IMAGE SEGMENTATION BY COOPERATIVE OPTIMIZATION

Xiaofei Huang

CallVista, Inc.,
Foster City, CA 94404, USA
huangxiaofei@ieee.org

ABSTRACT

This paper presents the application of a new cooperative optimization algorithm for image segmentation. In our experiments, it significantly outperforms graph cuts, an emerging powerful optimization algorithm for image processing and computer vision. Compared to graph cuts, it is 10 times faster, much less restrictive on energy function forms, has an error rate two to three times smaller, and does not need extra memory while graph cuts allocated 22 Mbytes more for a 384×288 image. Its operations are simple and fully parallel that can be implemented in a system of agents (e.g., neurons). Also, it has a solid theoretical foundation on its computational properties.

1. INTRODUCTION

Image segmentation is a fundamental problem in image processing and computer vision. Its aim is to partition an image into regions of similar image properties, such as texture, color, and brightness. It has been studied for decades and [1] provides a good overview of different approaches. Recent work includes a variety of techniques, such as stochastic model, morphological watershed based region growing, energy diffision, and graph-theoretical methods.

In particular, the graph-theoretical approaches define an energy function by the similarities of locally extracted image features first and apply different optimization techniques thereafter. Among them are graph cuts [2], normalized graph cuts [3], semidefinite programming [4].

However, those optimization methods become computationally demanding as image size increases. Also, using pairwise similarities has limited success in accuracy and tolerance to noise because they contain only binary constraints as the knowledge for segmentation.

This paper presents a new cooperative optimization algorithm for image segmentation. It has the same principle as the cooperative optimization proposed by D.Marr and T.Poggio [5], but with a different set of difference equations. Unlike [5], the new cooperative optimization has a clearly defined objective function to minimize. Also, it has a solid theoretical foundation on its convergence properties, completeness, equilibrium, quality of solutions, sufficient conditions, and necessary conditions. It is less restrictive on energy function forms than graph cuts and [5] so that higher order constraints can be introduced to improve the accuracy and noise tolerance level. It has also shown promising preliminary results in the applications for stereo matching [6], shape from shading [7], and DNA image analysis [8].

2. ENERGY MINIMIZATION IN IMAGE SEGMENTATION

Like many problems from image processing and computer vision, image segmentation can be formulated as the minimization of an energy function of the following form:

$$E(x_1, x_2, \ldots, x_n) = \sum_i C_i(x_i) + \sum_{i,j,i \neq j} C_{ij}(x_i, x_j) \ , \quad (1)$$

C_i is called a unary constraint on variable x_i and C_{ij} is called a binary constraint on variable x_i and x_j. x_i is a discrete variable of domain D_i, $x_i \in D_i$, of a finite size. Such a problem is called the binary constraint-based optimization. Like many other combinatorial optimization problems, it is NP-hard despite of the simplicity of its form.

The graph cut algorithm puts some restriction on the binary constraint $C_{ij}(x_i, x_j)$. That is, $D_i = D_j$, and

$$C_{ij}(x_1, x_2) = 0 \Leftrightarrow x_1 = x_2 \ ,$$

$$C_{ij}(x_1, x_2) = C_{ij}(x_2, x_1) \geq 0 \ ,$$

for any $x_1, x_2 \in D_i$. An binary constraint satisfying these restrictions is called a semimetric constraint. An interesting special case of that is the Potts model [2]. Also, many existing optimization techniques used for image processing work only with energy functions with constraints of order no higher than two due to the tractability issues. If we want to explicitly keep the discontinuity along object boundaries, we need to drop the semimetric restriction. If we want to obtain better solutions, we need to introduce higher order constraints into energy functions.

0-7803-8554-3/04/$20.00 ©2004 IEEE.

The cooperative optimization algorithm, called CCC algorithm, does not impose those restrictions as long as an energy function is of an aggregate form:

$$E(x_1, x_2, \ldots, x_n) = \sum_i E_i \ , \qquad (2)$$

where E_i is a sub-energy function defined on a subset of variables. Obviously, the energy function (1) is a special case of (2).

As a generalization of the original cooperative optimization, CCC algorithm uses a number of cells (neurons), one for each variable value. The state of the cell corresponding to the value x_i for variable i is denoted as $c_i^{(k)}(x_i)$, where k is the iteration step. The difference equations (termed as "update functions" in [5]) for the states of cells are

$$c_i^{(k)}(x_i) = \sigma_{t^{(k)}} \left(\min_{x_j \in X_i \setminus x_i} \tilde{E}_i \right), \text{for } x_i \in D_i \text{ and } \forall i \ , \quad (3)$$

where $\sigma_{t^{(k)}}$ is a threshold function with the threshold $t^{(k)}$. Any cell of a state value greater than $t^{(k)}$ will be inhibited. $X_i \setminus x_i$ denotes the set X_i minuses $\{x_i\}$. $\tilde{E}_i$ is called modified sub-objective function:

$$\tilde{E}_i = (1 - \lambda_k)\, E_i + \lambda_k \sum_j w_{ij} c_j^{(k-1)}(x_j) \ , \qquad (4)$$

where λ_k is a parameter controlling the level of the cooperation at step k, w_{ij} are non-negative real values and $(w_{ij})_{n\times n}$ should be a propagation matrix (see [7] for detail).

$(w_{ij})_{n\times n}$ defines the neighborhood relations among the variables. The cell for value x_i communicates with the cell for x_j in (3) only if w_{ij} is non-zero. The propagation matrix is a generalization of neighborhood in the special case of the pairwise interactions among cells.

One of the key operations of CCC algorithm is cell inhibition. Inhibiting a cell means discarding the value for a variable the cell corresponds to. Such a process is the same in principle as the one used by the original cooperative optimization. There do exist thresholds for us to discard variable values that can not be in the global optimum of an energy function [7]. If only one value is left for each variable after certain iteration, then the global optimum is found, guaranteed by theory.

3. THEORETICAL FOUNDATIONS

This subsection will show the convergence properties of the algorithm. For other properties, such as the necessary conditions, the sufficient conditions, the completeness of the algorithm please see [7]. Readers who are not interested in the theoretical part can skip this section.

The following theorem shows that $c_i^{(k)}(x_i)$ for $x_i \in D_i$ have a direct relationship to the lower bound on the optimal cost E^*.

Theorem 3.1 *Given any propagation matrix W and the general initial condition $c^{(0)} = 0$ or $\lambda_1 = 0$, $\sum_i c_i^{(k)}(x_i)$ is a lower bound on $E(x_1, \ldots, x_n)$, that is*

$$\sum_i c_i^{(k)}(x_i) \le E(x_1, x_2, \ldots, x_n), \quad \textit{for any } k \ge 1 \ . \quad (5)$$

In particular, let $E_-^{(k)} = \sum c_i^{(k)}(\tilde{x_i})$, then $E_-^{(k)}$ is a lower bound on the optimal cost E^*, that is $E_-^{(k)} \le E^*$.

Here, subscript "-" in $E_-^{(k)}$ indicates that it is a lower bound on E^*.

This theorem tells us that $\sum c_i^{(k)}(\tilde{x_i})$ provides a lower bound on the energy function E. We will show in the next theorem that this lower bound is guaranteed to be improved as the iteration proceeds.

Theorem 3.2 *Given any propagation matrix W, a constant cooperation strength λ, and the general condition $c^{(0)} = 0$, $\{E_-^{(k)} | k \ge 0\}$ is a non-decreasing sequence with upper bound E^*.*

The system is called reaching a consensus solution if, for any i and j, where E_j contains x_i,

$$\arg\min_{x_i} \tilde{E}_i = \arg\min_{x_i} \tilde{E}_j \ .$$

If a consensus solution is found at some step, then we can find out the closeness between the consensus solution and the global optimum in cost. If the algorithm converges to a consensus solution, then it must be the global optimum also. The following theorem makes those points clearer.

Theorem 3.3 *Given any propagation matrix W, and the general initial condition $c^{(0)} = 0$ or $\lambda_1 = 0$. If a consensus solution $\tilde{x}$ is found at iteration step k_1 and remains the same from step k_1 to step k_2, then the closeness between the cost of $\tilde{x}$, $E(\tilde{x})$, and the optimal cost, E^*, satisfies the following two inequalities,*

$$0 \le E(\tilde{x}) - E^* \le \left(\prod_{k=k_1}^{k_2} \lambda_k \right) \left(E(\tilde{x}) - E_-^{(k_1-1)} \right), \quad (6)$$

$$0 \le E(\tilde{x}) - E^* \le \frac{\prod_{k=k_1}^{k_2} \lambda_k}{1 - \prod_{k=k_1}^{k_2} \lambda_k} (E^* - E_-^{(k_1-1)}) \ , \quad (7)$$

where $(E^* - E_-^{(k_1-1)})$ is the difference between the optimal cost E^* and the lower bound on the optimal cost, $E_-^{(k_1-1)}$, obtained at step $k_1 - 1$. When $k_2 - k_1 \to \infty$ and $1 - \lambda_k \ge \epsilon > 0$ for $k_1 \le k \le k_2$, $E(\tilde{x}) \to E^*$.

The performance of CCC Algorithm depends on the dynamic behavior of the difference equations (3). Its convergence properties are revealed in the following two theorems.

The first one shows that, given any propagation matrix and a constant cooperation strength, there does exist a solution to satisfy the difference equations (3). The second one shows that the algorithm converges exponentially to that solution.

Theorem 3.4 *Given any symmetric propagation matrix W and a constant cooperation strength λ, then Difference Equations (3) have one and only one solution, denoted as $(c_i^{(\infty)}(x_i))$ or simply $c^{(\infty)}$.*

Theorem 3.5 *Given any symmetric propagation matrix W and a constant cooperation strength λ, CCC Algorithm, with any choice of the initial condition $c^{(0)}$, converges to $c^{(\infty)}$ with an exponential convergence rate λ. That is*

$$\|c^{(k)} - c^{(\infty)}\|_\infty \leq \lambda^k \|c^{(0)} - c^{(\infty)}\|_\infty . \qquad (8)$$

This theorem is called the convergence theorem. It indicates that CCC Algorithm is stable and has a unique attractor, $c^{(\infty)}$. Hence, the evolution of CCC Algorithm is robust, insensitive to the perturbations, and the final solution of the algorithm is independent of initial conditions. In contrast, conventional algorithms [9] based on iterative improvement, such as Local Search, have many local attractors due to the local minima problem. The evolutions of these algorithms are sensitive to perturbations, and the final solutions of these algorithms are dependent on initial conditions.

4. EXPERIMENTAL RESULTS

Images containing real world objects are used in our experiments. The images are from the evaluation framework offered by Middlebury College [10]. It is originally used for evaluating different stereo matching algorithms. What we need from the framework are the Potts energy model, error calculation, the graph cut algorithm, and the images. One interesting image is from the University of Tsukuba (see Fig. 1).

Like many other image segmentation experiments, we artificially add noise to the original images. The form of noise is the data drop-out noise (commonly referred to as intensity spikes, speckle or salt and pepper noise). In our case, pixels are randomly set with a uniform distribution from the minimum value to the maximum value with a probability p (see Fig. 1).

The parameters we picked for the Potts model are based on the best values for the stereo matching. The unary constraint $C_i(x_i) = 0$ (see (1)) if x_i equals the observed value at site i, otherwise it is set to 10. Only adjacent sites have binary constraints. $C_{ij}(x_i, x_j) = 0$ if $x_i = x_j$, otherwise it is set to 40. In an image, each site has four adjacent sites (neighbors).

With this setting, the result of the graph cut algorithm is shown in Fig. 1. It has 1.87% bad pixels and an error rate of 1.03. The error rate is satisfying, but we can clearly see the image has been over flattened and the camera's post and the lamp's post are almost completely merged into the background.

In our algorithm, the unary constraint $C_i(x_i)$ is the same as the one used by graph cuts. Instead of using the binary constraint like the graph cuts due to its limitation, we choose 9-ary constraint, denoted as $C^9(x_{i1}, \ldots, x_{i9})$, defined by a 3×3 mask. That is, given a site i and its eight neighbors, the constraint is defined on the nine corresponding variables. If a site j is of the same value as the one of the central site in the mask, then the value at site j is set to 1. Otherwise, it is set to 0. Therefore, there are in total $2^9 = 512$ tuples defined by the constraint. We rank those tuples based on their occurring probabilities through a learning process using the four images, pick the top ones that have a probability of 99.8% in total, and set $C^9(x_{i1}, \ldots, x_{i9}) = 0$, for those tuples, and a very high value for other tuples.

Since a 3×3 mask is used, for the propagation matrix, we set $w_{ij} = 1/9$ for x_i and its eight neighbor variables, and $w_{ij} = 0$ for the other variables. In the iterative function (3), the parameter λ_k is updated as

$$\lambda_k = (k-1)/k, \quad \text{where } k \geq 1.$$

With this setting, the result of our algorithm is shown in Fig. 1. It has 0.63% bad pixels, which is three times smaller than that of graph cuts, and an error rate of 0.49, which is more than two times smaller than that of graph cuts. Our algorithm did not allocate any extra memory besides the 15Mbytes used by the framework, while graph cuts allocated 22 Mbytes more. Graph cuts took 278 sec. on an XP machine of Pentium $1.4G$, while our algorithm took only 33 sec.

Table 1 shows the comparison of our algorithm versus graph cuts using the Tsukuba image with drop-out noise of different probabilities p. Table 2 shows the performance of the two algorithms using the Sawtooth image of size 434×380 from the framework. Clearly, our algorithm significantly outperforms graph cuts both in solution quality and speed in all cases.

5. CONCLUSIONS

CCC Algorithm is a general global optimization algorithm. It is applicable to the special case of the binary smoothness constraints [10], as well as to other cases where constraints are of higher orders and are non-smooth. In our experiments, it significantly outperformed graph cuts (a popular powerful optimization algorithm in image processing and

Table 1. Performance on the Tsububa image

	Our algorithm / Graph cuts		
Noise	Error rate	Bad pixels	Time (in sec.)
25%	0.49 / 1.03	0.63% / 1.87%	33 / 278
27%	0.52 / 1.05	0.70% / 1.97%	33 / 279
29%	0.54 / 1.05	0.85% / 1.97%	34 / 364
31%	0.61 / 1.04	0.97% / 1.97%	34 / 367
33%	0.64 / 1.06	1.15% / 2.05%	33 / 375

Table 2. Performance on the Sawtooth image

	Our algorithm / Graph cuts		
Noise	Error rate	Bad pixels	Time (in sec.)
25%	0.66 / 0.81	0.28% / 0.65%	37 / 619
27%	0.66 / 0.85	0.31% / 0.96%	37 / 355
29%	0.72 / 0.84	0.42% / 0.90%	37 / 347
31%	0.74 / 0.88	0.50% / 1.03%	37 / 352
33%	0.81 / 0.87	0.65% / 1.00%	37 / 432

computer vision) in terms of solution quality and speed. Its operations are simple and inherently parallel. They can be directly implemented in systems of locally connected neurons or processors, where their spatio-temporal dynamics is described by the difference equations in the algorithm.

Our algorithm has a solid theoretical foundation. It is guaranteed to converge to an unique equilibrium, which must be the global optimum if it is a consensus one. The convergence is exponential, and the process is insensitive to perturbations to the initial conditions and intermediate solutions. It has both sufficient conditions for recognizing global optima and necessary conditions for reducing the search space. A lower bound on the energy function is also provided by the algorithm which is guaranteed to be improved after each iteration. All these properties make this algorithm attractive and remarkably different from classic algorithms for combinatorial global optimization.

Unlike the original cooperative optimization [5], the one proposed in this paper has a clearly defined objective function (2). Its difference equations (3) are also substantially different from the original ones. The inhibition process used by both of the cooperative optimization algorithms makes them fundamentally different from the most known optimization methods. As Steven Pinker pointed out in his book "How the Mind Works", the cooperative optimization captures the flavor of the brain's computation of stereo vision. Further investigation on this new optimization method is desirable both from the theoretical perspective for understanding human vision and from the practical perspective for a wide range of applications in image processing, pattern recognition, and computer vision.

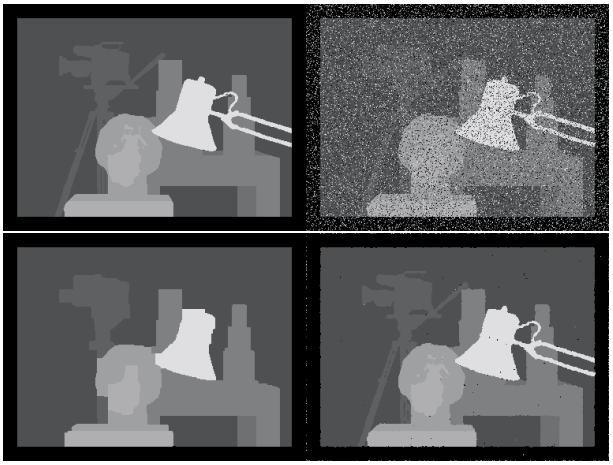

Fig. 1. Original image (top-left), Noisy image (top-right), Graph cuts (bottom-left), and Cooperative algorithm (bottom-right).

6. REFERENCES

[1] R.M. Haralick and L.G. Shapiro, "Image segmentation techniques," *CVGIP*, vol. 29, no. 1, pp. 100–132, January 1985.

[2] Y. Boykov, O. Veksler, and R. Zabih, "Fast approximate energy minimization via graph cut," *IEEE TPAMI*, vol. 23, no. 11, pp. 1222–1239, 2001.

[3] J. Shi and J. Malik, "Normalized cut and image segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1997.

[4] J. Keuchel, C. Schnoörr, C. Schellewald, and D. Cremers, "Unsupervised image partitioning with semidefinite programming," in *Pattern Recognition (24th DAGM Symposium, Zürich),Springer LNCS*, Luc Van Gool, Ed., 2002, vol. 2449 of *Springer LNCS*, pp. 141–149.

[5] D. Marr and T. Poggio, "Cooperative computation of stereo disparity," *Science*, vol. 194, pp. 209–236, June 1976.

[6] X. Huang, "A general global optimization algorithm for energy minimization from stereo matching," in *ACCV*, Korea, 2004.

[7] Xiaofei Huang, "A general framework for constructing cooperative global optimization algorithms," *4th International Conference on Frontiers in Global Optimization*, June 2003.

[8] Xiaofei Huang, "Cooperative optimization for solving large scale combinatorial problems," in *4th International Conference on Cooperative Control and Optimization*, Don Grundel, Robert Murphey, and Panos Pardalos, Eds., Destin, Florida, U.S.A., November 2003.

[9] Z. Michalewicz and D. Fogel, *How to Solve It: Modern Heuristics*, Springer-Verlag, New York, 2002.

[10] D. Scharstein and R. Szeliski, "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms," *IJCV*, vol. 47, pp. 7–42, April-June 2002.