowing to apply the coarsening procedure recursively to construct an energy pyramid. As in other multiscale schemes, the performance of the approximation heavily depends on the interpolation matrix P. Poorly constructed interpolation matrices will fail to expose the multiscale landscape of the functional. Below, we describe a principled energy-aware method for computing the interpolation matrix.

Energy-aware interpolation. Clearly, the suggested multiscale scheme approximates the original energy functional using a decreasing number of degrees of freedom, thus excluding some solutions from the original discrete search space. Which solutions are excluded is determined by the interpolation matrix P. *A good interpolation tends not to exclude low energy assignments at coarse levels.* The matrix P can be interpreted as an operator that aggregates fine scale variables into coarse ones. Aggregating fine variables i and j into a single coarser variable excludes from the search space all assignments for which $l_i \neq l_j$. This aggregation is undesired if assigning i and j to different labels yields low energy. However, if assigning $l_i = l_j$ yields low energy, then aggregating them together allows for efficient exploration of low energy assignments. Therefore, we need to answer the non-trivial question how to estimate the label agreement between variables.

A naive approach would assume that neighboring variables are always in agreement (e.g., [24], [35]). This assumption clearly does not hold in general and may yield a poor interpolation matrix. Recent work suggests using the energy itself to estimate variable agreements [32]. However, the suggested method is incapable of balancing the effect of the unary and pair-wise terms of the energy. Indeed, it is not straightforward to decide which term dominates and how to fuse these two terms together. [3] suggests an empirical scheme for agreement estimation that naturally integrates the influence of both unary and pair-wise terms. This is done by generating several samples with locally low energy assignments, and measuring the label agreement between neighboring variables in these samples. Starting with K random initializations of label assignments and employing t Iterated Conditional Modes (ICM) [7] iterations for each start, provides K samples $\{L^k\}_{k=1}^K$. Utilizing the label disagreement cost encoded in the matrix V, the disagreement between neighboring variable i and j is estimated as

$d_{ij} = \frac{1}{K}\sum_k V_{l_i^k, l_j^k}$, where l_i^k is the label of variable i in the sample L^k. The agreement is determined as $c_{ij} = \exp(-\frac{d_{ij}}{m})$, where m is proportional to the maximum value encoded in V. Now, we can think of the agreements $\{c_{ij}\}_{(i,j)\in\mathcal{E}}$ as a new set of weights of the graph. The next step is to select coarse nodes and to construct the interpolation matrix. This is done in the same manner as in the case of the multiscale graph coarsening, see Sec. 4.4.

See Algorithm 3 for a pseudo-code for the MRF energy minimization.

Algorithm 3: MRF energy minimization (F-cycle), [3]

```
Input: Energy $(n^0, l, D^0, W^0, V)$.
Output: $U^0$
Init $s \leftarrow 0$// fine scale
// Energy pyramid construction:
while $n^s \geq 10$ do
    Estimate pair-wise agreements $c_{ij}$ at scale $s$
    Compute interpolation matrix $P^s$
    Derive coarse energy $(n^s, l, D^s, W^s, V)$
    $s++$
// Coarse-to-fine optimization:
while $s \geq 0$ do
    $U^s \leftarrow$ Relax $(\tilde{U}^s)$
    $\tilde{U}^{s-1} = P^s U^s$// interpolate a solution
    $s--$
```

where **Relax** $(\tilde{U}^s)$ uses “off-the-shelf” algorithm to optimize the energy (n^s, l, D^s, W^s, V) with $\tilde{U}^s$ as an initialization.

Numerical experiments

The multiscale scheme for MRF coarsening suggested in [3] and [4] is evaluated on synthetic and real datasets. The protocol of the experiments follows [57] that uses the lower bound of TRW-S [34] as a baseline for comparing performance of different optimization methods. The reported results are the ratio between the resulting energy value and the lower bound (in percents), closer to 100% is better.

Synthetic results An *arbitrary* synthetic energy defined over a $4-$connected grid graph of size 50×50 and 5 labels. The terms in (4.20) are drawn from the following distributions: the unary term $D \sim \mathcal{N}(0,1)$, the pairwise term $V_{\alpha\beta} = V_{\beta\alpha} \sim \mathcal{U}(0,1)(V_{\alpha\alpha} = 0)$ and $w_{ij} = w_{ji} \sim \lambda \cdot \mathcal{U}(-1,1)$. The parameter λ controls the relative strength of the pairwise term, stronger (i.e., larger λ) results with more challenging energies to optimize (see [34]). Table 4 shows results averaged over an ensemble of 100 experiments. The multiscale algorithm ([3] and [4]) is compared to ICM, swap-move with binary sub-problem solved by QPBO, expansion-move with binary sub-problem solved by QPBO and TRW-S, which is considered a state-of-the-art method.

Table 4: Synthetic energies: showing percent of achieved energy value relative to the lower bound (closer to 100% is better) for varying strengths of the pairwise term ($\lambda = 5, 10, 15$, stronger $\rightarrow$ harder to optimize.). Retrieved from [4].

	ICM		Swap(QPBO)		Expand(QPBO)		TRW-S
λ	Multiscale	single scale	Multiscale	single scale	Multiscale	single scale	
5	112.6%	115.9%	108.9%	110.0%	110.5%	110.0%	116.6%
10	123.6%	130.2%	118.5%	120.2%	121.5%	121.0%	134.6%
15	127.1%	135.8%	122.1%	124.1%	124.6%	125.1%	138.3%

Table 5: Chinese character inpainting energies: (a) mean ratios of achieved energy value relative to baseline of Nowozin *et al.* [47] (lower is better, less than $100\% =$ lower than baseline) (b) percent of instances for which strictly lower energy than the baseline was achieved. Retrieved from [4].

	ICM		QPBO		TRW-S
λ	Multiscale	single scale	Multiscale	single scale	
(a)	114.0%	114.0%	97.8%	106.2%	108.6%
(b)	7.0%	7.0%	77.0%	34.0%	25.0%

Chinese character inpainting A *non-submodular* binary energy whose parameters are learned from a training set containing 100 instances over $64-$connected grid. These energies were designed and trained to perform the task of learning Chinese calligraphy [47]. As presented in Table 5, the multiscale algorithm ([3] and [4]) is compared to ICM, QPBO and TRW-S.

Conclusion

We see that it is possible to obtain relatively low energy values when applying an appropriate AMG-like strategy to MRF minimization problems, particularly when minimizing challenging energies. As demonstrated in Table 5, the performance of the solver depends heavily on the relaxation module, (e.g., ICM vs. QPBO). Moreover, the scheme presented here is a coarse-to-fine scheme. We believe that a two-way scheme (fine-to-coarse and coarse-to-fine) will allow improving the performance significantly, and this is currently under investigation.

4.4. Multiscale graph coarsening by AMG-like scheme

Here we review works in the fields of image segmentation and data clustering that employ multiscale graph coarsening by AMG-like scheme. The works appeared in a sequence of papers, [52], [53], [25] and [54], introducing boundary and texture considerations to the multiscale scheme. A further probabilistic multiscale graph coarsening and a new benchmark dataset are presented in [1]. Lastly, a work [30] in the field of cell microscopy introduced a scale-invariant saliency measure, related to the coherence measure (2.8), and applied the multiscale graph coarsening to segmentation and tracking of cells.

The AMG-like scheme seeks coherent clusters (image segments), i.e., groups of data points (pixels) with low values of $\Gamma(S)$ (2.8), by considering the similarity of the data points (pixels) at different resolutions, from fine scales to coarser ones. A key point in this approach is the calculation and incorporation of *scale-dependent global properties* such as the density, shape and orientation of a collection of data points as well as the average color and color variations of collection of pixels. These properties are called *aggregative features*. These aggregative features affect the formation of the coherent clusters (segments). To our best understanding it should be difficult to employ such use of multiscale aggregative features by any uniscale procedure, or by a formulation of the problem as a functional minimization problem, including spectral clustering on the fine scale data.

Starting from the given graph $G^{[0]} = G$, the AMG-like scheme recursively coarsens the coherence measure $\Gamma(S)$, creating a sequence of graphs $G^{[1]}, ..., G^{[k]}$ of decreasing size, yielding approximate solutions to low values of $\Gamma(S)$. Each node, at a certain scale, represents an *aggregate* which is the weighted collection of the original data points (pixels). Any low-Γ node emerging at a certain scale as a single node is considered a *cluster* (segment), i.e., *coherent aggregate*.

Similar to the classical AMG setting, the choice of the coarse variables, the design of the fine-to-coarse aggregation (or the coarse-to-fine interpolation), and the derivation of the coarse problem, are determined automatically. More precisely, the construction of a coarse graph from a given one is performed in three stages:

1. A subset of the fine nodes is chosen to serve as the *coarse nodes* or the *seeds* of the aggregates.

2. The rules for interpolation are determined, thereby establishing for each non-seed node to what fraction it belongs to each aggregate.

3. The weights of the edges between the coarse nodes are calculated.

Selection of coarse nodes. The formation of the set of seeds C and its complement, denoted by $F = V \backslash C$, is guided by the principle that each $F-$node should be "strongly coupled" to C, i.e., $\forall i \in F$, the following relation should be satisfied $\sum_{j \in C} w_{ij} \geq \alpha \sum_{j \in V} w_{ij}$, where α is a predefined parameter. This can be achieved either by a greedy approach or by an appropriate Integer Programming optimization.

Interpolation matrix. Each node in the chosen set C becomes the seed of an aggregate that will constitute one coarse scale node. For each node $i \in F$, its neighborhood in the coarse graph is determined, $N_i = \{j \in C, w_{ij} > 0\}$. Let $I(j)$ be the index in the coarse graph of the node that represents the aggregate around a seed whose index at the fine scale is j. The interpolation matrix P (of size $n \times N$, where $N = |C|$) is defined by

$$P_{iI(j)} = \begin{cases} \frac{w_{ij}}{\sum_{k \in N_i} w_{ik}} & \text{for } i \in F,\ j \in N_i, \\ 1 & \text{for } i \in C,\ j = i, \\ 0 & \text{otherwise.} \end{cases} \tag{4.22}$$

By selecting "strongly coupled" coarse nodes followed by establishing an appropriate interpolation matrix we ensure that a low Γ aggregate with a state vector $u = (u_1, ..., u_n)$ will be well represented via the interpolation P of a corresponding coarse level state vector $U = (U_1, ..., U_N)$, i.e., $u \approx PU$.

The coarse problem. Following the construction of P, the coherence measure can be approximated by the coarse state variable $U = (U_1, ..., U_N)$ as follows

$$\Gamma = \frac{2u^T Lu}{u^T Wu} \approx \frac{2U^T P^T LPU}{U^T P^T WPU}. \tag{4.23}$$

Therefore, in order to keep the same general formulation, inspired by the Galerkin coarsening (see Sec. 3.1), the coarse problem can be determined by $W^{\text{coarse}} = P^T WP$ and $L^{\text{coarse}} = P^T LP$. However, in practice the coarse problem is determined slightly differently by applying the weighted aggregation scheme [54], [38]. The weighted aggregation scheme allows to keep the structure of the graph constructed at coarse levels. An edge connecting two coarse aggregates p and q is assigned with the coupling weight

$$w_{pq}^{\text{coarse}} = \begin{cases} (P^T WP)_{pq} & p \neq q \\ 0 & p = q. \end{cases} \tag{4.24}$$

Intuitively, the coupling weight between a pair of coarse aggregates is the weighted sum of the coupling weights between their sub-aggregates. Exploiting the sparseness of the interpolation matrix P, the calculation of W^{coarse} is inexpensive and L^{coarse} is approximated by the relation (2.10). This coarsening procedure is performed recursively.

See Algorithm 4 for a pseudo-code for the multiscale graph coarsening for clustering and image segmentation. Note that in this case, there is no relaxation (or smoothing) step.

Experiments

In this section we demonstrate segmentation and clustering results, obtained using the multilevel graph coarsening. See Figs. 4, 5 and 6.

Conclusion

The AMG-like graph coarsening strategy enables us to treat data which contains clusters (segments) at different scales. Moreover, the calculation and integration of the

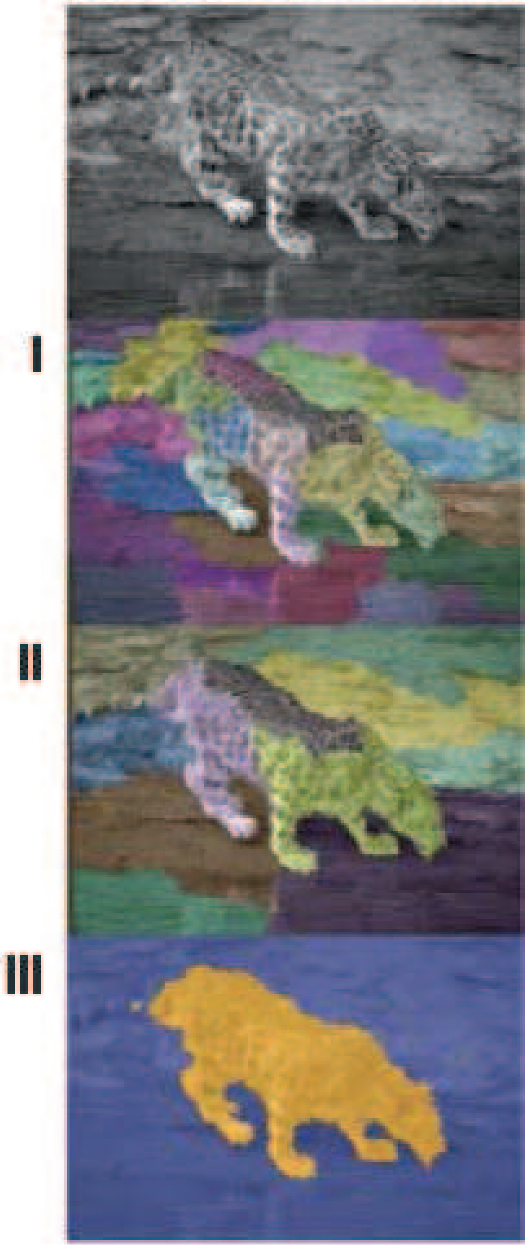

Figure 4: Image segmentation. A hierarchy composing a segment and its background. The leopard segment (III) is shown along with two out of ten levels of aggregates composing it (I, II). The original image is shown at the top. Retrieved from [54].

aggregative features as the graph coarsening proceeds affect significantly the formation of the coherent clusters (segments). This property distinguishes the AMG-like graph coarsening from other clustering methods.

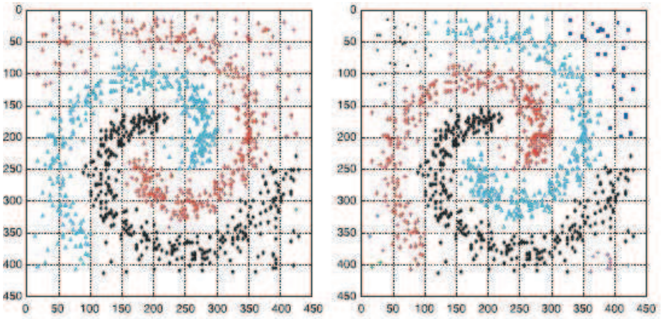

Figure 5: Clustering of data points in 2D. Inaccurate bottom-up clustering and cured top-down clustering: after first bottom-up process obtained with misclassifications (left), by applying top-down density refinement the background noise is separated from the clusters (right). Retrieved from [38].

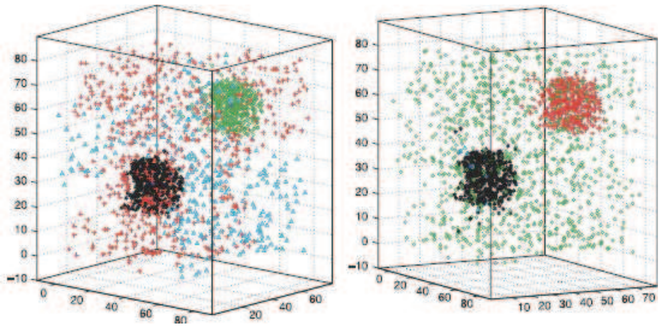

Figure 6: Clustering of data points in 3D. Inaccurate bottom-up clustering and cured top-down clustering: after first bottom-up process obtained with misclassifications (left), by applying top-down density refinement the background noise is separated from the clusters (right). Retrieved from [38].

Algorithm 4: Multiscale graph coarsening for clustering and image segmentation, [54] and [38].

```
Input: similarity matrix $W$
Output: salient clusters (segments) at different scales
Init $s \leftarrow 1$// fine scale
// Create coarser level graph
while $n^s > 1$ do
    Initialize the set of seeds $(C)$ and its complement $(F)$: $C = \emptyset, F = V^{s-1}$
    for all nodes $i \in F$ do
        If $\sum_{j \in C} w_{ij}^{s-1} < \alpha \sum_{k \in V^{s-1}} w_{ik}^{s-1}$ then $C \leftarrow \{C \cup \{i\}\}$; $F \leftarrow \{F \backslash \{i\}\}$
    Calculate $P_s^{s-1}$ the interpolation matrix
    Calculate aggregative properties
    Calculate $W^s$ coarse-level couplings by weighted aggregation and modify according to similarity in aggregative properties
    $s++$
// extract salient clusters (segments)
while $s \geq 1$ do
    Evaluate coherence measure of the clusters (segments)
    Determine the support of salient clusters (segments) by a top-down process
    $s--$
```

Acknowledgments

This research was supported in part by the Israel Science Foundation, Grant No. 764/10, the Israeli Ministry of Science, by the Citigroup Foundation and by Minerva Foundation. We also thank the Belfer Fund and the Amir Pnueli visiting scholar program. The vision group at the Weizmann Institute is supported in part by the Moross Laboratory for Vision Research and Robotics.

5. Conclusion

In this review we survey AMG-like algorithms for data and image analysis applications. We focus on the problems of clustering, image segmentation, quantization, discrete energy minimization and multi-dimensional scaling. Common to all the methods is the AMG-like structure and the fine-to-coarse and coarse-to-fine cross talk between the different scales. We believe that other applications, in signal processing for example, can be inspired by the AMG frameworks that we described.

References

[1] S. Alpert, M. Galun, A. Brandt, and R. Basri, *Image segmentation by probabilistic bottom-up aggregation and cue integration*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 34(2), pp. 315–327, 2012.

[2] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik, *Contour detection and hierarchical image segmentation*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 33(5), pp. 898–916, 2011.

[3] S. Bagon and M. Galun, *A multiscale framework for challenging discrete optimization*, NIPS Workshop on Optimization for Machine Learning, 2012.

[4] S. Bagon and M. Galun, *A unified multiscale framework for discrete energy minimization*, Report, arXiv:1204.4867, 2012.

[5] S. Bagon, *Discrete Energy Minimization, beyond Submodularity: Applications and Approximations*, PhD Thesis, the Weizmann Insitute of Science, arXiv:1210.7362, 2012.

[6] P. Berkhin, *A survey of clustering data mining techniques*, Grouping multidimensional data, Springer, pp. 25–71, 2006.

[7] J. Besag, *On the statistical analysis of dirty pictures*, Journal of the Royal Statistical Society, Series B (Methodological), pp. 259–302, 1986.

[8] I. Borg and P. J. Groenen, *Modern Multidimensional Scaling: Theory and Applications*, Springer, 2005.

[9] A. Borzi and G. Borzi, *Algebraic multigrid methods for solving generalized eigenvalue problems*, International journal for numerical methods in engineering, 65(8), pp. 1186–1196, 2006.

[10] Y. Boykov, O. Veksler and R. Zabih, *Fast approximate energy minimization via graph cuts*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 23(11), pp. 1222–1239, 2001.

[11] A. Brandt, *Multi-level adaptive solutions to boundary-value problems*, Mathematics of computation, 31(138), pp. 333–390, 1977.

[12] A. Brandt and O. E. Livne, *Multigrid techniques: 1984 guide with applications to fluid dynamics*, 67, SIAM, 2011.

[13] A. Brandt, S.F. McCormick and J.W. Ruge, *Algebraic multigrid (AMG) for automatic multigrid solutions with application to geodetic computations*, Report, Inst. for Computational Studies, Fort Collins, CO, October 1982.

[14] W. L. Briggs, V. E. Henson, and S. F. McCormick, *A Multigrid Tutorial*, 2nd Ed., SIAM, 2000.

[15] M. M. Bronstein, A. M. Bronstein, R. Kimmel, and I. Yavneh, *Multigrid multidimensional scaling*, Numerical linear algebra with applications, 13(2-3), pp. 149–171, 2006.

[16] F. Corpet, *Multiple sequence alignment with hierarchical clustering*, Nucleic acids research, 16(22), pp. 10881–10890, 1988.

[17] Z. Di, M. Emelianenko, and S. G. Nash, *Truncated newton-based multigrid algorithm for centroidal Voronoi diagram calculation*, Numerical Mathematics: Theory, Methods Applications 5, pp. 242–259, 2012.

[18] Q. Du, V. Faber, and M. Gunzburger, *Centroidal Voronoi tessellations: applications and algorithms* , J. SIAM Review, 41(4), pp. 637–676, 1999,

[19] Q. Du and M. Emelianenko, *Acceleration schemes for computing centroidal Voronoi tessellations*, Numerical Linear Algebra with Applications 13(2-3), pp. 173–192, 2006.

[20] Q. Du and M. Emelianenko, *Uniform convergence of a nonlinear energy-based multilevel quantization scheme*, SIAM Journal on Numerical Analysis 46(3), pp. 1483–1502, 2008.

[21] Y. Eldar, M. Lindenbaum, M. Porat and Y. Y. Zeevi, *The farthest point strategy for progressive image sampling*, IEEE Transactions on Image Processing, 6(9), pp. 1305–1315, 1997.

[22] R. D. Falgout, *An introduction to algebraic multigrid*, Computing in Science and Engineering, 8(6), pp. 24–33, 2006.

[23] P. D. Felzenszwalb and D. P. Huttenlocher, *Efficient graph-based image segmentation*, International Journal of Computer Vision, 59(2), pp. 167–181, 2004.

[24] P. F. Felzenszwalb and D. P. Huttenlocher, *Efficient belief propagation for early vision*, *International journal of computer vision*, 70(1), pp. 41–54, 2006.

[25] M. Galun, E. Sharon, R. Basri, and A. Brandt, *Texture segmentation by multiscale aggregation of filter responses and shape elements*, ICCV proceedings, pp. 716–723, 2003.

[26] A. Gersho and R. M. Gray, *Vector quantization and signal compression*, Springer, 1992.

[27] R. M. Gray, *Vector quantization*, ASSP Magazine, IEEE, 1(2), pp. 4–29, 1984.

[28] W. Hackbusch, *On the multi-grid method applied to difference equations*, Computing, 20(4), pp. 291-306, 1978.

[29] W. Hackbusch, *Multi-grid methods and applications*, 4, Berlin: Springer-Verlag, 1985.

[30] T. Inglis, H. D. Sterck, G. Sanders, H. Djambazian, R. Sladek, S. Sundararajan and T. J. Hudson, *Multilevel space-time aggregation for bright field cell microscopy segmentation and tracking*, Journal of Biomedical Imaging, 8, 2010.

[31] J. H. Kappes, B. Andres, F. A. Hamprecht, C. Schnörr, S. Nowozin, D. Batra, S. Kim, B. X. Kausler, J. Lellmann, N. Komodakis, and C. Rother, *A comparative study of modern inference techniques for discrete energy minimization problems*, CVPR, 2013.

[32] T. Kim, S. Nowozin, P. Kohli and C. D. Yoo, *Variable grouping for energy minimization*, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1913–1920, 2011.

[33] V. Kolmogorov and R. Zabih, *What energy functions can be minimized via graph cuts*, IEEE Trans. Pattern Analysis and Machine Intelligence, 26(2), pp. 147–159, 2004.

[34] V. Kolmogorov, *Convergent tree-reweighted message passing for energy minimization*,

IEEE Trans. Pattern Analysis and Machine Intelligence, 28(10), pp. 1568–1583, 2006.

[35] N. Komodakis, *Towards more efficient and effective LP-based algorithms for MRF optimization*, ECCV, pp. 520–534, 2010.

[36] Y. Koren, I. Yavneh, and A. Spira, *A multigrid approach to the scalar quantization problem*, IEEE Transactions on Information Theory, 51(8), pp. 2993–2998, 2005.

[37] Y. Koren, and I. Yavneh, *Adaptive multiscale redistribution for vector quantization*, SIAM Journal on Scientific Computing, 27(5), pp. 1573–1593, 2006,

[38] D. Kushnir, M. Galun, and A. Brandt, *Fast multiscale clustering and manifold identification*, Pattern Recognition, 39(10), pp. 1876–1891, 2006.

[39] D. Kushnir, M. Galun, and A. Brandt, *Efficient multilevel eigensolvers with applications to data analysis tasks*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 32(8), pp. 1377–1391, 2010.

[40] N. Linial, E. London, and Y. Rabinovich, *The geometry of graphs and some of its algorithmic applications*, Combinatorica, 15(2), pp. 215–245, 1995.

[41] Y. Liu, W. Wang, B. Lvy, F. Sun, D. M. Yan, L. Lu, and C. Yang, *On centroidal voronoi tessellationenergy smoothness and fast computation*, ACM Transactions on Graphics (ToG), 28(4), 2009.

[42] S. Lloyd, *Least squares quantization in PCM*, IEEE Transactions on Information Theory, 28(2), pp. 129–137, 1982.

[43] B. Nadler and M. Galun, *Fundamental limitations of spectral clustering*, Advances in Neural Information Processing Systems, 2006.

[44] S. G. Nash, *A multigrid approach to discretized optimization problems. Optimization Methods and Software*, 14(1-2), pp. 99–116, 2000.

[45] N. M. Nasrabadi and R. A. King, *Image coding using vector quantization: A review*, IEEE Transactions on Communications, 36(8), pp. 957–971, 1988.

[46] A. Y. Ng, M. I. Jordan and Y. Weiss, *On spectral clustering: Analysis and an algorithm*, Advances in neural information processing systems 2, pp. 849–856, 2002.

[47] S. Nowozin, C. Rother, S. Bagon, T. Sharp, B. Yao and P. Kohli *Decision tree fields*, IEEE International Conference on Computer Vision (ICCV), pp. 1668–1675, 2011

[48] J. Pearl, *Probabilistic reasoning in intelligent systems: networks of plausible inference*, 2nd Ed., Morgan Kaufmann, 1988.

[49] C. Rother, V. Kolmogorov, V. Lempitsky and M. Szummer, *Optimizing binary MRFs via extended roof duality*, IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–8, 2007.

[50] J. W. Ruge and K. Stuben, *Algebraic multigrid*, Multigrid methods 3, pp. 73–130, 1987.

[51] D. Schlesinger and B. Flach, *Transforming an arbitrary minsum problem into a binary one*, TU, Fak. Informatik, 2006.

[52] E. Sharon, A. Brandt, and R. Basri, *Fast multiscale image segmentation*, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2000.

[53] E. Sharon, A. Brandt, and R. Basri, *Segmentation and boundary detection using multiscale intensity measurements*, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2001.

[54] E. Sharon, M. Galun, D. Sharon, R. Basri, and A. Brandt, *Hierarchy and adaptivity in segmenting visual scenes*, Nature 442(7104), pp. 810–813, 2006.

[55] J. Shi and J. Malik, *Normalized cuts and image segmentation*, IEEE Trans. Pattern Analysis and Machine Intelligence, 22(8), pp. 888–905, 2000.

[56] K. Stuben, *Algebraic multigrid (AMG): experiences and comparisons*, Applied mathematics and computation, 13(3), pp. 419–451, 1983.

[57] R. Szeliski, R. Zabin, D. Scharstein, O. Veksler, V. Kolmogorov, A. Agarwala, M. Tappen, and C. Rother, *A comparative study of energy minimization methods for Markov Random Fields with smoothness-based priors*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 30(6), 2008.

[58] U. Trottenberg, C. W. Oosterlee, and A. Schuller, *Multigrid*, Academic press, 2000.

[59] U. Von Luxburg, *A tutorial on spectral clustering." Statistics and computing*, 17(4), pp. 395–416, 2007.

[60] D. F. Watson, *Computing the n-dimensional Delaunay tessellation with application to Voronoi polytopes*, The computer journal, 24(2), pp. 167–172, 1981.

[61] R. Xu and D. Wunsch, *Survey of clustering algorithms*, IEEE Transactions on Neural Networks, 16(3), pp. 645–678, 2005.