Hierarchical Preconditioners for Computer Vision Problems

Richard Szeliski
Interactive Visual Media Group
Microsoft Research

NIPS Workshop on Numerical Mathematics
Challenges in Machine Learning

December 11, 2010
Variational problems in CV and CG

- Computer Graphics and Computational Photography applications:
  - Sparse data interpolation [Terzopoulos’86]
  - Poisson blending [Perez’03, Levin’04]
  - Colorization [Levin’04]
  - Interactive tone mapping [Lischinski’06]
Variational Problems

\[ E_s = \sum_{i,j} s_i^x (f_{i+1,j} - f_{i,j} - g_{i,j}^x)^2 + s_i^y (f_{i,j+1} - f_{i,j} - g_{i,j}^y)^2 \]

\[ E_d = \sum_{i,j} w_{i,j} (f_{i,j} - d_{i,j})^2 \]
Discrete quadratic energy

- Quadratic energy function
  \[ E = \frac{1}{2} x^T A x - b^T x + c \]
- Sparse linear system
  \[ Ax = b \]

How to solve sparse multi-banded system?
Solution techniques

How to solve sparse multi-banded system?

- Direct solvers have too much \textit{fill in}
- Iterative solvers perform better [Saad 03]
- Multi-grid [Briggs’00, Trottenberg’00]:
  - smooth inter-level transfers
- Hierarchical preconditioners:
  - global bases (wavelets)
Hierarchical basis preconditioning

- Perform change of basis
  \[ x = Sy \]
  \( x \): nodal (original) basis, \( y \): hierarchical basis

- New Hessian matrix
  \[ A' = S^TAS \]
  has better condition
Hierarchical basis preconditioning

- Intuitive explanation:
- **nodal** variables \( \mathbf{x} \) are too local \( \rightarrow \) highly coupled
Hierarchical basis preconditioning

- Intuitive explanation:
  - **nodal** variables \( x \) are too local → highly coupled
  - **hierarchical** variables \( y \) span low-frequency modes → more independent, better conditioning

[Yserentant’86, Szeliski’90, Gortler & Cohen’95]
Convergence

L=4  [Szeliski’90]  L=6
Convergence

- Rate depends on number of preconditioning levels $L$
- Optimal # of levels matches data spacing

[Szeliski’90]
Shortcomings

• Original hierarchical basis functions:
  – Need to hand-tune number of levels
  – Do not adapt to local data spacing and strengths
  – Do not adapt to local smoothness / discontinuities

• **Goal:** *automatically* adapt bases to local problem structure
Outline

• Variational problems in computer graphics
• Iterative solvers, hierarchical preconditioning
• 1D solution: cyclic reduction
• 2D solution: repeated red-black reduction
• Computer graphics applications
• Extensions and future work
1D first order problem

- Discrete 1D energy:

\[ E_d = \sum_i w_i(f_i - d_i)^2, \quad E_s = \sum_i s_i(f_{i+1} - f_i)^2 \]

- Eliminate the odd variables \( x_{2i+1} \) from \( Ax = b \)

\[
\begin{align*}
    a_{2i}x_{2i} + a_{2i+1}x_{2i+1} + a_{2i+2}x_{2i+2} &= b_{2i+1} \\
    x_{2i+1} &= a_{2i+1}^{-1} \left[ b_{2i+1} - a_{2i}x_{2i} - a_{2i+2}x_{2i+2} \right].
\end{align*}
\]
1D first order problem

- Eliminate the odd variables $x_{2i+1}$ from $Ax = b$

\[
\begin{align*}
 a_{2i}x_{2i} + a_{2i+1}x_{2i+1} + a_{2i+2}x_{2i+2} &= b_{2i+1}, \\
 x_{2i+1} &= a_{2i+1}^{-1}\left[b_{2i+1} - a_{2i}x_{2i} - a_{2i+2}x_{2i+2}\right].
\end{align*}
\]
1D first order problem

- Given the *even* variables, can *exactly* solve for the *odd* variables
  - permute even and odd variables

\[
A = \begin{bmatrix}
  D & E \\
  E^T & F
\end{bmatrix}
\quad \text{and} \quad
S_1 = \begin{bmatrix}
  I & -D^{-1}E \\
  0^T & I
\end{bmatrix}
\]
1D first order problem

- Preconditioned coarse level matrix is **tridiagonal**, just like the original problem

\[
\hat{A}_1 = S_1^T A S_1 = \begin{bmatrix}
D & 0 \\
0^T & F - E^T D^{-1} E
\end{bmatrix} = \begin{bmatrix}
D & 0 \\
0^T & \hat{A}_1^c
\end{bmatrix}
\]

- Can recurse all the way to a diagonal preconditioned Hessian
Variable elimination

• Eliminating intermediate nodes is an old “trick” in graphical models
  – junction tree algorithms
  – easy to do if only one or two parents
  – otherwise, introduce larger cliques
Sample 1D problem
Sample 1D problem: convergence
1D first order problem

- Cyclic reduction $\leftrightarrow$ perfect hierarchical basis preconditioning (weighted wavelets) [Sweldens, SIAM J. Math. Anal. 97]

- Can we extend this to 2D?
Outline

• Variational problems in computer graphics
• Iterative solvers, hierarchical preconditioning
• 1D solution: cyclic reduction
• 2D solution: repeated red-black reduction
• Computer graphics applications
• Extensions and future work
2D first order problem

- Idea: eliminate *red* variables in red-black checkerboard (half-octave pyramid)

- Red variables only depend on black ones

- Repeated Red-Black (RRB) [Ciarlet 94]
2D first order problem

- Problem: bandwidth grows from $N_4$ to $N_8$

- Need to eliminate the “diagonal” $N_8$ connections
2D first order problem

• Solution: redistribute “diagonal” $N_8$ edge weights to desired $N_4$ connections

• Question: will this heuristic work?
Sample 2D problem (32x32)

- Synthetic 32x32 example, tear across bottom
  L=1(CG)  L = 3  LAHBF  solution

iter = 1
Sample 2D problem (32x32)
Sample 2D problem (32x32)

log error plot (convergence rate)
Relationship to previous RRB work

• New strategy redistributes neglected off-diagonal terms to other “springs”
  – Maintains overall “stiffness”

• Previous strategies either
  – Drop these terms (ILU0)
  – Add them back to diagonal (MILU)
  → Poorer approximation to fine-level problem
Sample 2D problem (32x32)
Algorithm summary: PCG

- Precompute basis functions (and stiffness matrices), fine → coarse
- Iterate:
  1. Compute current residual vector $r = Ax - b$
  2. Compute hierarchical residual $r' = S^T r$
  3. Divide by (preconditioned) $A'$ [diagonal $D$ (?)]
  4. Compute nodal (smoothed) $r'' = SD^{-1}S^T r$
  5. Compute conjugated descent direction, take a downhill step
GPU Implementation

• Each of the previous steps can be computed in parallel on a GPU [Bolz et al.’03].

• Pyramid operations require $L$ passes
  – can terminate early, use dense CPU solver
  – for really large problems, use streaming multigrid [Kazhdan & Hoppe’08]
Outline

• Variational problems in computer graphics
• Iterative solvers, hierarchical preconditioning
• 1D solution: cyclic reduction
• 2D solution: repeated red-black reduction
• Computer graphics applications
• Extensions and future work
Colorization

• Interpolate chrominance (U,V) with weak membrane [Levin’04]
Colorization
Poisson Blending

• Reconstruct image from gradient field

[Perez et al. 2003, Agarwala et al. 2004]
Poisson Blending
Interactive Tone Mapping

• Reconstruct exposure maps from sparse strokes [Lischinski et al. 2006]
Other applications

• HDR tone mapping (range compression)
• Single view modeling [Zhang et al. 2001]
• Reflection removal [Agrawal et al. 2005]
• Closed form matting [Levin et al. 2006]
• More SIGGRAPH papers …
Outline

• Variational problems in computer graphics
• Iterative solvers, hierarchical preconditioning
• 1D solution: cyclic reduction
• 2D solution: repeated red-black reduction
• Computer graphics applications
• Extensions and future work
Eigenvalue analysis
Spectral Graph Sparsification p. I
The Combinatorial Multigrid Solver

Gary Miller, Richard Peng, CMU
Yiannis Koutis, CMU → U of Puerto Rico, Rio Piedras

University of Pittsburgh Medical Center
Center for Computational Thinking
Carnegie Mellon
Microsoft Research
Laplacians of weighted graphs

- Random Walk Matrix: $D^{-1}L$
- Electrical network, Ohm's law: $Lv = i$
- Commute time of random walk proportional to effective resistance
Measuring graph similarity

A measure of Laplacian similarity

The support number

\[ \sigma(L_A, L_B) = \max_v \frac{v^T L_A v}{v^T L_B v} \]

The condition number

\[ \kappa(L_A, L_B) = \sigma(L_A, L_B) \sigma(L_B, L_A) \]
Measuring graph similarity

The Rayleigh Quotient: $v^T L v = \sum_{i,j} w_{i,j} (v_i - v_j)^2$

Measure of similarity of the energy profile of the two networks
Simple bounds on graph similarity

- Case II: A line and a line with two short loops

- Dilation of the embedding is 2
- Congestion on \((v_2, v_3)\) is 3
- Condition number is at most congestion \(*\) dilation
Eigenvalue analysis

• Locally adaptive hierarchical basis functions are constructed by alternating:
  1. coarsening (assuming red-black partition)
  2. sparsification (nearby resistive grid)

• How *good* is the approximation produced by the sparsification?
Eigenvalue analysis

• Reminder: redistribute “diagonal” $N_8$ edge weights to desired $N_4$ connections

• Question: what is the relative condition number?
Eigenvalues of simple Laplacian

I(1) = 0.000001
I(2) = 0.034055
I(3) = 0.034055
I(4) = 0.068109

I(287) = 7.830892
I(288) = 7.830892
I(289) = 7.931893

c.n. = 232.915573
Generalized eigenvalues $9 \rightarrow 5$
$n$-level preconditioning

- Use hierarchical basis with $n$ levels
- Solve exactly at coarse level
- What is relative condition number?
One-level full octave (non-adaptive)
Two-level full octave (non-adaptive)
Three-level full octave (non-adaptive)
One-level half octave (adaptive)
Next half octave (adaptive)
“Two-level” half octave (adaptive)
“Three-full-levels” half octave (adap.)
What’s going on???

- Half the eigenvectors are well approximated
- Most of the rest spread between $[0.25 \ 1]$
- $\kappa \approx 4$

$$\|e_{(i)}\|_A \leq 2 \left( \frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1} \right)^i \|e_{(0)}\|_A$$

- Is this a fluke, or repeatable across different problems?
What about multigrid?

• Multigrid uses pre- and post-smoothing steps to reduce aliasing during coarse level projection (restriction) and interpolation (prolongation)

• Takes care of fine-level errors

• Can we add smoothing to hierarchical basis preconditioning?
Multigrid vs. LAHBF

2D Membrane (L = 5, half-octave = 0) MSE Error

-6
-8
-10
-12
-14
-16
-18
-20
-22
-24
0 2 4 6 8 10 12 14 16 18 20

HBF
V-sweep
V-sweep SD
V-sweep CG
LAHBF
Half-octave Multigrid vs. LAHBF

2D Membrane (L = 5, half-octave = 1) MSE Error

-6
-8
-10
-12
-14
-16
-18
-20
-22
-24
0 2 4 6 8 10 12 14 16 18 20

HBF
V-sweep
V-sweep SD
V-sweep CG
LAHBF
Irregular problem with tear
Extension: 9→5 pre-sparsification

• Eliminate “diagonal” links before coarsening

• Can be used to solve $N_8$ problems, such as colorization
Extension: 9→7 sparsification

• Only eliminate red-red diagonals

• In theory, less approximation error; in practice, doesn’t seem to matter
What about multigrid?

• Can we add smoothing to hierarchical basis preconditioning?
• … still working on this, as well as condition number analysis for multigrid …
Limitations

- Red-black coarsening (or full octave) won’t work on highly irregular problems
  - For example, spiral embedded in grid
- Algebraic multigrid and combinatorial multigrid adaptively choose coarse-level variables
- Need to combine both approaches
Future work

- Extend to $2^{nd}$ order problems?
  - Tricky, because no independent sets
  - Gremban’s trick to split into two Laplacians?
Future work

• Extend to general Markov Random Fields
  – Same idea of coarsening and approximation
  – Minimizing Energy Functions on 4-connected Lattices Using Elimination, [Carr and Hartley’09]. Binary MRFs, used to get sub-modularity
  – Potentials get more complex for multi-label or non-quadratic continuous potentials
  – Combine with other kinds of trees or “thinning”?
Conclusions

- Effective multi-level preconditioner for irregular / inhomogeneous 1st-order GMRFs

- Useful for a number of applications:
  - Poisson and gradient-domain blending
  - Colorization [Levin’04]
  - Interactive tone mapping [Lischinski’06]

- General framework for solving PDEs/MRFs?