Memory Efficient Max Flow for Multi-label Submodular MRFs

Thalaiyasingam Ajanthan\textsuperscript{1,2} Richard Hartley\textsuperscript{1,2} Mathieu Salzmann\textsuperscript{3} \\
\textsuperscript{1}Australian National University \textsuperscript{2}Data61, CSIRO \textsuperscript{3}CVLab, EPFL

Introduction

Problem: Minimize a multi-label MRF with pairwise interactions

\[ E(x) = \sum_{i \in V} \theta_i(x_i) + \sum_{(i,j) \in E} \theta_{ij}(x_i, x_j), \]

where \( x_i \in \{0, 1, \cdots, L - 1\}. \)

Multi-label submodular:

\[ \theta_{ij}(x', y') + \theta_{ij}(y', x') - \theta_{ij}(x, y) - \theta_{ij}(y, x) \geq 0, \]

\[ \text{for all } i, j, x, y, \text{ where } L < x < x' \text{ and } y < y'. \]

Current method: Ishikawa algorithm [3].

\[ \text{Memory complexity: } O(|V|^2). \]

Contribution: An algorithm with memory complexity \( O(|E| \ell). \)

Memory Efficient Flow Encoding

Idea: Don’t store the residual graph but exit-flows between each pair of neighbouring columns.

Exit-flow: Given flow \( \psi \), an exit-flow is defined as

\[ \Sigma_{j \neq i} = \sum_{\psi_{j \neq i}}. \]

The residual graph can be rapidly computed from the exit-flows.

Rapidly computing the residual graph:


Algorithm

\[ \begin{align*}
\text{Require: } & \phi^0 & \text{Initial Ishikawa capacities} \\
\Sigma & \leftarrow 0 & \text{Initialize exit-flows} \\
\text{repeat} & \\
& P \leftarrow \text{augmenting path(} \phi^0, \Sigma \text{)} & \\
& \Sigma \leftarrow \text{augment}(P, \phi^0, \Sigma) & \\
\text{until no augmenting paths possible} & \\
\end{align*} \]

Assumption: \( \phi^0 \) can be stored in an efficient manner.

Efficiently Finding an Augmenting Path

Idea: Search for augmenting paths in a simplified graph.

Simplified graph:

- Unweighted sparse graph.
- Fewer augmenting paths.
- Good empirical performance.

Search-tree-recycling:

- Additional ideas.

Augmentation

Idea: Pass flow around loops.

- Push the maximum permissible flow through each flow-loop.
- Applying flow-loops translates to updating the exit-flows.

Experiments

Dataset:

- Middlebury stereo and inpainting instances.
- KITTI stereo instance.

Problem | Labels | Memory [MB] | Time [s] | Memory EIBFS MEMF | Time EIBFS MEMF
\hline
Tsukuba & 16 & 3195 & 2495 & 211 & 14 & 4 & 30
\hline
Venus & 20 & 7626 & 5907 & 396 & 35 & 9 & 60
\hline
Sawtooth & 20 & 7566 & 5860 & 393 & 31 & 8 & 35
\hline
Map & 30 & 6454 & 4946 & 219 & 57 & 9 & 36
\hline
Cones & 60 & *72303 & *55063 & 1200 & - & - & 371
\hline
Teddy & 60 & *72303 & *55063 & 1200 & - & - & 2118
\hline
KITTI & 40 & *88413 & *67316 & 2215 & - & - & 19008
\hline
Penguin & 256 & *173893 & *130728 & 663 & - & - & 6835
\hline
House & 256 & *521853 & *392315 & 1986 & - & - & 9290
\hline
\end{tabular}

Comparison with other max-flow implementations

Empirical time complexity: \( O(|V|^2 \ell) \)

Code: https://github.com/tajanthan/memf

References