Alan L. Yuille
Song-Chun Zhu
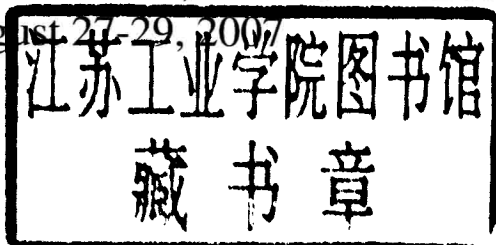Daniel Cremers
Yongtian Wang (Eds.)

Energy Minimization Methods in Computer Vision and Pattern Recognition

6th International Conference, EMMCVPR 2007
Ezhou, China, August 2007
Proceedings

Springer

Alan L. Yuille   Song-Chun Zhu
Daniel Cremers   Yongtian Wang (Eds.)

Energy Minimization Methods in Computer Vision and Pattern Recognition

6th International Conference, EMMCVPR 2007
Ezhou, China, August 27-29, 2007
Proceedings

Springer

Volume Editors

Alan L. Yuille
University of California, Los Angeles, USA
E-mail: yuille@stat.ucla.edu

Song-Chun Zhu
University of California, Los Angles, USA
E-mail: sczhu@stat.ucla.edu

Daniel Cremers
University of Bonn, Germany
E-mail: dcremers@cs.uni-bonn.de

Yongtian Wang
Beijing Insitute of Technology, China
E-mail: wyt@bit.edu.cn

Preface

This volume contains the papers presented at the Sixth International Conference on Energy Minimization Methods on Computer Vision and Pattern Recognition (EMMCVPR 2007), held at the Lotus Hill Institute, Ezhou, Hubei, China, August 27–29, 2007. The motivation for this conference is the realization that many problems in computer vision and pattern recognition can be formulated in terms of probabilistic inference or optimization of energy functions. EMMCVPR 2007 addressed the critical issues of representation, learning, and inference. Important new themes include probabilistic grammars, image parsing, and the use of datasets with ground-truth to act as benchmarks for evaluating algorithms and as a way to train learning algorithms. Other themes include the development of efficient inference algorithms using advanced techniques from statistics, computer science, and applied mathematics.

We received 140 submissions for this workshop. Each paper was reviewed by three committee members. Based on these reviews we selected 22 papers for oral presentation and 15 papers for poster presentation. This book makes no distinction between oral and poster papers. We have organized these papers in seven sections on algorithms, applications, image parsing, image processing, motion, shape, and three-dimensional processing.

Finally, we thank those people who helped make this workshop happen. We acknowledge the Program Committee for their help in reviewing the papers. We are grateful to Kent Shi and Jose Hales-Garcia for their help with organizing Web pages and dealing with the mechanics of collecting reviews, organizing registration, and so no. We also thank Mrs. Wenhau Xia at Lotus Hill Institute for her assistance. We are very grateful to Alfred Hofmann at Springer for agreeing to publish this volume in the *Lecture Notes in Computer Science* series.

June 2007

Alan Yuille
Song-Chun Zhu
Daniel Cremers
Yongtian Wang

Organization

Co-chairs

Alan L. Yuille, University of California, Los Angeles, USA
Song-Chun Zhu, University of California, Los Angeles, USA; Lotus Hill Institute,
    China
Daniel Cremers, University of Bonn, Genmany
Yongtian Wang, Beijing Institute of Technology, China

Program Committee

Serge Belongie, University of California, San Diego, USA
Andrew Blake, Microsoft Research Cambridge, UK
Yuri Boykov, University of Western Ontario, USA
Thomas Brox, University of Bonn, Germany
Joachim Buhmann, Swiss Federal Institute of Technology, Switzerland
Xilin Chen, Institute of Computing Technology, CAS, China
Jim Clark, McGill University, Canada
Laurent Cohen, Universite Paris IX Dauphine, France
Tim Cootes, University of Manchester, UK
Jason Corso, University of California, Los Angeles, USA
James Coughlan, Smith-Kettlewell Eye Research Institute, USA
Frank Dellaert, Georgia Institute of Technology, USA
Sven Dickinson, University of Toronto, Canada
Paolo Favaro, Heriot-Watt University, UK
Mário Figueiredo, Instituto Superior Técnico, Portugal
Daniel Freedman, Rensselaer Polytechnic Institute, USA
Wen Gao,Peking University, China
Feng Han, Sarnoff Corporation, USA
Edwin Hancock, University of York, USA
Martial Hebert, Carnegie Mellon University, USA
Zhiguo Jiang ,Bei-Hang University ,China
Dan Kersten, University of Minnesota, USA
Iasonas Kokkinos, National Tech University of Athens, Greece
Vladimir Kolmogorov, University College London, UK
Fei Fei Li, Princeton University, USA
Stan Li, Institute of Automation, CAS, China
Xiuwen Liu, Florida State University, USA
Yanxi Liu, Carnegie Mellon University, USA
HongJing Lu, University of Hong Kong, China
Petros Maragos, National Technical University of Athens, Greece
Stephen Maybank, Birkbeck College, University of London, UK

Dimitris Metaxas, State University of New Jersey, USA
Marcello Pelillo, Università Ca' Foscari di Venezia, Italy
Anand Rangarajan, University of Florida, USA
Nong Sang, HuaZhong University of Science and Technology, China
Guillermo Sapiro, University of Minnesota, USA
Christoph Schnoerr, University of Mannheim, Germany
Bernard Schoelkopf, MPI for Biological Cybernetics, Germany
Eitan Sharon, University of California, Los Angeles, USA
Jianbo Shi, University of Pennsylvania, USA
Anuj Srivastava, Florida State University, USA
Larry Staib, Yale University, USA
Xiaoou Tang, Microsoft Research Asia,China
Philip Torr, Oxford Brookes University, UK
Alain Trouve, Ecole Normale Supérieure de Cachan, France
Zhuowen Tu, University of California, Los Angeles, USA
Baba Vemuri, University of Florida, USA
Alessandro Verri, Università degli Studi di Genova, Italy
Luminita Vese, University of California, Los Angeles, USA
Yizhou Wang, Palo Alto Research Center, USA
Ying Wu, Northwestern University, USA
Laurent Younes, Johns Hopkins University, USA
Changshui Zhang, Tsinghua University,China
Tianxu Zhang, HuaZhong University of Science and Technology, China

Sponsors

The Lotus Hill Institute
The International Association for Pattern Recognition

Table of Contents

Image Parsing

Image Processing

Motion Analysis

Shape Analysis

Three-Dimensional Processing

An Effective Multi-level Algorithm Based on Simulated Annealing for Bisecting Graph

Lingyu Sun[1] and Ming Leng[2]

[1] Department of Computer Science,
Jinggangshan College, Ji'an, PR China 343009
[2] School of Computer Engineering and Science,
Shanghai University, Shanghai, PR China 200072
sunlingyu@jgsu.edu.cn,lengming@shu.edu.cn

Abstract. Partitioning is a fundamental problem in diverse fields of study such as knowledge discovery, data mining, image segmentation and grouping. The min-cut bipartitioning problem is a fundamental graph partitioning problem and is NP-Complete. In this paper, we present an effective multi-level algorithm based on simulated annealing for bisecting graph. The success of our algorithm relies on exploiting both the simulated annealing procedure and the concept of the graph core. Our experimental evaluations on 18 different graphs show that our algorithm produces encouraging solutions compared with those produced by MeTiS that is a state-of-the-art partitioner in the literature.

1   Introduction

Partitioning is a fundamental problem with extensive applications to many areas using a graph model, including VLSI design [1], knowledge discovery [2], data mining [3],[4], image segmentation and grouping [5],[6]. For example, inspired by spectral graph theory, Shi and Malik [6] formulate visual grouping as a graph partitioning problem. The nodes of the graph are image pixels. The edges between two nodes correspond to the strength with which these two nodes belong to one group. In image segmentation, the weights on the edges of the graph corresponds to how much two pixels agree in brightness, color, etc. Intuitively, the criterion for partitioning the graph will be to minimize the sum of weights of connections across the groups and maximize the sum of weights of connections within the groups. The *min-cut bipartitioning problem* is a fundamental partitioning problem and is NP-Complete [7]. The survey by Alpert and Kahng [1] provides a detailed description and comparison of various such schemes which can be classified as *move-based* approaches, *geometric representations*, *combinatorial* formulations, and *clustering* approaches.

Most existing partitioning algorithms are heuristics in nature and they seek to obtain reasonably good solutions in a reasonable amount of time. Kernighan and Lin (KL) [8] proposed a heuristic algorithm for partitioning graphs. The KL algorithm is an iterative improvement algorithm that consists of making several improvement passes. It starts with an initial bipartitioning and tries to

improve it by every pass. A pass consists of the identification of two subsets of vertices, one from each part such that can lead to an improved partitioning if the vertices in the two subsets switch sides. Fiduccia and Mattheyses (FM) [9] proposed a fast heuristic algorithm for bisecting a weighted graph by introducing the concept of cell *gain* into the KL algorithm. These algorithms belong to the class of *move-based* approaches in which the solution is built iteratively from an initial solution by applying a move or transformation to the current solution. Move-based approaches are the most frequently combined with stochastic hill-descending algorithms such as those based on Tabu Search[10],[11], Genetic Algorithms [12], Neural Networks [13], Ant Colony Optimization[14], Particle Swarm Optimization[15], Swarm Intelligence[16] etc., which allow movements towards solutions worse than the current one in order to escape from local minima.

As the problem sizes reach new levels of complexity, a new class of graph partitioning algorithms have been developed that are based on the multi-level paradigm. The multi-level graph partitioning schemes consist of three phases [17],[18],[19]. The *coarsening phase* is to reduce the size of the graph by collapsing vertex and edge until its size is smaller than a given threshold. The *initial partitioning phase* is to compute initial partition of the coarsest graph. The *uncoarsening phase* is to project successively the partition of the smaller graph back to the next level finer graph while applying an iterative refinement algorithm.

In this paper, we present a multi-level algorithm which integrates a new simulated annealing-based refinement approach and an effective matching-based coarsening scheme. Our work is motivated by the multi-level refined mixed simulated annealing and tabu search algorithm(MLrMSATS) of Gil which can be considered as a hybrid heuristic with additional elements of a tabu search in a simulated annealing algorithm for refining the partitioning in [20] and Karypis who introduces the concept of the graph *core* for coarsening the graph in [19] and supplies **MeTiS** [17], distributed as open source software package for partitioning unstructured graphs. We test our algorithm on 18 graphs that are converted from the hypergraphs of the ISPD98 benchmark suite [21]. Our comparative experiments show that our algorithm produces excellent partitions that are better than those produced by **MeTiS** in a reasonable time.

The rest of the paper is organized as follows. Section 2 provides some definitions and describes the notation used throughout the paper. Section 3 describes the motivation behind our algorithm. Section 4 presents an effective multi-level simulated annealing refinement algorithm. Section 5 experimentally evaluates our algorithm and compares it with **MeTiS**. Finally, Section 6 provides some concluding remarks and indicates the directions for further research.

2   Mathematical Description

A graph $G=(V,E)$ consists of a set of vertices V and a set of edges E such that each edge is a subset of two vertices in V. Throughout this paper, n and m denote the number of vertices and edges respectively. The vertices are numbered from 1 to n and each vertex $v \in V$ has an integer weight $S(v)$. The edges are numbered

from *1* to m and each edge $e \in E$ has an integer weight $W(e)$. A decomposition of a graph V into two disjoint subsets V^1 and V^2, such that $V^1 \cup V^2 = V$ and $V^1 \cap V^2 = \varnothing$, is called a *bipartitioning* of V. Let $S(A) = \sum_{v \in A} S(v)$ denotes the size of a subset $A \subseteq V$. Let ID_v be denoted as v's *internal degree* and is equal to the sum of the edge-weights of the adjacent vertices of v that are in the same side of the partitioning as v, and v's *external degree* denoted by ED_v is equal to the sum of edge-weights of the adjacent vertices of v that are in different sides. The *cut* of a *bipartitioning* $P = \{V^1, V^2\}$ is the sum of weights of edges which contain two vertices in V^1 and V^2 respectively. Naturally, vertex v belongs at the boundary if and only if $ED_v > 0$ and the *cut* of P is also equal to $0.5 \sum_{v \in V} ED_v$.

Given a balance constraint b, the *min-cut bipartitioning problem* seeks a solution $P = \{V^1, V^2\}$ that minimizes *cut(P)* subject to $(1-b)S(V)/2 \leq S(V^1), S(V^2) \leq (1+b)S(V)/2$. A *bipartitioning* is *bisection* if b is as small as possible. The task of minimizing *cut(P)* can be considered as the *objective* and the requirement that solution P will be of the same size can be considered as the *constraint*.

3 Motivation

Simulated annealing belongs to the probabilistic and iterative class of algorithms. It is a combinatorial optimization technique that is analogous to the annealing process used for metals [22]. The metal is heated to a very high temperature, so the atoms gain enough energy to break chemical bonds and become free to move. The metal is then carefully cooled down so that its atoms crystallize into high ordered state. In simulated annealing, the combinatorial optimization cost function is analogous to the energy $E(s)$ of a system in state s which must be minimized to achieve a stable system.

The main idea of simulated annealing is as follows: Starting from an initial configuration, different configurations of the system states are generated at random. A *perturbation* of a system state consists of reconfiguring the system from its current state to a next state within a neighborhood of the solution space. The change in energy cost between the two configurations is determined and used to compute the *probability p* of the system moving from the present state to the next. The *probability p* is given by $\exp(-\frac{\triangle E}{T})$, where $\triangle E$ is the increase in the energy cost and T is the temperature of the system. If $\triangle E$ is negative, then the change in state is always accepted. If not, then a random number r between 0 and 1 is generated and the new state of the system is accepted if $r \leq p$, else the system is returned to its original state. Initially, the temperature is high meaning that a large number of *perturbations* are accepted. The temperature is reduced gradually according to a *cooling schedule*, while allowing the system to reach equilibrium at each temperature through the cooling process.

In [23], Gil proposed the refinement of mixed simulated annealing and tabu search algorithm(RMSATS) that allows the search process to escape from local minima by the simulated annealing procedure, while simultaneously the occurrence of cycles is prevented by a simple tabu search strategy. At each iteration of

RMSATS, the hybrid heuristic strategy is used to obtain a new partitioning $\overline{s}$ in the neighbourhood, $N(s)$, of the current partitioning s through moving vertex v to the other side of the partitioning s. Every feasible partitioning, $\overline{s} \in N(s)$, is evaluated according to the cost function $c(\overline{s})$ to be optimized, thus determining a change in the value of the cost function, $c(\overline{s}) - c(s)$. The problem with local search techniques and hill climbing is that the searching may stop at local optimum. In order to overcome this drawback and reach the global optimum, RMSATS must sometimes accept the worse partitioning to jump out from a local optimum. Therefore, admissible moves are applied to the current partitioning allowing transitions that increase the cost function as in simulated annealing. When a move increasing the cost function is accepted, the reverse move should be forbidden during some iterations in order to avoid cycling, as in tabu search. In [20], Gil presents the MLrMSATS approach that is enhancement of the RMSATS algorithm with the multi-level paradigm and uses the RMSATS algorithm during the *uncoarsening and refinement phase* to improve the quality of the finer graph $G_l(V_l, E_l)$ partitioning $P_{G_l} = \{V_l^1, V_l^2\}$ which is projected from the partitioning $P_{G_{l+1}} = \{V_{l+1}^1, V_{l+1}^2\}$ of the coarser graph $G_{l+1}(V_{l+1}, E_{l+1})$.

In this paper, we present a new multi-level simulated annealing refinement algorithm(MLSAR) that combines the simulated annealing procedure with a boundary refinement policy. It has distinguishing features which are different from the MLrMSATS algorithm. First, MLSAR introduces the conception of *move-direction* to maintain the balance constraint of a new partitioning $\overline{s}$. Second, MLSAR defines $c(\overline{s}) = cut(\overline{s})$ and exploits the concept of *gain* to fast the computation of $c(\overline{s}) - c(s)$ that is computed by $ED(v)\text{-}ID(v)$, where the vertex v is chosen to move to the other side of the partitioning s. MLSAR also uses two buckets with the last-in first-out (LIFO) scheme to fast storage and update the gains of boundary vertices of two sides and facilitate retrieval the highest-gain vertex. Finally, MLSAR doesn't select vertex v to move at random in boundary vertices as in MLrMSATS, but always chooses to move a highest-gain vertex v from the larger side of the partitioning. It is important for simulated annealing to strengthen its effectiveness and achieve significant speedups for high quality solutions with well-designed heuristics and properly move generation strategy.

In [17], Karypis presents the sorted heavy-edge matching (SHEM) algorithm that identifies and collapses together groups of vertices that are highly connected. Firstly, SHEM sorts the vertices of the graph ascendingly based on the *degree* of the vertices. Next, the vertices are visited in this order and SHEM matches the vertex v with unmatched vertex u such that the weight of the edge $W(v, u)$ is maximum over all incident edges. In [19], Amine and Karypis introduce the concept of the graph *core* for coarsening the *power-law* graphs. In [11], Leng and Yu present the core-sorted heavy-edge matching (CSHEM) algorithm that combines the concept of the graph *core* with the SHEM scheme. Firstly, CSHEM sorts the vertices of the graph descendingly based on the *core* number of the vertices by the algorithm in [24]. Next, the vertices are visited in this order and CSHEM matches the vertex v with its unmatched neighboring vertex whose edge-weight is maximum.

In our multi-level algorithm, we adopt the MLSAR algorithm during the *refinement phase* and an effective matching-based coarsening scheme during the *coarsening phase* that uses the CSHEM algorithm on the original graph and the SHEM algorithm on the coarser graphs. The pseudocode of our multi-level algorithm is shown in Algorithm 1.

Algorithm 1 (our multi-level algorithm)

> INPUT: original graph $G(V,E)$
> OUTPUT: the partitioning P_G of graph G
> /*coarsening phase*/
> $l = 0$
> $G_l(V_l,E_l)=G(V,E)$
> $G_{l+1}(V_{l+1},E_{l+1})=\text{CSHEM}(G_l(V_l,E_l))$
> While ($|V_{l+1}| > 20$) do
> $l = l + 1$
> $G_{l+1}(V_{l+1},E_{l+1})=\text{SHEM}(G_l(V_l,E_l))$
> End While
> /*initial partitioning phase*/
> $P_{G_l}=\text{GGGP}(G_l)$
> /*refinement phase*/
> While ($l \geq 1$) do
> $P'_{G_l}=\text{MLSAR}(G_l,P_{G_l})$
>   Project P'_{G_l} to $P_{G_{l-1}}$;
> $l = l - 1$
> End While
> $P_G=\text{MLSAR}(G_l,P_{G_l})$
> Return P_G

4   An Effective Multi-level Simulated Annealing Refinement Algorithm

Informally, the MLSAR algorithm works as follows: At cycle zero, an initialization phase takes place during which the initial partitioning Q is projected from the partitioning $P_{G_{l+1}}$ of the coarser graph G_{l+1}, the Markov chain length L is set to be the number of vertices of the current level graph G_l, the internal and external degrees of all vertices are computed and etc. The main structure of MLSAR consists of a nested loop. The outer loop detects the frozen condition by an appropriate termination criterion whether the current temperature T_k is less than final temperature; the inner loop determines whether a thermal equilibrium at temperature T_k is reached by using the following criterions: The number of attempted moves exceeds L, or the bucket of the start side of the *move-direction* is empty. In the inner loop of the MLSAR algorithm, a neighbor of the current partitioning P is generated by selecting the vertex v with the highest gain from the larger side of the partitioning P and performing the move according to the

following rule: The move is certainly accepted if it improves $cut(P)$, or probabilistically accepted according to a random number uniformly distributed on the interval [0,1]. In the latter case, if the acceptance test is negative then no move is performed, and the current partitioning P is left unchanged. The pseudocode of MLSAR is shown in Algorithm 2. The cycles counter is denoted by k and L represents the Markov chain length. Let $Best$ be the best partitioning seen so far and P be the current partitioning. At cycle k, T_k represents the current temperature and the counter of neighbors sampled is denoted by L_k.

Algorithm 2 (MLSAR)

INPUT: initial bipartitioning Q,balance constraint b,attenuation rate α
         initial temperature T_i,final temperature T_f
OUTPUT: the best partitioning $Best$, cut of the best partitioning $cut(Best)$
MLSAR(
/*Initialization*/
$k = 0$
$T_k = T_i$
Set current parition $P = Q$;
Set the best parition $Best = Q$;
Set Markov chain length $L=|V|$;
For every vertex v in $G = (V, E)$ do
   $ID_v = \sum\limits_{(v,u)\in E \wedge P[v]=P[u]} W(v,u)$
   $ED_v = \sum\limits_{(v,u)\in E \wedge P[v]\neq P[u]} W(v,u)$
   Store v in $boundary\ hash\text{-}table$ if and only if $ED_v > 0$;
End For
/*Main loop*/
While $T_k \geq T_f$ do
  $L_k=1$
  Compute the gains of boundary vertices of two sides;
  Insert the gains of boundary vertices of two sides in buckets respectively;
  While $L_k \leq L$ do
   Decide the $move\text{-}direction$ of the current move;
   If (the bucket of the start side of the $move\text{-}direction$ is empty) then
    Break;
   Else
    Select the vertex v with the highest gain in the bucket;
    Designate the vertex v as tabu status by inserting v in tabu list;
    If ($random(0, 1) \leq min(1, exp(\frac{(ED_v-ID_v)\times|boundary\ hash\text{-}table|}{2\times cut(Q)\times T_k})))$ then
     $L_k=L_k+1$
     Update P by moving the vertex v to the other side;
     original cut Minus its original $gain$ as the cut of new partition P;
     Update the $internal$ and $external\ degrees$ of its neighboring vertices;
     Update the $gains$ of its neighboring vertices in two buckets;
     Update $boundary\ status$ of its neighboring vertices in $boundary\ hash\text{-}table$;

If (the *cut* is minimum and satisfies balance constraint b) then
     Best=P
     Record roll back point;
     Record new *cut* minumum;
   End If /* *cut is minimum*/
  End If /* $r \le p$*/
 End If /* *the bucket is empty*/
End While /* *thermal equilibrium* $L_k \le L$*/
Roll back to minumum *cut* point by undoing all moves and updating the
             internal and *external degrees* and *boundary hash-table*;
Empty the tabu list and two buckets;
$T_{(k+1)} = \alpha \times T_k$
$k = k + 1$
End While /* *frozen criterion* $T_k \ge T_f$*/
Return *Best* and *cut*(*Best*)

The MLSAR algorithm uses a tabu list, which is a short-term memory of moves that are forbidden to execute, to avoid cycling near local optimum and to enable moves towards worse solutions, as in the MLrMSATS algorithm. In the terminology of tabu search [25], the MLSAR strategy is a simple form of tabu restriction without aspiration criterion whose prohibition period is fixed at $|V_l|$. Because the MLSAR algorithm aggressively selects the best admissible vertex based on the tabu restriction, it must examine and compare a number of boundary vertices by the bucket that allows to storage, retrieval and update the gains of vertices very quickly. It is important to obtain the efficiency of MLSAR by using the bucket with the LIFO scheme, as tabu search memory structure. The *internal* and *external degrees* of all vertices, as complementary tabu search memory structures, help MLSAR to facilitate computation of vertex *gain* and judgement of boundary vertex. We also use a *boundary hash-table*, as another complementary tabu search memory structure, to store the boundary vertices whose *external degree* is greater than zero.

During each iteration of MLSAR, the *internal* and *external degrees* and *gains* of all vertices are kept consistent with respect to the current partitioning P. This can be done by updating the *degrees* and *gains* of the vertex v's neighboring vertices. Of course, the *boundary hash-table* might change as the current partitioning P changes. For example, due to a move in an other boundary vertex, a boundary vertex would no longer be such a boundary vertex and should be removed from the *boundary hash-table*. Furthermore, a no-boundary vertex can become such a vertex if it is connected to a boundary vertex which is moved to the other side and should be inserted in the *boundary hash-table*.

5   Experimental Results

We use the 18 graphs in our experiments that are converted from the hypergraphs of the ISPD98 benchmark suite [21] and range from 12,752 to 210,613 vertices.