

Dynamic programming versus graph cut algorithms for fitting non-parametric models to image data

C.A. Glasbey¹

¹ Biomathematics & Statistics Scotland, King's Buildings, Edinburgh EH9 3JZ, Scotland

Abstract: Dynamic programming and graph cut algorithms can, in some cases, find globally best fits of non-parametric models to image data, for restoration, segmentation and template matching. In this paper we compare conditions and results for the two methods, illustrated by restoration of a synthetic aperture radar (SAR) image.

Keywords: Image Restoration; Maximum a Posteriori Estimator; Markov Random Field; Penalised Likelihood; Synthetic Aperture Radar.

1 Introduction

Non-parametric models are fit to image data for many reasons, including restoration, segmentation and template matching. For example, synthetic aperture radar (SAR) is a form of remote sensing in which data have a large noise component due to speckle (see Figure 1(a)), and therefore need smoothing, or restoring, before they can be interpreted. In many cases, the problem of fitting a non-parametric model to image data (y) can be formulated, using either a Bayesian or penalised likelihood framework, as

$$\hat{\beta} = \arg \min_{\beta \in \mathcal{B}} \left\{ \sum_i f(y, \beta_i) + \lambda \sum_{i \sim j} g(\beta_i - \beta_j) \right\}. \quad (1)$$

Here, in full generality, β is an array of B -dimensional vectors indexed by i , an I -dimensional integer, f is a measure of model fit to the data, derived from the negative log-likelihood of y , and g is either an empirical term penalising lack of smoothness in β , or the log-prior of β , a Markov Random Field (MRF) with first-order neighbourhood specified by ' $i \sim j$ ', meaning $\sum_{k=1}^I |i_k - j_k| = 1$. The set \mathcal{B} constrains $\beta_i \in \{1, 2, \dots, n\}^B$, though this need not be overly restrictive as any desired level of resolution can be achieved by setting n sufficiently large. For image restoration, $f = (y_i - \beta_i)^2$, whereas for warping, or matching, to template μ , $f = (y_i - \mu_{\beta_i})^2$, and for

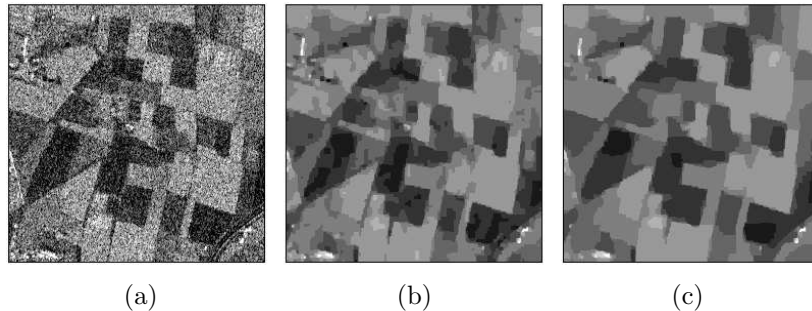


FIGURE 1. Synthetic aperture radar (SAR) image: (a) log-transformed data; (b) restoration using iterated dynamic programming (IDP); (c) restoration using graph cut algorithm.

segmentation $f = y_{\beta_i}^*$, where y^* denotes transformed data (Glasbey and Young, 2002). The penalty term may be convex, such as $g(z) = z^2$ or $|z|$, or non-convex, such as the indicator function, and λ is a non-negative constant whose magnitude determines the smoothness of the fit.

Solving (1) is computationally challenging, both because image data sets are large and because objective functions are prone to local optima. However, if certain conditions apply, dynamic programming and graph cut algorithms can be used, and they are among that rare class of optimisation algorithms, those that are both fast and global!

2 Algorithms

Dynamic programming (DP) is an elegant method for finding the global solution to (1), but only if $I = 1$, using a recursive algorithm. Cases include 1-D image warping, also termed dynamic time warping, and finding 1-D boundaries to segment 2-D images, by finding a path between opposite sides of an image, with β_i specifying the row location of the boundary in column i (Glasbey and Young, 2002). Glasbey (2009) proposed generalisations of DP to solve higher dimensional problems, but without the guarantee of global optimality, the simplest being iterated dynamic programming (IDP).

DP cannot be used to restore the SAR image, because $I = 2$, but IDP can be used to find a local solution to (1) with $f = (y_i - \beta_i)^2$, and $g(z) = |z|$ chosen to impose smoothness on the restoration while tolerating the step changes we expect at agricultural field boundaries. Note that, although the objective function is convex, this is a pathological case for which gradient descent algorithms are unsuccessful (Kunsch, 1994), leading Kovac and Smith (2009) to develop a solution using a taut-string type algorithm. The data, y , are a $P = 250 \times 250$ array of log-transformed SAR values, rescaled

to the range 0 and 255. To simplify computation, we limit β to a coarse set of values $\{65, 75, \dots, 165\}$, and set $\lambda = 50$, chosen by eye to produce a realistic restoration. IDP was initialised by applying DP separately to each image column, and subsequently DP was applied alternately to all rows and columns, taking into account neighbouring values of β . After 26 iterations, which took 1.2sec of time on a single core of a 3.2Ghz AMD Opteron processor, β converged to an approximation to the maximum *a posteriori* (MAP) or maximum penalised likelihood estimator shown in Figure 1(b), with a minimised value of the objective function of 388*P*.

Graph cut (GC) algorithms can also be used to find the global solution to (1), provided $B = 1$ and g is a convex function; conditions which SAR image restoration satisfies. GC reformulates the optimisation problem as finding the maximum flow through a network from a source to a sink, or equivalently, the minimum-cost subset of edges which disconnect the source from the sink. Grieg et al (1989) were the first to use GC in image analysis, to restore binary images. More recently, Boykov and co-workers (see, for example, Boykov and Kolmogorov, 2004) have extended GC to a broader class of image problems.

We create an array of $P \times (n + 1)$ nodes, indexed by (i, β) , with edges and directional flow capacities specified by:

$$\begin{aligned} C\{\text{source} \rightarrow (i, 1)\} &= \infty \\ C\{(i, n + 1) \rightarrow \text{sink}\} &= \infty \\ C\{(i, \beta) \rightarrow (i, \beta + 1)\} &= f(y, \beta) \\ C\{(i, \beta) \rightarrow (j, \beta) | i \sim j\} &= \lambda g(1) \\ C\{(i, \beta) \rightarrow (j, \beta - k) | i \sim j\} &= \lambda \{g(k + 1) - 2g(k) + g(k - 1)\} \quad \text{for } k \geq 1 \end{aligned}$$

For SAR restoration, $g(z) = |z|$, and the last, large set of edges is not needed, though there are still 750,000 nodes and 4.6 million edges. Boykov's implementation of the GC algorithm (available at <http://vision.csd.uwo.ca/code/>) took 1.1sec of CPU time to find the globally optimal result. However, there are many alternative GC algorithms, and speed can vary enormously. For example, another public domain implementation, NETFLO (Nijenhuis and Wilf, 1978), took 10^3 more time, 21 minutes to reach the same solution. The MAP or maximum penalised likelihood estimator is shown in Figure 1(c), with a minimised value of the objective function of 374*P*. We see that the agricultural fields are more clearly identified than in Figure 1(b).

3 Discussion

DP and GC can each fit non-parametric models to image data by finding the global solution to (1) provided certain conditions apply. For DP, the condition is $I = 1$, whereas for GC, $B = 1$ and g must be convex. As we have seen, GC can restore 2-D images, or any higher dimension, for

some choices of g , whereas DP can only restore 1-D images, and beyond this we have to resort to approximations such as IDP, though they have the advantage of permitting non-convex g . We have also compared IDP and GC in finding 2-D surfaces to segment 3-D images ($B = 1, I = 2$), which DP cannot achieve. However, if instead the model required us to find a 1-D path across a 3-D image ($B = 2, I = 1$) then DP could do it but GC could not. DP and GC can both be used in 1-D warping ($B = 1, I = 1$), but only GC can perform so-called $1\frac{1}{2}$ -D warping, which occurs when matching stereo image pairs, where no warping is needed between rows ($B = 1, I = 2$). However, neither DP nor GC can perform 2-D or 3-D warping, though DP seems to be more amenable to generalisations such as IDP. Overall, GC is a much more complicated algorithm than DP, and its many variants can have an enormous range of performance speeds. DP is more flexible, being generalisable to MRFs with higher-order neighbourhoods, and to find local optima using IDP even when DP cannot be used.

Acknowledgments: I am grateful to Alec Mann for assistance in using Boykov's C++ algorithm. The work was funded by the Scottish Government.

References

- Boykov, Y., and Kolmogorov, V. (2004). An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **26**, 1124-1137.
- Glasbey, C.A. (2009). Two-dimensional generalisations of dynamic programming for image analysis. *Statistics and Computing*, **19**, 49-56.
- Glasbey, C.A., and Young, M.J. (2002). Maximum *a posteriori* estimation of image boundaries by dynamic programming. *Applied Statistics*, **51**, 209-221.
- Greig, D.M., Porteous, B.T., and Seheult, A.H. (1989). Exact maximum *a posteriori* estimation for binary images. *Journal of the Royal Statistical Society, Series B*, **51**, 271-279.
- Kovac, A., and Smith, A.D.A.C. (2009). Regression on a graph. (Available on arXiv, at <http://arxiv.org/abs/0911.1928>)
- Kunsch, H.R. (1994). Robust priors for smoothing and image restoration. *Annals of the Institute of Statistical Mathematics*, **46**, 1-19.
- Nijenhuis, A., and Wilf, H.S. (1978). *Combinatorial Algorithms* (2nd edition). London: Academic Press