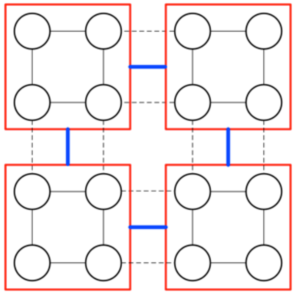


Inference by Learning: Speeding-up Graphical Model Optimization via a Coarse-to-Fine Cascade of Pruning Classifiers (NIPS 2014)

On-line resources: http://imagine.enpc.fr/~conejob/ibyl/

TBD, X/X/2014

Bruno Conejo, Phd student Ecole des Ponts ParisTech / Research Analyst GPS, Caltech

Nikos Komodakis, Associate Professor Ecole des Ponts ParisTech

with S. Leprince & JP. Avouac (GPS, Caltech)

Introduction:

Associated materials:

Paper, code, slides and poster are available on-line :

http://imagine.enpc.fr/~conejob/ibyl/

Motivations:

1. Speed-up the inference of MRFs that have a piecewise smooth MAP.
2. While maintaining the accuracy of the inference solution of MRFs.

Approach:

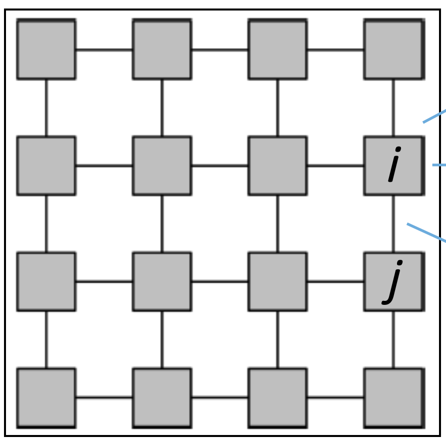
1. Exploit the piecewise smooth structure of the MAP by iteratively optimizing a coarse to fine representation of the MRF.
2. Rely on learning to progressively reduce the solution space.

Notations:

Let's consider the following discrete MRF:

$$\mathcal{M} = (\mathcal{G} = [\mathcal{V}, \mathcal{E}], \mathcal{L}, \phi = [\{\phi_i\}_{i \in \mathcal{V}}, \{\phi_{ij}\}_{(i,j) \in \mathcal{E}}])$$

support graph ($\mathcal{G}$), *set of vertices* ($\mathcal{V}$), *set of edges* ($\mathcal{E}$), *discrete label set* ($\mathcal{L}$), *set of unary terms* ($\{\phi_i\}$), *set of pairwise terms* ($\{\phi_{ij}\}$)



$\mathcal{G} = [\mathcal{V}, \mathcal{E}]$

$x_i \in \mathcal{L}$: the configuration of vertex *i*

$\phi_i : \mathcal{L} \to \mathbb{R}$: the unary terms of vertex *i*

$\phi_{ij} : \mathcal{L}^2 \to \mathbb{R}$: the pairwise terms of edge *ij*

$x = (x_i)_{i \in \mathcal{V}}$: the configuration of the MRF $\mathcal{M}$

Notations:

The Energy: i.e., the total cost is given by

$$E(x|\mathcal{M}) = \sum_{i \in V} \phi_i(x_i) + \sum_{(i,j)\in\mathcal{E}} \phi_{ij}(x_i, x_j)$$

The MAP: Maximum At Posteriori

$$x_{\mathrm{MAP}} = \arg\min_{x \in \mathcal{L}^{|\mathcal{V}|}} E(x|\mathcal{M})$$

The pruning matrix: $A : \mathcal{V} \times \mathcal{L} \to \{0, 1\}$ is defined by:

$$A(i,l) = \begin{cases} 1 & \text{if label } l \text{ is active at vertex } i \\ 0 & \text{if label } l \text{ is pruned at vertex } i \end{cases}$$

and its associated solution space:

$$\mathcal{S}(\mathcal{M}, A) = \left\{ x \in \mathcal{L}^{|\mathcal{V}|} \mid (\forall i), A(i, x_i) = 1 \right\}$$

Approach:

To obtain an inference speed-up we solve:

$$\min_{x \in \mathcal{S}(\mathcal{M}, A)} E(x|\mathcal{M})$$

With A such that:

x_{MAP} belongs to $\mathcal{S}(\mathcal{M}, A)$

most elements of A are 0 (i.e., the labels are pruned)

<u>The IbyL framework</u> iteratively estimates A by:

1. Building a coarse to fine set of MRFs from $\mathcal{M}$
2. Learning at each scale pruning classifiers to refine A

Model coarsening:

Given $\mathcal{M}$ and a grouping function $g : \mathcal{V} \rightarrow \mathcal{N}$, we create a coarsen MRF (slight abuse of the notation):

$$g(\mathcal{M}) = \mathcal{M}' = \left(\mathcal{V}', \mathcal{E}', \mathcal{L}, \{\phi'_i\}_{i \in \mathcal{V}'}, \{\phi'_{ij}\}_{(i,j) \in \mathcal{E}'}\right)$$

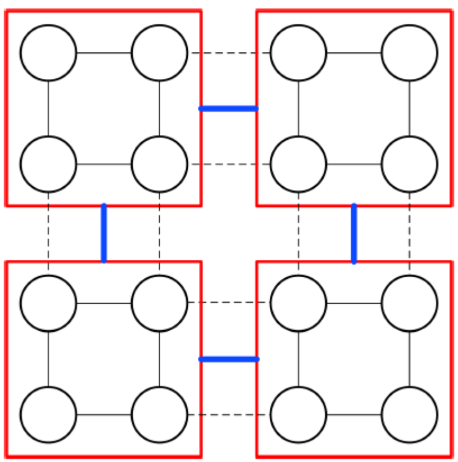


Figure 1: Black circles: $\mathcal{V}$, Black lines: $\mathcal{E}$, Red squares: $\mathcal{V}'$, Blue lines: $\mathcal{E}'$.

The vertices and edges of $\mathcal{M}'$ are given by

$$\mathcal{V}' = \{i' \mid \exists i \in \mathcal{V}, i' = g(i)\},$$

$$\mathcal{E}' = \{(i', j') \mid \exists (i,j) \in \mathcal{E}, i' = g(i), j' = g(j), i' \neq j'\}$$

The unary and pairwise potentials of $\mathcal{M}'$ are given by

$$(\forall i' \in \mathcal{V}'), \quad \phi'_{i'}(l) = \sum_{i \in \mathcal{V} \mid i' = g(i)} \phi_i(l) + \sum_{(i,j) \in \mathcal{E} \mid i' = g(i) = g(j)} \phi_{ij}(l, l)$$

$$(\forall (i', j') \in \mathcal{E}'), \quad \phi'_{i'j'}(l_0, l_1) = \sum_{(i,j) \in \mathcal{E} \mid i' = g(i), j' = g(j)} \phi_{ij}(l_0, l_1)$$

Coarse to fine optimization and label pruning

We iteratively apply the previous coarsening to build a coarse to fine set of N+1 progressively coarser MRFs:

$$\mathcal{M}^{(0)} = \mathcal{M} \quad \text{and} \quad (\forall s),\ \mathcal{M}^{(s+1)} = g^{(s)}(\mathcal{M}^{(s)})$$

with:

$$\mathcal{M}^{(s)} = \left(\mathcal{V}^{(s)}, \mathcal{E}^{(s)}, \mathcal{L}, \{\phi_i^{(s)}\}_{i \in \mathcal{V}^{(s)}}, \{\phi_{ij}^{(s)}\}_{(i,j) \in \mathcal{E}^{(s)}}\right)$$

We set all elements of the coarsest pruning matrix $A^{(N)}$ to 1 (no pruning)

At each scale (s) we apply the following steps:

1. Optimize the current MRF $\mathcal{M}^{(s)}$
2. Update the next scale pruning matrix $A^{(s-1)}$
3. Up-sample the current solution for next scale.

Coarse to fine optimization and label pruning

Step 1: Optimize the current MRF:

$$x^{(s)} \approx \arg\min_{x \in \mathcal{S}(\mathcal{M}^{(s)}, A^{(s)})} E(x|\mathcal{M}^{(s)})$$

Step 2: Update the next scale pruning matrix:

i. Compute feature map:

$$z^{(s)} : \mathcal{V}^{(s)} \times \mathcal{L} \rightarrow \mathbb{R}^K$$

ii. Update pruning matrix from off-line trained classifier:

$$f^{(s)} : \mathbb{R}^K \rightarrow \{0, 1\}$$

$$A^{(s-1)}(i, l) = f^{(s)}(z^{(s)}(g^{(s-1)}(i), l))$$

Step 3: Up-sample the current solution:

$$x^{(s-1)} = [g^{(s-1)}]^{-1}(x^{(s)})$$

Inference by learning framework

Algorithm 1: Inference by learning framework

Data: Model $\mathcal{M}$, grouping functions $(g^{(s)})_{0\leq s<N}$, classifiers $(f^{(s)})_{0<s\leq N}$
Result: $x^{(0)}$
Compute the coarse to fine sequence of MRFs:
$\mathcal{M}^{(0)} \leftarrow \mathcal{M}$
for $s = [0 \ldots N-1]$ **do**
  $\mathcal{M}^{(s+1)} \leftarrow g^{(s)}(\mathcal{M}^{(s)})$
Optimize the coarse to fine sequence of MRFs over pruned solution spaces:
$(\forall i \in \mathcal{V}^{(N)}, \forall l \in \mathcal{L}),\ A^{(N)}(i,l) \leftarrow 1$
Initialize $x^{(N)}$
for $s = [N...0]$ **do**
  Update $x^{(s)}$ by iterative minimization: $x^{(s)} \approx \arg\min_{x\in\mathcal{S}(\mathcal{M}^{(s)},A^{(s)})} E(x|\mathcal{M}^{(s)})$
  if $s \neq 0$ **then**
    Compute feature map $z^{(s)}$
    Update pruning matrix for next finer scale: $A^{(s-1)}(i,l) = f^{(s)}(z^{(s)}(g^{(s-1)}(i),l))$
    Upsample $x^{(s)}$ for initializing solution $x^{(s-1)}$ at next scale: $x^{(s-1)} \leftarrow [g^{(s-1)}]^{-1}(x^{(s)})$

We still need to define:

1. How to compute the feature map $z^{(s)}$
2. How to train the classifiers $f^{(s)}$

Feature map

We stack *K* individual scalar features defined on $\mathcal{V}^{(s)} \times \mathcal{L}$

Presence of strong discontinuity:

$$\mathrm{PSD}^{(s)}(i,l) = \begin{cases} 1 & \exists (i,j) \in \mathcal{E}^{(s)} |\ \phi_{ij}(x_i^{(s)}, x_j^{(s)}) > \rho \\ 0 & \text{otherwise} \end{cases}$$

Local energy variation:

$$\mathrm{LEV}^{(s)}(i,l) = \frac{\phi_i^{(s)}(l) - \phi_i^{(s)}(x_i^{(s)})}{N_{\mathcal{V}}^{(s)}(i)} + \sum_{j:(i,j)\in\mathcal{E}^{(s)}} \frac{\phi_{ij}^{(s)}(l, x_j^{(s)}) - \phi_{ij}^{(s)}(x_i^{(s)}, x_j^{(s)})}{N_{\mathcal{E}}^{(s)}(i)}$$

with $N_{\mathcal{V}}^{(s)}(i) = \mathrm{card}\{i' \in \mathcal{V}^{(s-1)} : g^{(s-1)}(i') = i\}$ and $N_{\mathcal{E}}^{(s)}(i) = \mathrm{card}\{(i',j') \in \mathcal{E}^{(s-1)} : g^{(s-1)}(i') = i, g^{(s-1)}(j') = j\}$.

Unary coarsening:

$$\mathrm{UC}^{(s)}(i,l) = \sum_{i'\in\mathcal{V}^{(s-1)} | g^{(s-1)}(i')=i} \frac{\left|\phi_{i'}^{(s-1)}(l) - \frac{\phi_i^{(s)}(l)}{N_{\mathcal{V}}^{(s)}(i)}\right|}{N_{\mathcal{V}}^{(s)}(i)}$$

Learning pruning classifiers

On a training set of MRFs, we run the IbyL framework without any pruning, i.e., $A^{(s)} \equiv 1$. We keep track of features and compute:

$$X_{\mathrm{MAP}}^{(s)} : \mathcal{V}^{(s)} \times \mathcal{L} \rightarrow \{0, 1\}$$

such that:

$$(\forall i \in \mathcal{V}, \forall l \in \mathcal{L}), \quad X_{\mathrm{MAP}}^{(0)}(i, l) = \begin{cases} 1, & \text{if } l \text{ is the ground truth label for node } i \\ 0, & \text{otherwise} \end{cases}$$

$$(\forall s > 0)(\forall i \in \mathcal{V}^{(s)}, \forall l \in \mathcal{L}), \quad X_{\mathrm{MAP}}^{(s)}(i, l) = \bigvee_{i' \in \mathcal{V}^{(s-1)} : g^{(s)}(i') = i} X_{\mathrm{MAP}}^{(s-1)}(i', l) \ ,$$

where $\bigvee$ denotes the binary OR operator.

Learning pruning classifiers

We split the features in two groups w.r.t the PSD feature.

For each group:

1. We have two classes (c0=to prune and c1=to remain active) defined from $X_{\mathrm{MAP}}^{(s)}$
2. We weight c0 to 1, and c1 to $\lambda\frac{\mathrm{card}(c_0)}{\mathrm{card}(c_1)}$
3. We train a linear C-SVM classifier

During testing, the classifier $f^{(s)}$ apply the trained classifier of the corresponding group (w.r.t. the PSD feature).

Experiments: Setup

We experiment with two problems:

1. Stereo-matching.
2. Optical flow.

We evaluate different pruning aggressiveness factor λ , and compute:

1. The speed-up w.r.t. the direct optimization.
2. The ratio of active labels.
3. The energy ratio w.r.t. the direct optimization.
4. The MAP agreement w.r.t. the best computed solution.

As an optimization sub-routine we use Fast-PD.

Stereo matching:

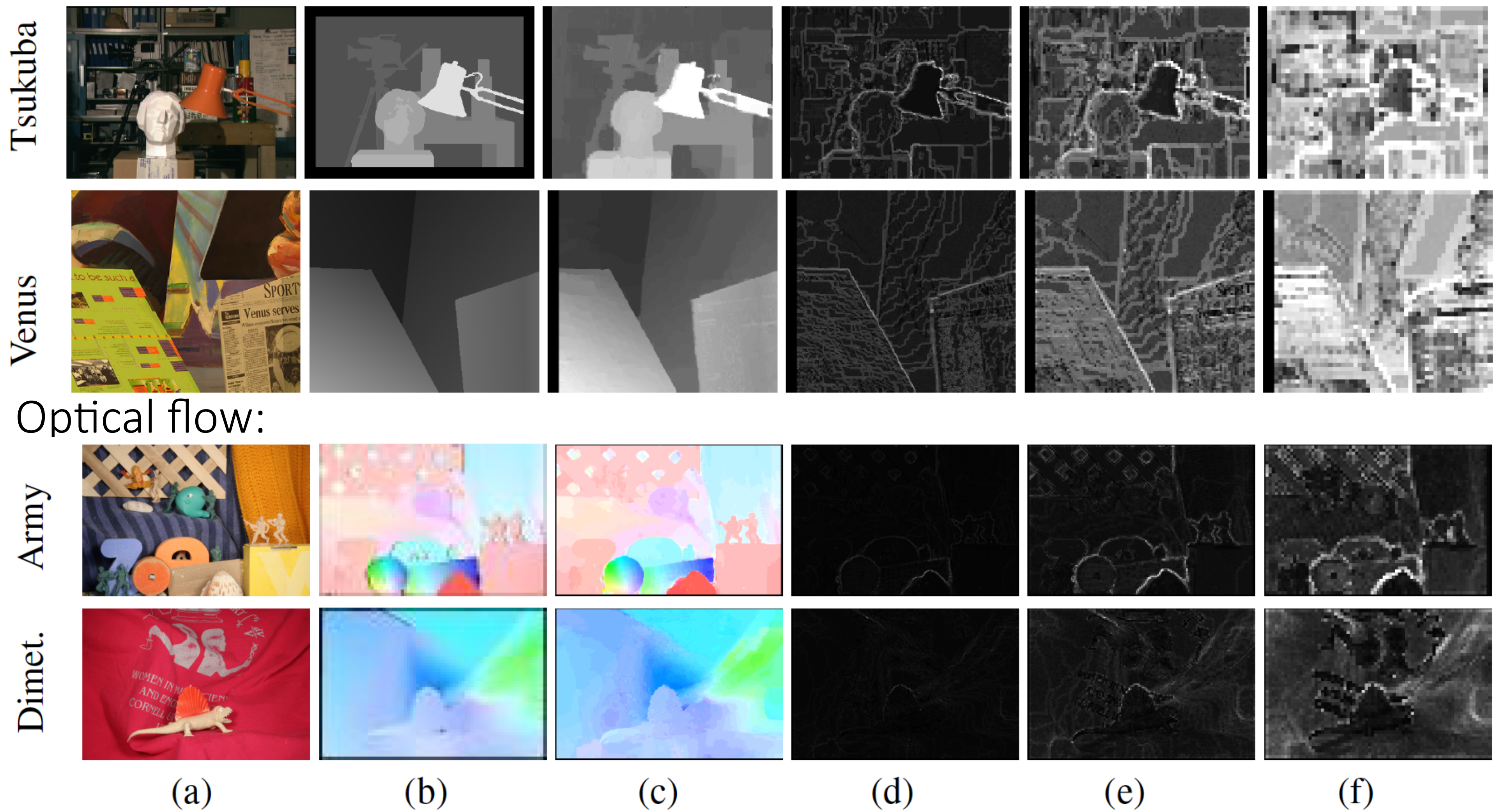
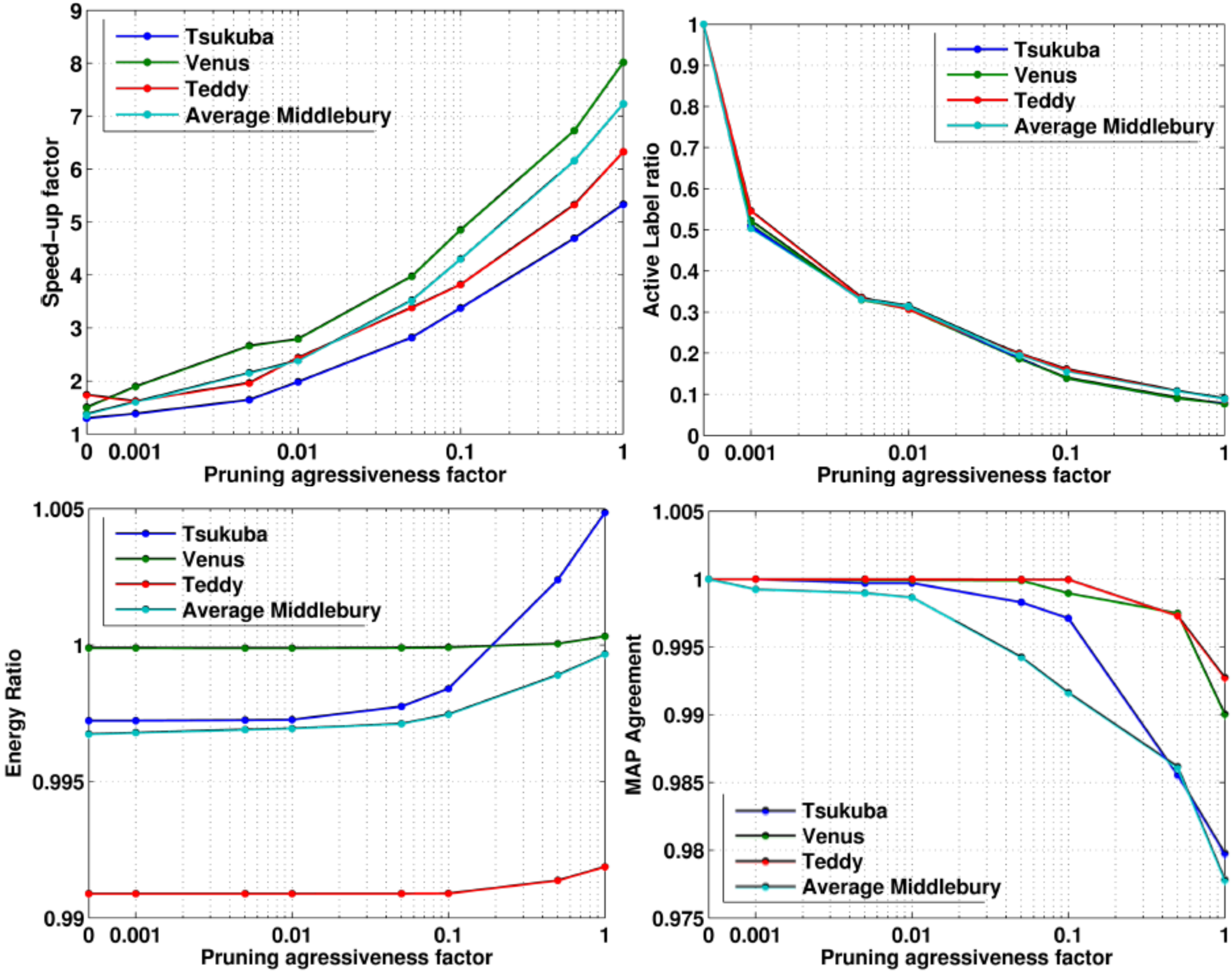
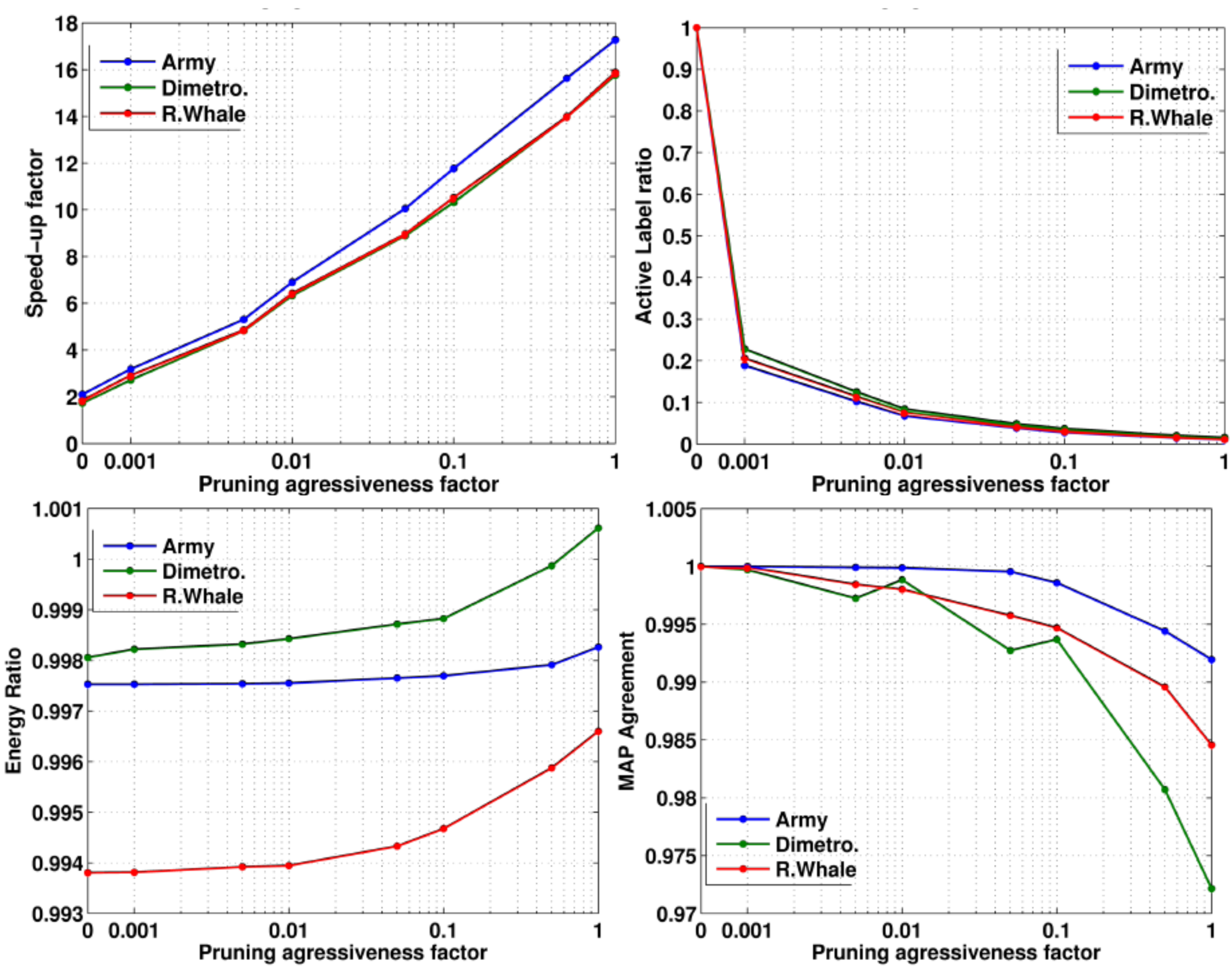


Figure 3: Results of our Inference by Learning framework for $\lambda = 0.1$. Each row is a different MRF problem. (a) original image, (b) ground truth, (c) solution of the pruning framework, (d,e,f) percentage of active labels per vertex for scale 0, 1 and 2 (black 0%, white 100%).

Experiments: Stereo matching



Experiments: Optical Flow



Conclusions

The IbyL framework:

1. Gives an important speed-up while maintaining excellent accuracy of the solution.

2. Can be easily adapted to any MRF task by computing task dependent features.

3. Can be easily adapted to high order MRFs.

4. Is available to download at:
   http://imagine.enpc.fr/~conejob/ibyl/


THANK
YOU
GRACIAS
ARIGATO
SHUKURIA
DANKSCHEEN
BİYAN
SHUKRIA
TINGKI
YAQHANYELAY
TASHAKKUR ATU
CHALTU
NUHUN
SPASSIBO
SNACHALHUYA
YUSPAGARÄTAM
WABEEJA
MAITEKA
HUI
SUKSAMA
EKHMET
DHANYABAAD
ANIHA
ATTO
MERSI
SPASIBO
DENKAUJA
NENACHALHYA
UNALCHÉESH
GUI
HATUR
EKOJU
SIKOMO
MAKETAI
MEHRBANI
GRAZIE
MAAKE
KOMAPSUMNIDA
SANCO
MERASTAWHY
GAEJTHO
LAH
PALDIES
GOZAIMASHITA
EFCHARISTO
AGUYJE
FAKAAUE
TAVTAPUCH
MEDAWAGSE
BAIIKA
JUSPAXAR
BOLZÏN
MERCI
MINMONCHAR