

# Multilevel Spectral Partitioning for Efficient Image Segmentation and Tracking

David Tolliver

Robert T. Collins

Simon Baker

Robotics Institute, Carnegie Mellon University  
Pittsburgh, PA 15213

## Abstract

*An efficient multilevel method for solving normalized cut image segmentation problems is presented. The method uses the lattice geometry of images to define a set of coarsened graph partitioning problems. This problem hierarchy provides a framework for rapidly estimating the eigenvectors of normalized graph Laplacians. Within this framework, a coarse solution obtained with a standard eigensolver is propagated to increasingly fine problem instances and refined using subspace iterations. Results are presented for image segmentation and tracking problems. The computational cost of the multilevel method is an order of magnitude lower than current sampling techniques and results in more stable image segmentations.*

## 1 Introduction

The normalized cut (NCut), proposed by Shi and Malik [11], provides a rigorous computational foundation for image segmentation problems. A graph is constructed from the image such that the pixels constitute the vertex set, connected by weighted edges representing similarity between nearby pixels. This formulation allows the image segmentation problem to be treated as a graph partition of the vertex set. The partition measure associated with the NCut seeks to simultaneously minimize the cross segment connectivity and maximize the within segment association. The normalized cut criterion effectively suppresses the spuriously small segmentations that arise in minimum cut formulations.

The authors of [11] develop a spectral relaxation of the normalized cut criterion to optimize their NP-hard objective function. This relaxation provides a bounded quality cut with respect to the original discrete problem. Relaxations of this form were further extended by Yu and Shi in [13, 14], who allow constraints to be included in the cut function and derive multiway cuts. While powerful, computational cost remains a major obstacle to the integration of these techniques into vision systems.

In this paper we present a technique that reduces the computational cost while yielding the exact continuous optimal solution. For image segmentation the method exploits

the lattice structure of the pixel coordinates to define a multilevel approach to computing the eigenstructure of graph Laplacians. A coarse solution is obtained using a standard eigensolver and propagated through increasingly refined problem instances. At each level the projected solution is fit to the current problem by *subspace iterations*<sup>1</sup> and then projected onto a further refined problem instance. The resulting solution hierarchy defined by interpolation and refinement rules provides a good initial approximation to the eigenvectors of the Laplacian of the full graph.

Our multilevel method results in orders of magnitude speed improvements over the Lanczos algorithm (*see [4]*), for eigenspace computation, used by Shi and Malik in [11]. The solutions obtained are shown to be superior to approximation techniques in [13, 5] that sample the graph representation to reduce computational cost of the eigenstructure computation. Unlike approximation techniques, including the algebraic multigrid approach of Sharon *et al.*[10], our multilevel method maintains the quality bound given for spectral relaxations. We present image segmentation and tracking results. These results are obtained orders of magnitude faster than the full eigenproblem posed in [11], and are more stable than those obtained by samplings techniques. We compare our technique with the sampling heuristic employed by Yu and Shi[13] and the Nyström approximation of Fowlkes *et al.*[5]. We also demonstrate how the multilevel graph structure regularizes the problem, alleviating the sensitivity to parameters such as neighborhood density and scale.

## 2 Graph Partitioning

To pose image segmentation as a graph partitioning problem, we must construct a graph from the image data. This is done by assigning a vertex in the graph to each pixel in the image. The connectivity of the graph is defined by spatial neighborhoods in the image plane. For any two neighboring pixels an edge weight is assigned in proportion to the level of feature agreement between the pixels. Typical pairwise pixel features include color similarity, texture descriptions,

<sup>1</sup>Subspace Iterations [4] are a generalization of the power-method, for estimating the dominant eigenvector, to invariant subspace estimation.

and the magnitude of edges along the path between two pixels. Given such a graph  $G = (V, E)$  with a vertex set  $V$  and edge set  $E$  a cut separates the graph into disjoint subsets such that  $V \supseteq \bigcup V_i$  and  $\forall(i, j) V_i \cap V_j = \emptyset$ .

There are many *cut* criteria, such as the minimum, maximum, average, and normalized cut [11]. The normalized cut is a desirable measure as it seeks to partition the graph into strongly connected subgraphs that are weakly connected to each other. The normalized cut criterion is:

$$\underset{V_1, \dots, V_p}{\operatorname{argmin}} : \frac{1}{p} \sum_{i=1}^p \frac{|E(V_i, V \setminus V_i)|}{\operatorname{vol}(V_i)} \quad (1)$$

where  $\operatorname{vol}(V_i)$  is the sum of edge weights associated with the vertices in  $V_i$ , and  $|E(V_i, V \setminus V_i)|$  is the sum of the edge weights connecting  $V_i$  to remainder of the graph.

In addition to the *cut* criterion, or objective function, the formulation can be generalized to include balance and co-membership constraints. The problem of obtaining the optimal cut with respect to Equation 1 is NP-hard in general. One family of tractable techniques for computing a bounded quality normalized cut of  $G$  is the spectral relaxation.

## 2.1 The Spectral Relaxation

The normalized cut graph partitioning criterion can be expressed as a quadratic form on the matrix representation of the graph. The combinatorial problem formulation leads to an NP-hard quadratic program for the optimal p-way cut [13], expressed as:

$$\underset{Z}{\min} : \frac{1}{p} \operatorname{tr} \frac{Z^T(D - W)Z}{Z^T D Z} \quad (2)$$

$$\text{s.t.} : Z_{ij} \in \{0, 1\} \quad (3)$$

$$: Z^T Z = \operatorname{diag}(|V_1|, \dots, |V_p|) \quad (4)$$

where  $W$  is the weight matrix associated with  $G$  and  $d_{ii}$  is the  $i^{\text{th}}$  entry of the diagonal mass matrix of  $W$ , defined as  $D(i, i) = \sum_{j=1}^n W(i, j) = \operatorname{vol}(v_i)$ . The constraints in Equation 3 and 4 insure that the partition matrix  $Z$  is binary and that the partitions are disjoint. The relaxation to an eigenstructure problem is obtained by removing the discrete constraint,  $Z_{i,j} \in \{0, 1\}$  on the entries of  $Z$ , while preserving the orthogonality constraint implicit in Equation 4 (i.e. that  $Z^T Z = I_p$ ). It is clear that Rayleigh Quotient expressed in Equation 2 subject only to the orthogonality constraint is exactly a generalized eigenproblem in  $Z$ .

The eigen-relaxation of Equation 2 can be further simplified by working with the symmetrized version of the generalized eigenproblem. The symmetric eigenproblem equivalent to Equation 2 is defined as:

$$\underset{X}{\min} : \operatorname{tr} (X^T X)^{-1} X^T \mathcal{L}(W) X \quad (5)$$

where  $\mathcal{L}(W) \doteq D^{-1/2}(D - W)D^{-1/2}$ . The solution to Equation 5 can be related exactly to the relaxed solution to the generalized eigenproblem as  $Z = D^{-1/2}X$ . This relaxation demonstrates, in concert with Ky Fan's dominance theorem [6], that the minimizer of the relaxed instance of Equation 5 is in fact the  $p$ -dimensional subspace  $X$  corresponding to the  $p$  smallest eigenvalues of  $\mathcal{L}(W)$ .

The continuous solution  $X$  to Equation 5 must be rounded to obtain a discrete solution that induces a cut in  $G$ . Several methods have been proposed for obtaining a discrete solution from the eigenvectors. The embedding methods treat the eigenvectors as defining a  $p$ -dimensional coordinate for each vertex in the graph. These points are then clustered using k-means or a similar technique. In Chan *et al.*[3], the clustering is performed on the angles between data points, based on the observation that the objective value of Equation 5 is invariant to change of basis in  $X$ , and therefore only the relative angles should be employed in comparing points. Ng *et al.*[9] made a similar observation, and recommended projecting the embedded points onto the sphere to satisfy the invariance. This projection allowed k-means clustering to be used directly on the projected coordinates. Yu and Shi [13] use the projection recommended in [9] but iteratively align the points with the positive orthant of the  $p$ -sphere. Unlike Ng, they show that the first eigenvector of  $\mathcal{L}(W)$ ,  $\hat{x}_1 = D^{1/2}\hat{1}$ , should be used in discretization step. The spectral partition community has suggested several other approaches, such as recursive 2-way partitioning. We employ the method in [13] to obtain our results. This decision was motivated by the empirical consistency of the solutions obtained across multiple initializations.

Several approximate normalized cuts (NCuts) techniques have been proposed to overcome the computational cost incurred in diagonalizing large matrices. The Nyström approximation for NCuts proposed by Fowlkes *et al.* [5] randomly samples the graph and then projects the subsampled solution onto the original problem. This can be interpreted as a very simple multilevel technique with uncertain convergence. Sharon *et al.* [10] proposed an approximation of the normalized cut, solved using Algebraic Multigrid (AMG). While both of these methods provide great computational gains neither method is solving the exact normalized cut or a complete bounded relaxation. Our proposed method, similar to Barnard [2] in the domain decomposition literature, solves the exact spectral relaxation of the NCuts problem. This is accomplished by efficiently solving the eigenvector computation required by Equation 2. This has the advantage of maintaining the bound on solution quality derived from the relaxation.

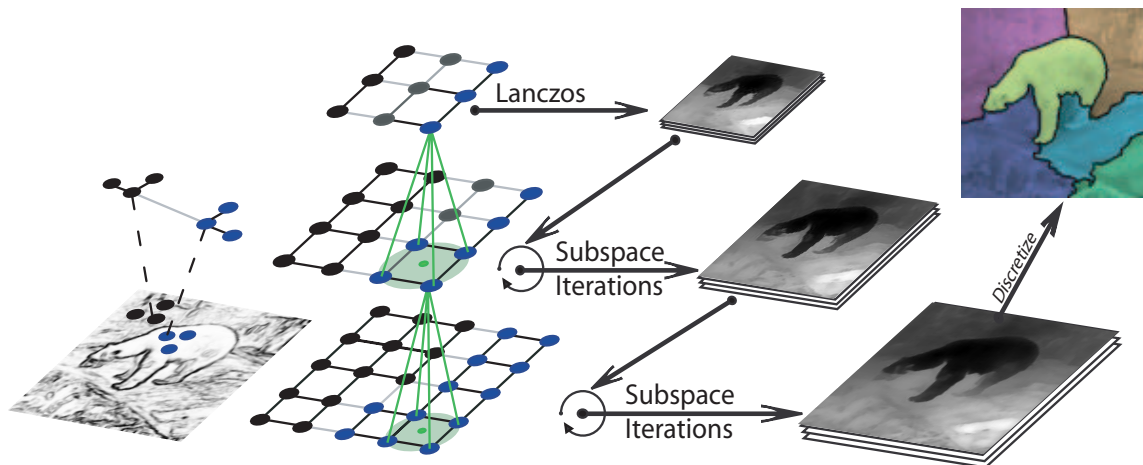


Figure 1: A schematic of the multilevel method. The pairwise similarity graph  $G_0$  is computed using the feature image at the original resolution. The eigenproblem  $Prob_0$  is coarsened  $n$  times using a kernel on the image lattice geometry; this results in the problem hierarchy  $Prob_0 \succ \dots \succ Prob_{n-1}$ . The Lanczos algorithm is used to estimate the eigenvectors for  $Prob_{n-1}$ , this solution is then refined and used to initialize the *subspace iterations*[4]. These refinement and fitting operations step solutions down the hierarchy until a solution at the original problem fidelity is obtained. The continuous solution is then discretized using the alignment method proposed in [13].

### 3 The Multilevel Method

The relaxation of Equation 2 to a continuous-valued problem allows the solution to the NP-hard optimal cut problem to be approximated as an eigenstructure problem. We address the efficient computation of this class of relaxations, the spectral relaxation, by developing a multilevel technique for computing the eigenvectors of graph Laplacians. Intuitively, multilevel methods are effective for mixing algorithms (such as the power method) as they operate on a series of related graphs with increasingly reduced diameter. A schematic of the proposed multilevel method can be found in Figure 1.

Based on Teng's framework [12] for specifying multilevel methods, we decompose our technique into three distinct classes of operations: *coarsening*, *refinement*, and *fitting*. The proposed method is situated in this framework as follows:

- 1 *Coarsening Operation*: the geometric interpolation rule is employed recursively to construct  $n-1$  reduced problems  $Prob_0 \succ Prob_1 \succ \dots \succ Prob_{n-1}$ .
- 2 *Refinement Operation*: maps the solution  $\Pi_k$  to an estimate  $\Pi_{k-1}^*$  for the refined problem  $Prob_{k-1}$  by applying the transpose of the interpolation rule to the solution  $\Pi_k$ .
- 3 *Fitting Operation*: improves the estimate of the eigenstructure  $\Pi_{k-1}^* = (\Lambda_k^*, X_k^*)$  through an iterative estimation technique such as *subspace iteration*[4].

#### 3.1 Coarsening with Geometric Graph Kernels

To map weights from a graph  $G$  to a coarsened graph  $G'$  we define a  $|V'| \times |V|$  interpolation matrix  $T$  using a kernel on the geometry of  $V$  and  $V'$  such that  $T(u, v) = K(x_u, x_v) : u \in V', v \in V$ , where  $x_v$  denotes the coordinate assigned to the vertex  $v$ . To insure a well behaved matrix  $T$ , we place the following conditions on the kernel  $K$ :  $K(x_u, x_v) \geq 0$  and  $\sum_{v \in V} K(x_u, x_v)^2 = 1$ . These conditions restrict  $T^T T, T T^T$  to the set of doubly stochastic matrices. Interpolations of form  $T$  preserve the normalization central to NCuts as  $T^T T$  and  $T T^T$  are both doubly stochastic.

We define a bilinear lattice kernel (BLK) on the coordinates of image pixels to determine the *coarsening* and *refinement* operations. The entries in the sparse bilinear interpolation matrix  $B$  are determined as follows:

$$B(u, v) = \begin{cases} .5 & : x_v \in N_4(u, V) \\ 0 & : otherwise \end{cases} \quad (6)$$

where  $N_4$  is the  $2 \times 2$  refined lattice neighborhood centered at the coordinate  $x_u$ . The vertices of  $V'$  were positioned between every 4 vertices in  $V$ . This results in a coarsening of vertex set  $V$  by approximately 1/4 to the vertex set  $V'$ . It is clear from the construction of the kernel  $B$  that the operator satisfies our kernel criteria placed on the interpolation matrices  $T$ . Other interpolation matrices of this form may be applied to the NCuts matrix as long as the kernel conditions

are satisfied.

### 3.2 Refinement: the Transpose of a Geometric Graph Kernel

The refinement operation is, in general, the transpose of the interpolation matrix  $T$  used to coarsen the problem. The BLK, in Equation 6, is perhaps the simplest effective kernel for coarsening and refining the graph. Solution refinement with  $B^T$  duplicates the eigenvector valuation  $X_k$  for each vertex over its four ancestors in  $Prob_{k-1}$ . This operation is performed for each vertex in  $Prob_k$ , generating an approximate eigenvector for  $Prob_{k-1}$ . Conveniently, BLK refinement preserves orthogonality (as  $X_{k-1}^T X_k = X_k^T B B^T X_k = X_k^T X_k = I$ ).

To mitigate the excessive smoothing accompanying the BLK (Equation 6), we add an empirical diffusion to the refinement operation. The empirical smoother for refining the solution for  $k$  to an estimate for  $k-1$  is defined as:

$$E = D_{k-1}^{-1} W_{k-1} B^T \quad (7)$$

where  $D_k$  and  $W_k$  are the mass and weight matrices derived from  $G$ . This preserves more of the structure of the local weighting associated with each vertex.

### 3.3 Fitting: Iterative Subspace Estimation

The solution  $\Pi_k$  is refined to generate  $\Pi_{k-1}^*$  for problem  $Prob_{k-1}$ , and is improved using an iterative invariant subspace estimation technique. In this work we employ *subspace iterations* [4].

Subspace iterations generalize the power method, which is an algorithm for estimating the eigenvector paired with largest eigenvalue of a matrix  $A$ . The algorithm operates by iterating the following two steps until convergence. If  $X_{(p)}$  is the current estimate of the solution (a set of  $p$  orthonormal eigenvectors):

$$Z_p \leftarrow A X_{(p)} \quad (8)$$

$$X_p R \leftarrow Z_p \text{ (Q.R. decomposition) } \quad (9)$$

Intuitively, the algorithm operates as follows. The current estimate of the  $X_{(p)}$  can be split into two components: (1) the component in the subspace spanned by the  $p$  largest eigenvectors, (2) the component in the orthogonal complement. Each time step 1 (Equation 8) is applied the component in the subspace spanned by the  $p$  largest eigenvectors is multiplied by at least  $\lambda_p$  and the other component by less than  $\lambda_{p+1}$ . Hence after the re-orthonormalization the component in the orthogonal complement is reduced by at least  $\lambda_{p+1}/\lambda_p$  in each iteration. In the limit the orthogonal complement disappears and all that is left is the component in the subspace spanned by the  $p$  largest eigenvectors. The

only requirement for convergence is that the initial estimate has a non-zero component in each direction in the target subspace. This occurs with probability 1 for a random initialization, although there are pathological counter examples.

To make  $\mathcal{L}$  suitable for power-type methods, we must transform its eigenvalues. The maximization form denoted  $\mathcal{L}_+$  associates the target subspace with the largest eigenvalues of  $\mathcal{L}_+$  so that the *subspace iteration* will converge to the desired subspace of  $\mathcal{L}$ . This is accomplished by changing the sign on the eigenvalues of  $\mathcal{L}$  and adding 1. Recall that the matrix form of Equation 5 is  $\mathcal{L} = D^{-1/2}(D-W)D^{-1/2}$  with eigenvalues in  $[0, 2]$ , and note that this equivalent to  $I - D^{-1/2}W D^{-1/2}$ . By dropping the subtraction and adding  $I$  we arrive at an appropriate matrix for maximization  $\mathcal{L}_+ = D^{-1/2}W D^{-1/2} + I$ . As the eigenvectors that minimize  $trace X^T \mathcal{L}(W)X$  maximize  $trace X^T \mathcal{L}_+(W)X$ , therefore we can use *subspace iterations* to compute the appropriate eigenvectors for partitioning.

Finally, the normalization used to compute the Laplacian differs from the Markov chain matrix  $P$ . As  $\mathcal{L}_+$  involves an inhomogenous normalization function dictating that the coarsening operation must be computed as  $\mathcal{L}_k = \mathcal{L}_+(T W_{k-1} T^T)$  rather than  $\mathcal{L}_k = T \mathcal{L}_{k-1} T^T$ .

## 4 Image Segmentation Results

We demonstrate our technique on images of size  $300 \times 300$  pixels. The results, shown in Figure 2, are compared with segmentations generated by the methods of Yu and Shi [13] and Fowlkes *et al.* [5]. To facilitate comparison, overlapping parameters such as feature significance and the number of segments, between the methods were fixed. In the case of [13, 5] the method specific parameters such as the number of samples and the spatial neighborhood size were set to values published by the authors. For the multilevel method we used a 4 level problem hierarchy with the BLK for the coarsening operator and the smoother from Equation 7 for the refinement operator.

As in [13] we use the *intervening contour* (IC) from Malik *et al* [7] to assign weights between neighboring pixels (vertices). The IC specifies that the edge weight between connected pixels is inversely proportional to largest edge intensity along the line connecting the pixels. We used  $\sigma_{IC} = .15$ , as the significance of the IC based on the parameters given in Yu and Shi [13]. Figure 2 column 2 displays the filtered images used to compute the graph edge weights.

Overall our method is nearly 10 times faster than the sampling methods used in [13] and developed in [5], as shown in Figure 2, and approximately 100 times than MATLAB's Lanczos implementation applied to full resolution graph, as shown in Figure 3. As the multilevel method solves for the eigenvectors of the full fidelity matrix, the

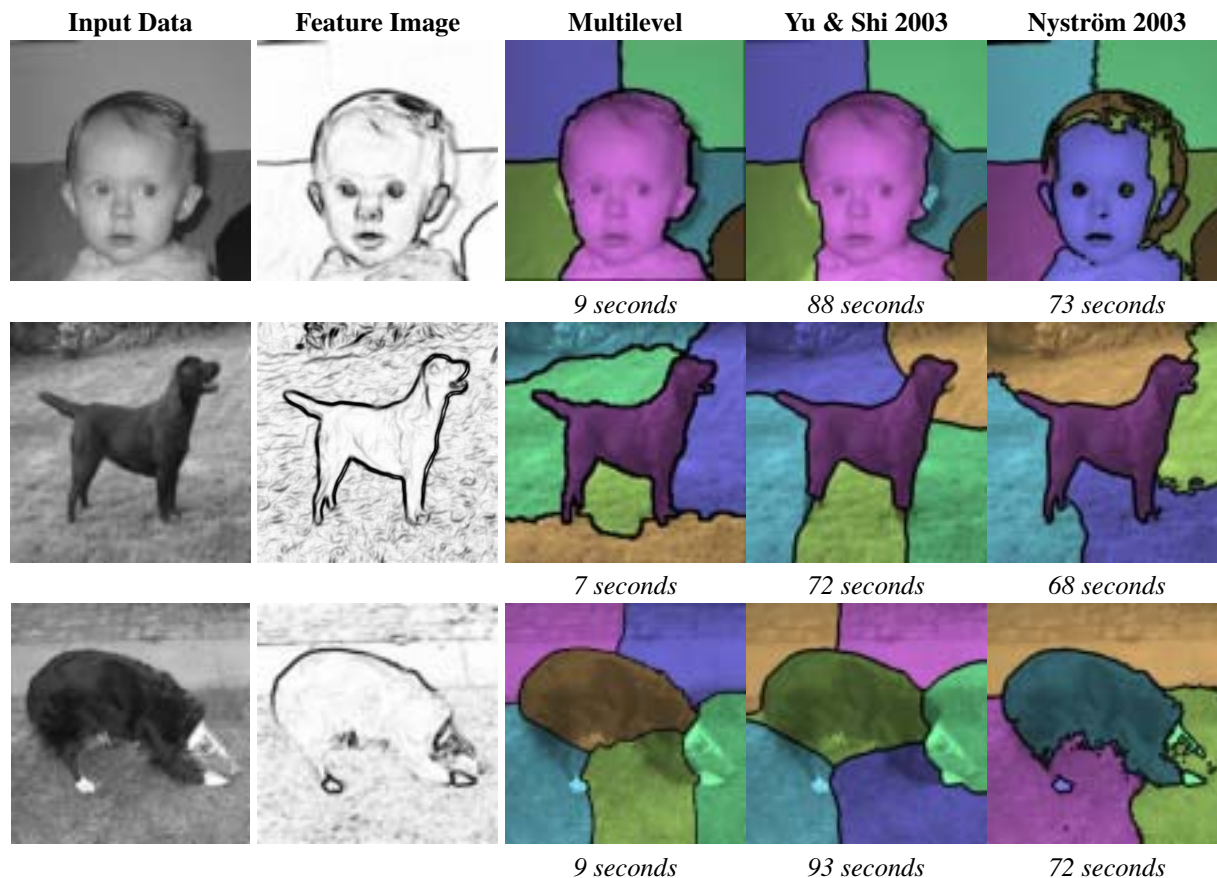


Figure 2: A comparison of accelerated NCuts image segmentation methods. Timing numbers are given for the eigenvector estimation only. The numbers were generated on an Apple 2GHz G5 processor running MATLAB with matrix multiplication in C. Results are given for the propose method (column 3), Yu and Shi [13] (column 4), and Nyström approximation [5] (column 5). The major computational bottleneck for [13] was in computing the partial eigensolution to the sample matrix via MATLAB's Lanczos implementation. For the Nyström approximation [5] the bottleneck was the inversion of a small sample matrix. In the multilevel method the majority of the computational time was spent on *subspace iterations* at the finest representation of the graph.

solutions are more consistent with the image data and do not violate the normalized cut criterion given in Equation 1.

Each sampling technique has deficits. The sampling technique used in [13] neglects intensity boundaries in the image. This effect is visible in the image of the child in row 1, column 4 and the limbs of the dog in row 2, column 4. The Nyström approximation fails in that it generates spuriously small segments such as the rear paw of the dog in row 3, column 5 and segments that are fragmented in the image plane. Both of these errors are alleviated by exploiting all the data in the graph as is done in the proposed multilevel method.

#### 4.1 Regularization through a Multilevel Method

The hierarchy of graph problems results in a well conditioned computation for estimating the eigenvectors of  $\mathcal{L}$ .

Figure 3 illustrates how a small change in the parameters used to generate the graph makes the problem ill-conditioned with respect to the MATLAB's Lanczos implementation. This results in long convergence times for the Lanczos algorithm in most cases as well as the occasional failure to converge.

## 5 Segmentation Tracking

We demonstrate a simultaneous segmentation and tracking technique that uses the solution for the previous frame to initialize solution estimation for the current frame. The method adapts to the shape of the object being tracked as the appearance is loosely modeled by the presence of separating edges between the target and the background. Tracking is accomplished by passing the eigensolution from the fine level of the hierarchy of the previous frame to the cur-

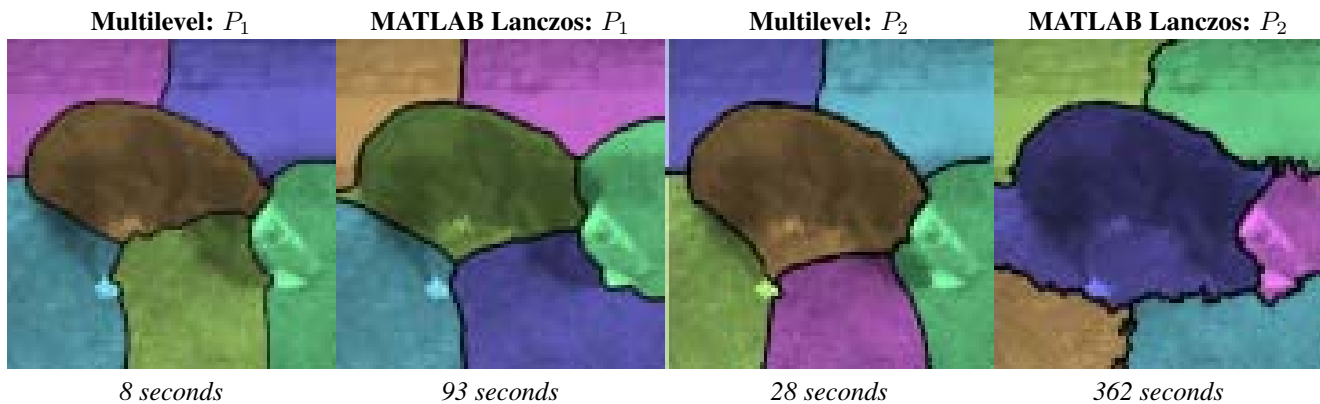


Figure 3: A comparison between the Multi-level Method and the MATLAB Lanczos solver. Notice that the *Lanczos* method is shown to be exceedingly sensitive to the graph generation parameters. In the  $P_1$  case the parameters are set as in [13], in  $P_2$  the sampling density in the pixel neighborhoods was increased by 15%, and the significance of the IC cue was decreased by 5%. The figure in the fourth column demonstrates that unconditioned *Lanczos* iterations fail to converge to a meaningful eigenvector for the problem. In contrast the multilevel method is far more robust to slight changes in the parameters used to control graph construction.

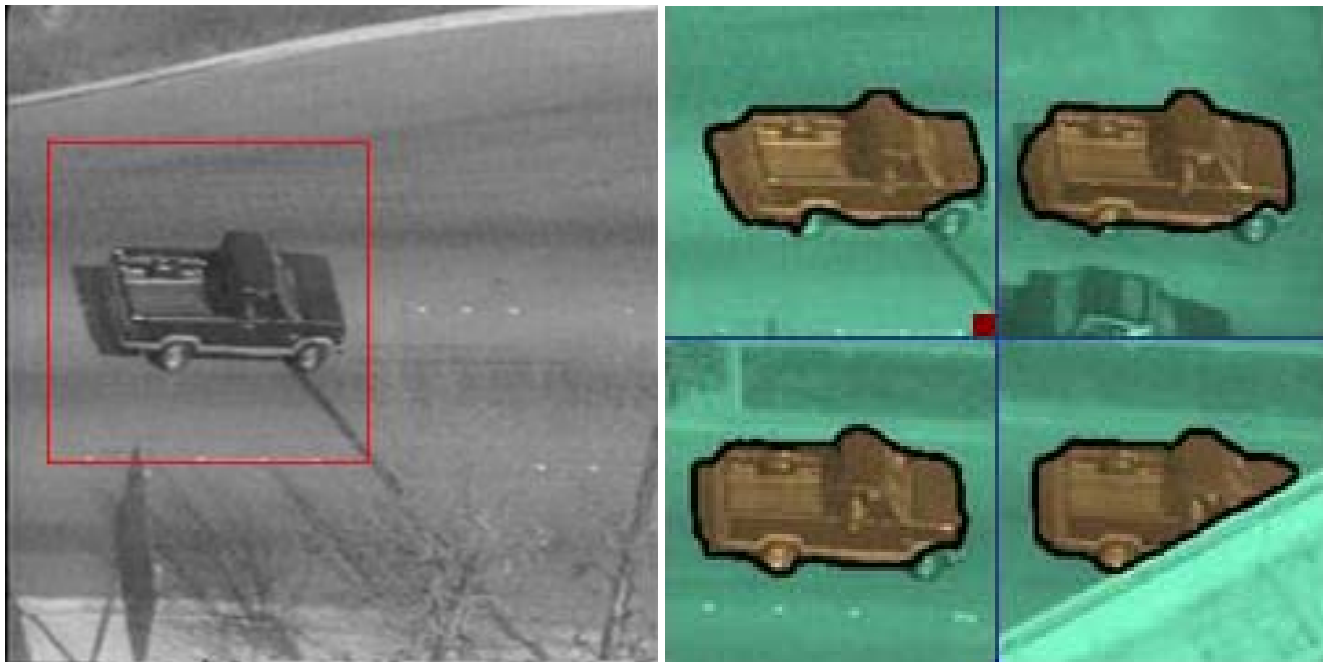


Figure 4: Simultaneous foreground object tracking and segmentation is achieved by using the continuous solution from the previous frame to initialize the estimation procedure for the current frame. This results in a tracker that is robust to smooth shape variation in the tracked object and surroundings. Timing: the target was tracked in a  $150 \times 150$  window over the image sequence. The solution for the **initialization frame**, marked with the red dot took, 2 seconds with the multilevel method. The remaining frames all took approximately  $\frac{1}{3}$  of a second to converge to a solution. (See supplemental video)

rent frame. This solution is then fit to the current problem using *subspace iterations*.

By constraining border pixels in the tracking window to the background, a foreground tracking algorithm emerges. Using the techniques described in [14], we integrate grouping constraints into the partitioning step by forcing a small set of known background pixels into one segment. This entails modifying the eigenproblem in Equation 5 by adding a constraint matrix of the form  $C^T X = 0$ . This matrix  $C$  defines a diffusion over pre-assigned pixel labels, forcing the target pixels into the foreground while suppressing spurious segmentations. The constraints in  $C$  are easily enforced by projecting  $\mathcal{L}$  onto the null space of  $C$ .

To initialize the tracker, a region is selected in the first frame as the target of interest. A simple rectangular constraint mask is automatically computed such that border pixels are constrained to be members of the background segment. Appropriately scaled instantiations of this mask are used in the hierarchy to enforce the constraints at each level. In the context of the tracking problem, the image region cut from the border pixels is labeled as the target. Examples of these partitions are given in Figure 5 for an automobile tracking sequence.

We employ a simple heuristic to cope with temporary occlusion and solution instability. Restarts of the subspace tracker are triggered when the foreground segment intersects the constraint mask. This operation involves backtracking to the previous frame, widening the target window, and computing a constrained eigensolution using the Lanczos algorithm at the coarsest level of the pyramid for that frame. The window is adjusted in size until the new estimate of the foreground segment matches the previous foreground estimate. If a satisfactory segment is not found the widened window is advanced in the image sequence along the predicted object trajectory.

## 6 Conclusion

We present a technique that reduces the computational cost of graph partitioning while yielding the exact continuous optimal solution to the spectral relaxation. This is accomplished through a hierarchical multilevel method that generates several reduced graph cut problems. In image segmentation, the method exploits the lattice structure of the pixel coordinates to define the multilevel method. A coarse solution is obtained using a standard eigensolver and propagated through increasingly refined problem instances. At each level the projected solution is refined by *subspace iterations* and then projected onto a further refined problem instance. The resulting graph-solution hierarchy defined by interpolation and refinement rules provides a good initial approximation to the eigenvectors of the Laplacian of the full graph.

This approach was applied to image segmentation problems using the normalized cut criterion [11]. The method yields an order of magnitude speed improvement over current sampling techniques. In tracking, solution reuse reduces the computation time by an additional order of magnitude.

We are currently extending the multilevel method to address large scale spectral clustering problems such as those posed in [9]. This involves building geometric kernels using the feature space geometry defined by the data. Similarly, we are examining irregularly sampled anisotropic kernels to preserve graph structure in image and volume segmentation problems. We are evaluating alternative iterative methods for computing eigenvectors such as the Grassmann Rayleigh Quotient Iteration [1]. Such methods demonstrate more rapid convergence than *subspace iterations*.

**Acknowledgments** This work is supported in part by DARPA/IXO VIVID contract NBCH1030013.

## References

- [1] P. Absil, R. Mahony, R. Sepulchre, and P. Van Dooren. A grassmann-rayleigh quotient iteration for computing invariant subspaces. *SIAM Review*, 44(1):57–73, 2002.
- [2] S. Barnard and H. D. Simon. A fast multilevel implementation of recursive spectral bisection for partitioning unstructured problems. *Concurrency*, 6(2), April 1994.
- [3] P. K. Chan, M. D. F. Schlag, and J. Y. Zien. Spectral k-way ratio-cut partitioning and clustering. *IEEE CAD-ICS*, 11(9):1088–1096, September 1994.
- [4] J. W. Demmel. *Applied Numerical Linear Algebra*. SIAM, 1st edition, 1997.
- [5] C. Fowlkes, S. Belongie, F. Chung, and J. Malik. Efficient spatiotemporal grouping using the nystrom method. In *PAMI*, December 2003.
- [6] R. Horn and C. R. Johnson. *Matrix Analysis*. Johns Hopkins Press, reprint edition, 1985.
- [7] J. Malik, S. Belongie, T. Leung, and J. Shi. Contour and texture analysis for image segmentation. In *IJCV*, volume 43(1), pages 7–27, June 2001.
- [8] M. Meila and J. Shi. A random walks view of spectral segmentation. In *AISTATS*, 2001.
- [9] A. Ng, M. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In *NIPS*, 2002.
- [10] E. Sharon, A. Brandt, and R. Basri. Fast multiscale image segmentation. In *CVPR*, pages 1070–1077, November 2000.
- [11] J. Shi and J. Malik. Normalized cuts and image segmentation. In *PAMI*, 2000.
- [12] S. H. Teng. Coarsening, sampling, and smoothing: Elements of the multilevel method. *UNM Technical Report*, 1998.
- [13] S. Yu and J. Shi. Multiclass spectral clustering. In *ICCV*, October 2003.
- [14] S. X. Yu and J. Shi. Grouping with bias. In *NIPS*, 2002.