

Non-Separable Multi-Dimensional Network Flows for Visual Computing

V. Ehm^{1,2} D. Cremers^{1,2} F. Bernard³

¹TU Munich ² Munich Center for Machine Learning ³University of Bonn

Abstract

Flows in networks (or graphs) play a significant role in numerous computer vision tasks. The scalar-valued edges in these graphs often lead to a loss of information and thereby to limitations in terms of expressiveness. For example, oftentimes high-dimensional data (e.g. feature descriptors) are mapped to a single scalar value (e.g. the similarity between two feature descriptors). To overcome this limitation, we propose a novel formalism for non-separable multi-dimensional network flows. By doing so, we enable an automatic and adaptive feature selection strategy – since the flow is defined on a per-dimension basis, the maximizing flow automatically chooses the best matching feature dimensions. As a proof of concept, we apply our formalism to the multi-object tracking problem and demonstrate that our approach outperforms scalar formulations on the MOT16 benchmark in terms of robustness to noise.

CCS Concepts

• **Theory of computation** → Design and analysis of algorithms; Theory and algorithms for application domains;

1. Introduction

Network flow algorithms are popular in computer vision and image analysis due to their broad range of applications, including image segmentation [EBA06] or multi-object tracking (MOT) [ZLN08]. Yet, they require that multi-dimensional information is mapped to scalar values which often has the downside that information is lost and thereby the expressiveness limited. Hence, in this work we take the first step to explore the novel direction of non-separable multi-dimensional flows:

- For the first time we present a **non-separable multi-commodity flow formulation**, where a flow unit with all its commodities cannot be separated throughout the graph.
- As a proof of concept, we show how our formulation can be applied in the context of **multi-object tracking**, for which we demonstrate that it increases the **robustness to noise**.

2. Related Work

Garg *et al.* [GVY96] extend traditional maximum flows [HR55] to multi-commodity flows by considering multi-dimensional flows per edge. Li *et al.* [LAB10] use a binary variable to ensure that only one path per commodity can be used to reach the sink. To the best of our knowledge, there does not exist a non-separable maximum multi-commodity formulation, where non-separable, in this case, means that only one path for all commodities is used.

While traditional MOT approaches [ZLN08] minimize the cost through the graph we maximize a multi-commodity flow.

3. Method

A graph is a tuple $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where \mathcal{V} represents the nodes and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ the set of edges. In the scalar maximum flow approach [HR55] the sum of flows $f_{uv} \in \mathbb{R}$ from the source node $s \in \mathcal{V}$ to the sink node $t \in \mathcal{V}$ is maximized while not exceeding the capacity $c_{uv} \in \mathbb{R}_+$ on every edge (see Figure 1).

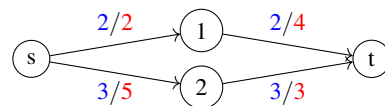


Figure 1: Scalar Maximum Flow: Maximize the flow (blue) from source node s to sink node t while not exceeding the capacity (red).

In comparison to the scalar approach, we assign a capacity vector $c_{uv} \in \mathbb{R}_+^k$ with k dimensions to every edge $(u, v) \in \mathcal{E}$. Additionally, we define a multi-dimensional flow $f_{uv} \in \mathbb{R}^k$ and a decision variable $b_{uv} \in \{0, 1\}$, indicating whether an edge is active or not. An active edge means that the flow can be non-zero, whereas an inactive edge ensures that its flow is the zero vector. We impose the capacity constraint that each flow f_{uv} through an edge must be elementwise smaller than its capacity. The sum of flows of all incoming edges at every node needs to equal the sum of flows of all outgoing edges (flow conservation). Further, we constrain that for each node only a single incoming and a single outgoing flow vector may have a non-zero flow, thereby ensuring that incoming flow cannot be separated into multiple edges (node count constraint). To

fix the total number of flow entities leaving the source node, and entering the target node to a constant d we define the total count constraint. Overall, the resulting problem is a mixed-integer programming (MIP) problem that reads

$$\begin{aligned}
 & \underset{\substack{f_{uv} \in \mathbb{R}^k \\ b_{uv} \in \{0, 1\}}}{\text{maximize}} && \sum_{v:s \rightarrow v} f_{sv}^T \mathbf{1} \\
 & \text{subject to} && f_{uv} \leq b_{uv} c_{uv}, \quad \forall (u, v) \in E \text{ (capacity),} \\
 & && \sum_{u:u \rightarrow v} f_{uv} = \sum_{w:v \rightarrow w} f_{vw}, \quad \forall v \neq s, t \text{ (flow cons.),} \\
 & && \sum_{u:u \rightarrow v} b_{uv} = \sum_{w:v \rightarrow w} b_{vw} = 1, \quad \forall v \neq s, t \text{ (node count),} \\
 & && \sum_{u:s \rightarrow u} b_{su} = \sum_{v:v \rightarrow t} b_{vt} = d, \quad \forall u, v \neq s, t \text{ (total count),} \\
 & && f_{uv} \geq \mathbf{0} \quad \forall (u, v) \in E \text{ (non-neg.).}
 \end{aligned}$$

3.1. Application to Multi-Object Tracking

Similar to the scalar flow model for MOT [ZLN08], we define three different types of edges: (i) observation edges, (ii) transition edges, and (iii) enter/exit edges. We represent every detected object x_i in a frame by a vector and assign this vector to an observation edge (e_i, o_i) as a capacity vector. We show an example graph in Figure 2.

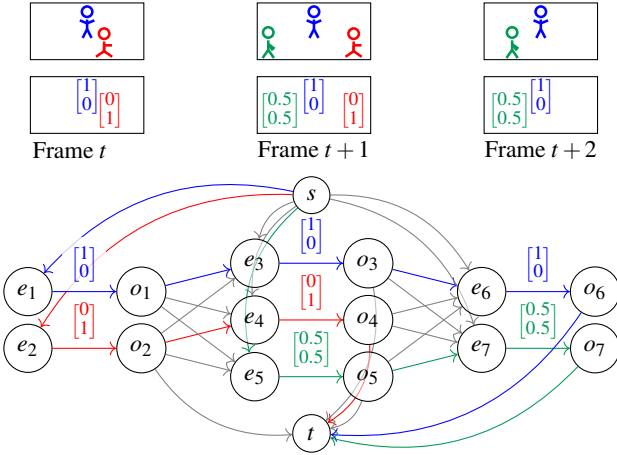


Figure 2: Graph construction: Objects of the sample frames (top row) are represented by feature vectors (middle row). These vectors are set as capacity on the corresponding object edges (bottom row).

By connecting the source to all nodes e_i and connecting all nodes o_i to the sink, objects can appear/disappear at any time (enter/exit nodes). Additionally, we connect all end nodes o_i with all start nodes e_j in a timespan Δt (transition edges), such that trajectories can skip the next Δt frames. Enter, exit, and transition edges are assigned infinite capacity. While the graph connections look similar to the scalar method [ZLN08], the capacities are set differently such that we can send vector-valued instead of scalar-valued flows.

4. Results

We choose training sequences (2,4,5,9,10 and 11) of the MOT16 Challenge dataset [MLTR*16] as a benchmark. We provide the

ground truth boxes and the ground truth number of individual objects to the algorithms such that we focus on the tracking rather than the detection part. For our experiments, we use two different feature descriptors: color histograms and deep features ([HZRS16]). To evaluate the robustness, we add random Gaussian noise with different variances to every image. We reduce the runtime by pruning and batch splitting. To evaluate our algorithm we use a metric that normalizes the identity switches (IDSW) by the total number of ground truth boxes (GT) for all frames t : $IDSW_{norm} = \frac{\sum_t IDSW_t}{\sum_t GT_t}$.

In Figure 3, we show that our algorithm performs substantially better on noisy data than the scalar baseline method [ZLN08] when using color features and deep features. Our method automatically allows to select (per object) feature dimensions that have the smallest variability across the entire sequence. The scalar method is not able to dynamically select features, since it computes a scalar score that summarizes feature similarities, and thus performs worse.

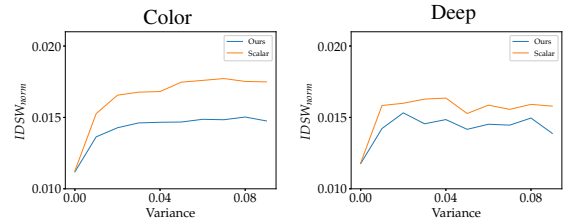


Figure 3: $IDSW_{norm}$ (\downarrow) for different noise values: Our method performs better on noisy images than the scalar method with different feature descriptors (left: color, right: deep features).

5. Conclusion

For the first time we conceptualized a non-separable multi-dimensional maximum flow formulation, and we demonstrated that such a formalism can naturally be applied to multi-object tracking. Since our flow is defined on a per-dimension basis, the maximizing flow automatically chooses the subset of feature dimensions that best match across a sequence.

References

- [EBA06] ERIKSSON A. P., BARR O., ASTROM K.: Image segmentation using minimal graph cuts. In *SSBA Symposium on Image Analysis* (2006). 1
- [GVY96] GARG N., VAZIRANI V. V., YANNAKAKIS M.: Approximate max-flow min-(multi) cut theorems and their applications. *SIAM Journal on Computing* 25 (1996). 1
- [HR55] HARRIS T., ROSS F.: *Fundamentals of a method for evaluating rail net capacities*. Tech. rep., RAND CORP SANTA MONICA CA, 1955. 1
- [HZRS16] HE K., ZHANG X., REN S., SUN J.: Deep residual learning for image recognition. In *CVPR* (2016). 2
- [LAB10] LI X., ANEJA Y. P., BAKI F.: An ant colony optimization metaheuristic for single-path multicommodity network flow problems. *Journal of the Operational Research Society* 61 (2010). 1
- [MLTR*16] MILAN A., LEAL-TAIXÉ L., REID I., ROTH S., SCHINDLER K.: Mot16: A benchmark for multi-object tracking. *arXiv preprint arXiv:1603.00831* (2016). 2
- [ZLN08] ZHANG L., LI Y., NEVATIA R.: Global data association for multi-object tracking using network flows. In *CVPR* (2008). 1, 2