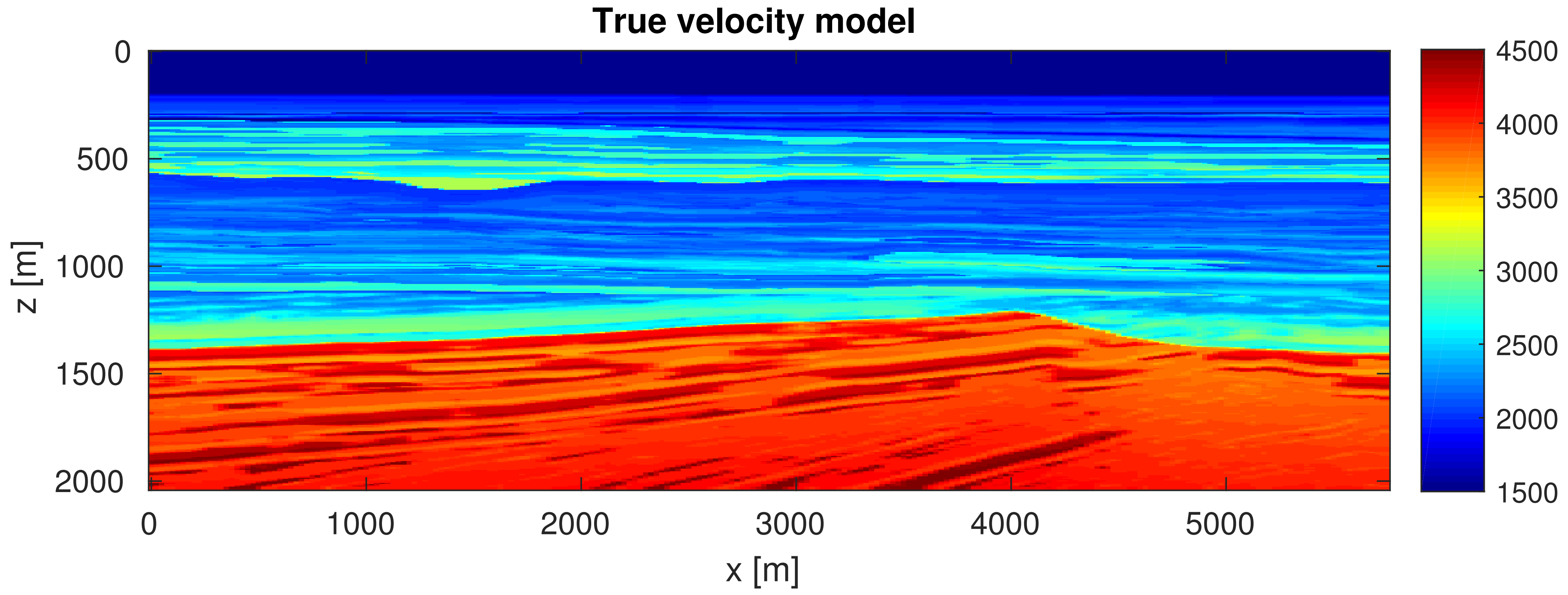
Generalized Minkowski sets for the regularization of inverse problems

Bas Peters (UBC) & Felix J. Herrmann (Georgia Tech)

SIAM Conference on Mathematical & Computational Issues in the Geosciences- March 2019
Practical Aspects of Large-scale Sparsity-promoting Seismic Inversion

SLIM
University of British Columbia

Prior information for geophysical models



- smoothness
- blockiness
- approximately layered media
- number of velocity jumps up or down
- maximum & minimum values, well-log information, reference models

Physical parameter estimation

$$\min_{m} f(m) \quad \text{s.t.} \quad m \in \bigcap_{i=1}^{p} \mathcal{V}_i$$

Geophysical applications:

- single $\mathcal{V}$ (bounds) **[Zeev et al. (2006) and Bello and Raydan (2007)]**
- two sets **[Lelièvre and Oldenburg (2009), Baumstein (2013), Smithyman et al. (2015), Esser et al. (2015ab, 2016ab), Peters and Herrmann (2017), Yong et al. (2018), Trinh et al. (2018), Peters and Herrmann (2019)]**

Physical parameter estimation

$$\min_{m} f(m) \quad \text{s.t.} \quad m \in \bigcap_{i=1}^{p} \mathcal{V}_i$$

Projection based algorithms: SPG, PQN, projected Newton-type guarantee that m satisfies *all* constraints, *every* iteration. **[Birgin et. al. (1999); Schmidt et. al. (2009); Schmidt et. al. (2012)]**

Physical parameter estimation

$$\min_{m} f(m) \quad \text{s.t.} \quad m \in \bigcap_{i=1}^{p} \mathcal{V}_i$$

Projection based algorithms: SPG, PQN, projected Newton-type guarantee that m satisfies *all* constraints, *every* iteration. **[Birgin et. al. (1999); Schmidt et. al. (2009); Schmidt et. al. (2012)]**

$$m^{k+1} = (1-\gamma)m^k - \gamma \mathcal{P}_{\mathcal{V}}(m^k - \beta \nabla_m f(m^k))$$

β : Barzilai-Borwein scaling

γ : non-monotone line search step length

Physical parameter estimation

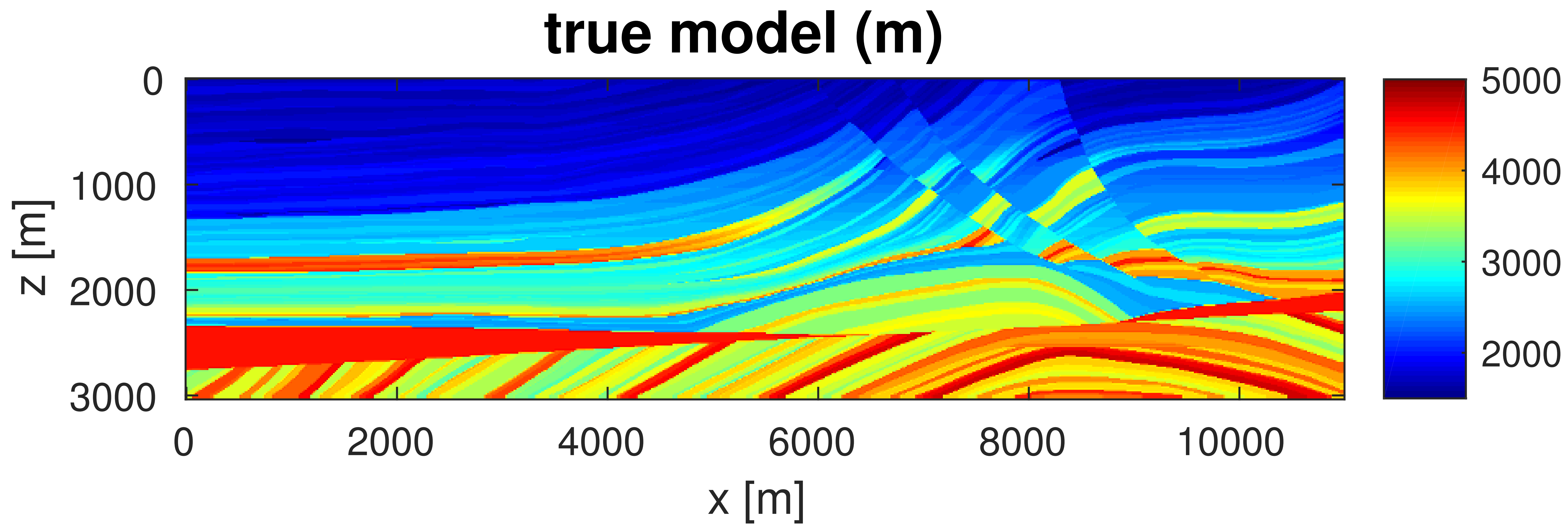
$$\min_{m} f(m) \quad \text{s.t.} \quad m \in \bigcap_{i=1}^{p} \mathcal{V}_i$$

Prior knowledge as constraint sets especially practical if we have many sets.

Each constraint set defined independently of all others.

Complex models

Models with smooth, blocky and diagonal features do not fit any of the standard constraints (rank, total-variation, smoothness promoting).



Question:

How to construct (convex) sets suitable to regularize this type of models?

Inspiration

cartoon-texture decomposition / morphological component analysis

[Osher et al. (2003); Starck et al. (2005); Schaeffer and Osher (2013); Ono et al. (2014)]

often stated as: $$\min_{u,v} \|m - u - v\| + \frac{\alpha}{2}\|Au\| + \frac{\beta}{2}\|Bv\|.$$

approximately decompose m into

1. u : cartoon/background/piecewise-smooth or constant component
2. v : texture/details/pattern/oscillatory component

closely related to to robust PCA and variants [Candes et al. (2011); Gao et al., 2011a ; Gao et al., 2011b]

Example

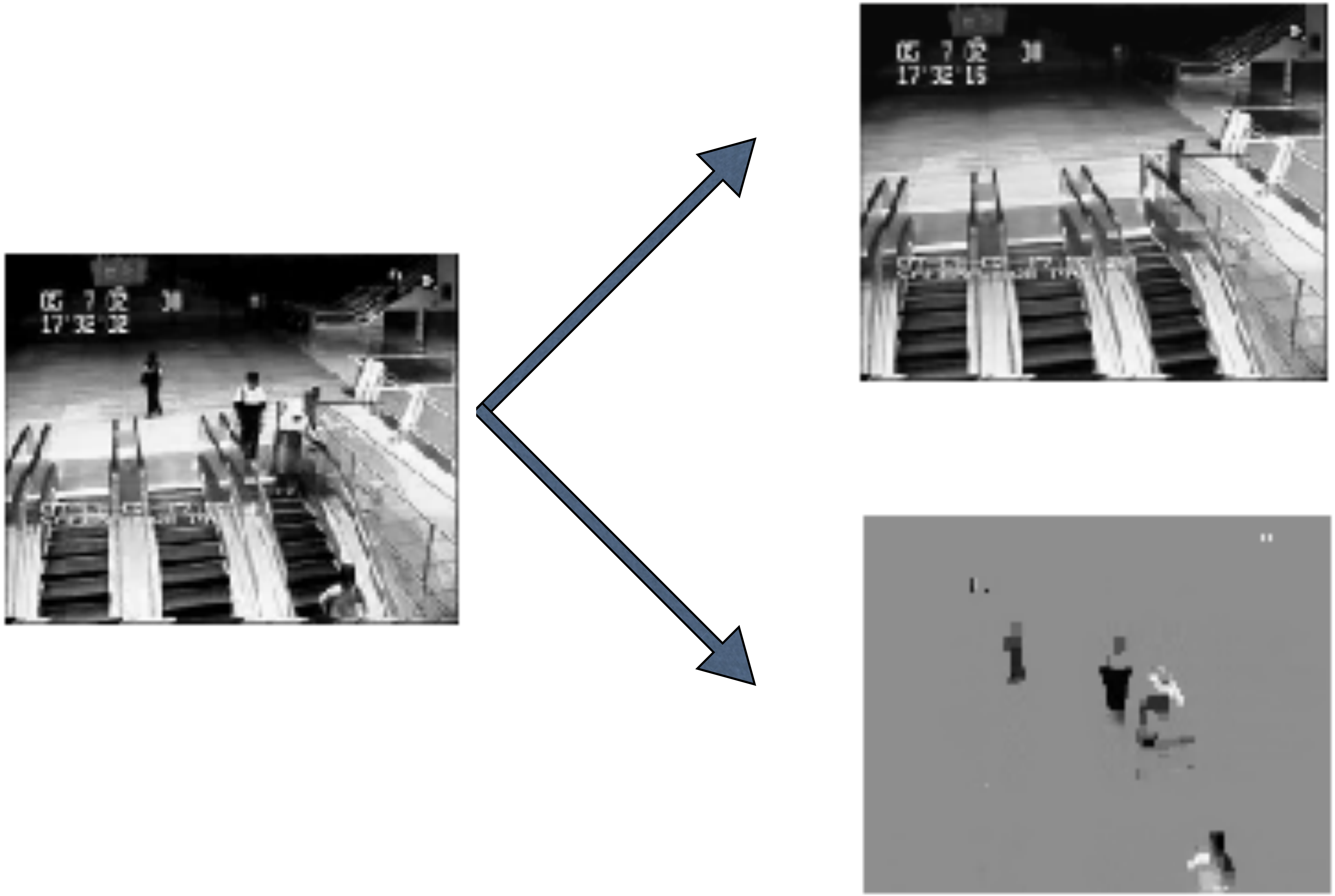
original image

cartoon part

texture part

[Szolgay D, & Szirányi T., 2012]

Example



Minkowski sum constraint sets

Idea:

- merge strengths of additive models and intersections of constraint sets
- represent a complex model as a sum of simple ones
- use different constraint on each part of the sum
- avoid penalties
- no explicit spatial segmentation, components may overlap

Minkowski sum constraint sets

Idea:

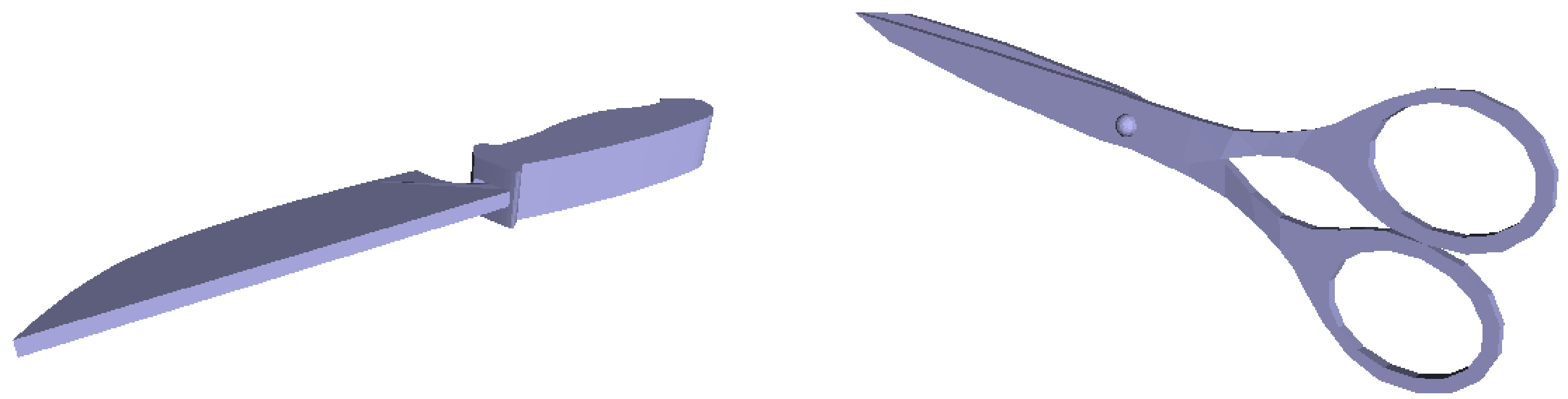
- merge strengths of additive models and intersections of constraint sets
- represent a complex model as a sum of simple ones
- use different constraint on each part of the sum
- avoid penalties
- no explicit spatial segmentation, components may overlap

The resulting model is an element of the Minkowski set (vector sum):

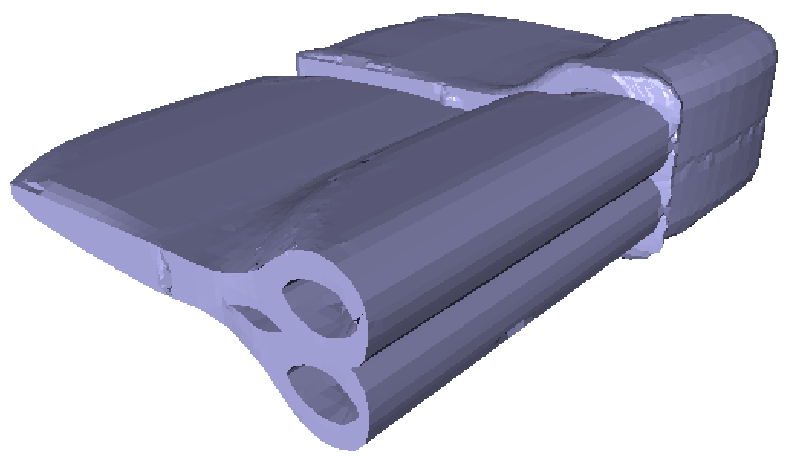
$$\mathcal{V} \equiv \mathcal{C}_1 + \mathcal{C}_1 = \{m = u + v \,|\, u \in \mathcal{C}_1, v \in \mathcal{C}_2\}$$

convex if both sets are convex

Minkowski sum

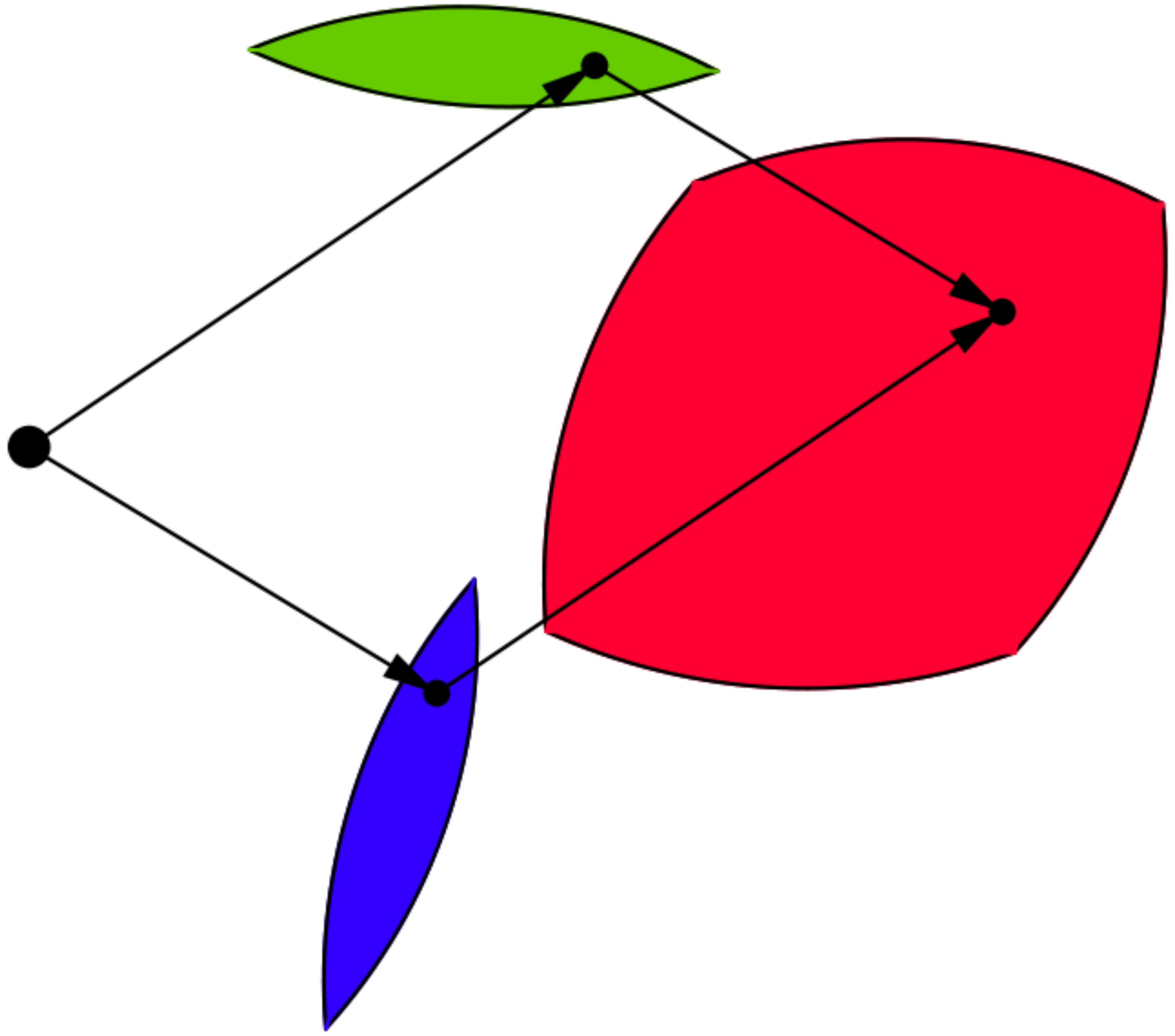


Knife (516 tris) Scissors (636 tris)



Knife ⊕ Scissors

Accurate Minkowski Sum Approximation of Polyhedral Models
Gokul Varadhan, Dinesh Manocha,
Pacific Graphics, 2004



https://en.wikipedia.org/wiki/Minkowski_addition

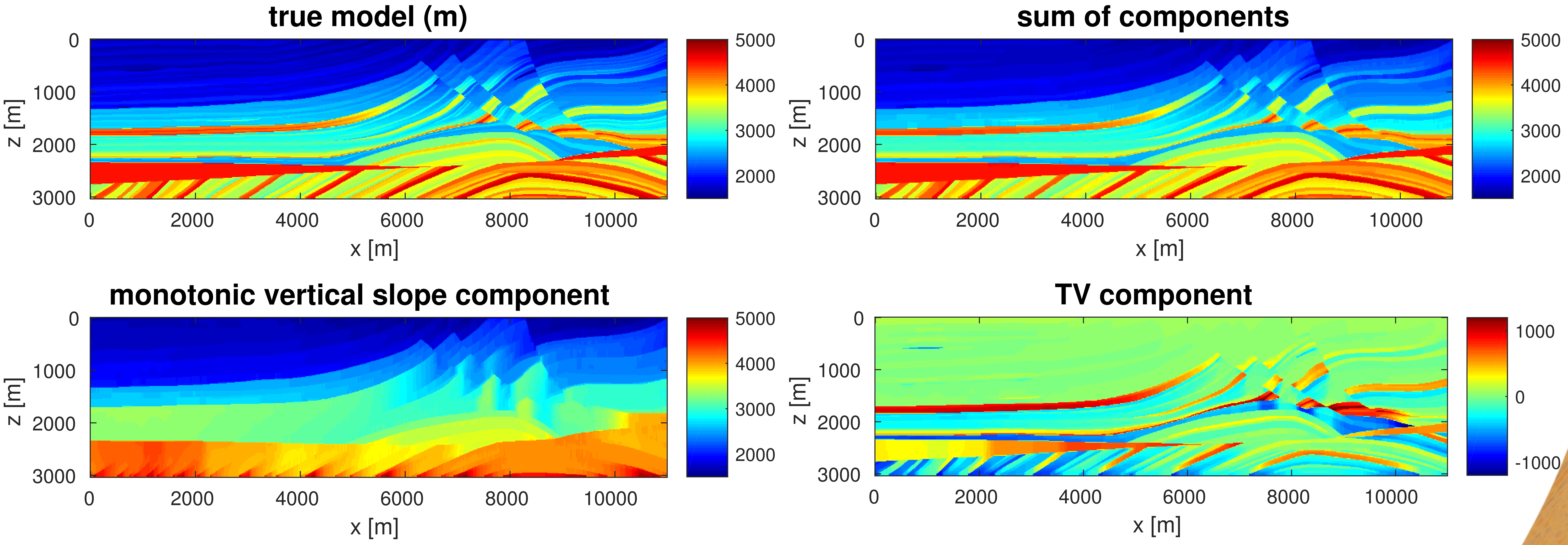
Minkowski sum constraint sets

$$\mathcal{V} \equiv \mathcal{C}_1 + \mathcal{C}_1 = \{m = u + v \,|\, u \in \mathcal{C}_1, v \in \mathcal{C}_2\}$$

Minkowski set not suitable by itself:

- need bound constraints and more on m
- would like more than one constraint on u and v

Minkowski decomposition



Definition 1: Generalized Minkowski set

$$\mathcal{M} \equiv \{m = u + v \,|\, u \in \bigcap_{i=1}^{p} \mathcal{D}_i,\; v \in \bigcap_{j=1}^{q} \mathcal{E}_j,\; m \in \bigcap_{k=1}^{r} \mathcal{F}_k\}$$

- m is an element of the intersection of two sets, one is the Minkowski set (sum of to intersections), the other is another intersection.
- can extend to more than two components

Definition 1: Generalized Minkowski set

$$\mathcal{M} \equiv \{m = u + v \,|\, u \in \bigcap_{i=1}^{p} \mathcal{D}_i,\; v \in \bigcap_{j=1}^{q} \mathcal{E}_j,\; m \in \bigcap_{k=1}^{r} \mathcal{F}_k\}$$

Proposition 1. *The generalized Minkowski set is convex if* $\mathcal{D}_i$, $\mathcal{E}_j$, *and* $\mathcal{F}_k$ *are convex sets for all* i, j *and* k.

Proof. *It follows from the definition (almost)*

Proposal: Generalized Minkowski set

SLIM

$$\mathcal{M} \equiv \{m = u + v \mid u \in \bigcap_{i=1}^{p} \mathcal{D}_i,\; v \in \bigcap_{j=1}^{q} \mathcal{E}_j,\; m \in \bigcap_{k=1}^{r} \mathcal{F}_k\}$$

Projection:

$$\operatorname*{arg\,min}_{u,v,w} \frac{1}{2}\|w-m\|_2^2 + \sum_{i=1}^{p} \iota_{\mathcal{D}_i}(A_i u) + \sum_{j=1}^{q} \iota_{\mathcal{E}_j}(B_j v) + \sum_{k=1}^{r} \iota_{\mathcal{F}_k}(C_k w) + \iota_{w=u+v}(w, u, v)$$

Proposal: Generalized Minkowski set

$$\mathcal{M} \equiv \{m = u + v \mid u \in \bigcap_{i=1}^{p} \mathcal{D}_i,\ v \in \bigcap_{j=1}^{q} \mathcal{E}_j,\ m \in \bigcap_{k=1}^{r} \mathcal{F}_k\}$$

Projection:

$$\underset{u,v,w}{\arg\min}\ \frac{1}{2}\|w-m\|_2^2+\sum_{i=1}^{p}\iota_{\mathcal{D}_i}(A_i u)+\sum_{j=1}^{q}\iota_{\mathcal{E}_j}(B_j v)+\sum_{k=1}^{r}\iota_{\mathcal{F}_k}(C_k w)+\iota_{w=u+v}(w,u,v)$$

discrete derivatives matrices, DFT, Wavelet T, ...

New algorithm (1)

Goals:

Construct an algorithm to project onto an intersection

- allow non-orthogonal linear operators in set definitions
- exploit similarity between sets
- use coarse and fine-grained parallel resources

New algorithm (1)

Goals:

Construct an algorithm to project onto an intersection

- allow non-orthogonal linear operators in set definitions
- exploit similarity between sets
- use coarse and fine-grained parallel resources

[Afonso et. al., 2011], [Combettes & Pesquet, 2011 ; Kitic et. al. 2016]

Merge ideas from SALSA/SDMM and ARADMM

- recast as known algorithm for known problem → convergence guarantees
- automatic (acceleration) parameter selection [Xu et. al. ,2016a ; Xu et. al. ,2017]

New algorithm (2)

Reformulate projection onto an intersection:

$$\min_x \frac{1}{2}\|x - m\|_2^2 + \sum_{i=1}^{p} \iota_{\mathcal{C}_i}(A_i x)$$

New algorithm (2)

Reformulate projection onto an intersection:

$$\min_{x} \frac{1}{2}\|x - m\|_2^2 + \sum_{i=1}^{p} \iota_{\mathcal{C}_i}(A_i x)$$

Split indicators and linear operators:

$$\min_{x,\{y_i\}} \frac{1}{2}\|x - m\|_2^2 + \sum_{i=1}^{p} \iota_{\mathcal{C}_i}(y_i) \quad \text{s.t.} \quad A_i x = y_i$$

New algorithm (2)

Reformulate projection onto an intersection:

$$\min_{x} \frac{1}{2}\|x - m\|_2^2 + \sum_{i=1}^{p} \iota_{\mathcal{C}_i}(A_i x)$$

Split indicators and linear operators:

$$\min_{x,\{y_i\}} \frac{1}{2}\|x - m\|_2^2 + \sum_{i=1}^{p} \iota_{\mathcal{C}_i}(y_i) \quad \text{s.t.} \quad A_i x = y_i$$

put smooth and non-smooth terms on same footing

$$\frac{1}{2}\|x - m\|_2^2 \equiv f(y_{p+1})$$

$$\tilde{f}(\tilde{y}) = f(y_{p+1}) + \sum_{i=1}^{p} \iota_{\mathcal{C}_i}(y_i)$$

SLIM

New algorithm (3)

Define matrix and vectors: $$\tilde{A} = \begin{pmatrix} A_1 \\ \vdots \\ A_{p+1} = I_N \end{pmatrix}, \quad \tilde{y} = \begin{pmatrix} y_1 \\ \vdots \\ y_{p+1} \end{pmatrix}, \quad \tilde{v} = \begin{pmatrix} v_1 \\ \vdots \\ v_{p+1} \end{pmatrix}$$

Final problem formulation: $$\min_{x,\tilde{y}} \tilde{f}(\tilde{y}) \quad \text{s.t.} \quad \tilde{A}x = \tilde{y}$$

New algorithm (3)

Define matrix and vectors:

$$\tilde{A} = \begin{pmatrix} A_1 \\ \vdots \\ A_{p+1} = I_N \end{pmatrix}, \quad \tilde{y} = \begin{pmatrix} y_1 \\ \vdots \\ y_{p+1} \end{pmatrix}, \quad \tilde{v} = \begin{pmatrix} v_1 \\ \vdots \\ v_{p+1} \end{pmatrix}$$

Final problem formulation:

$$\min_{x,\tilde{y}} \tilde{f}(\tilde{y}) \quad \text{s.t.} \quad \tilde{A}x = \tilde{y}$$

Augmented Lagrangian:

$$L_{\rho_1,\ldots,\rho_{p+1}}(x, y_1, \ldots, y_{p+1}, v_1, \ldots, v_{p+1}) = \sum_{i=1}^{p+1} \left[f_i(y_i) + v_i^\top (y_i - A_i x) + \frac{\rho_i}{2} \|y_i - A_i x\|_2^2 \right]$$

New algorithm (4)

Iterations for our problem: (equivalent to SDMM + over/under relaxation)

$$
\begin{aligned}
x^{k+1} &= \Big[\sum_{i=1}^{p}(\rho_i^k A_i^\top A_i) + \rho_{p+1}^k I_N\Big]^{-1} \sum_{i=1}^{p+1}\Big[A_i^\top(\rho_i^k y_i^k + v_i^k)\Big] \\
\bar{x}_i^{k+1} &= \gamma_i^k A_i x_i^{k+1} + (1-\gamma_i^k) y_i^k \\
y_i^{k+1} &= \operatorname{prox}_{f_i,\rho_i^k}(\bar{x}_i^{k+1} - \frac{v_i^k}{\rho_i^k}) \\
v_i^{k+1} &= v_i^k + \rho_i^k(y_i^{k+1} - \bar{x}_i^{k+1}).
\end{aligned}
$$

SLIM

New algorithm (5)

PARSDMM:

projection adaptive-relaxed simultaneous direction method of multipliers

Iterations for our problem: (equivalent to SDMM + over/under relaxation)

$$x^{k+1} = \Big[\sum_{i=1}^{p} (\rho_i^k A_i^\top A_i) + \rho_{p+1}^k I_N \Big]^{-1} \sum_{i=1}^{p+1} \Big[A_i^\top (\rho_i^k y_i^k + v_i^k) \Big] \longrightarrow \text{warm-start CG}$$

$$\bar{x}_i^{k+1} = \gamma_i^k A_i x_i^{k+1} + (1 - \gamma_i^k) y_i^k$$

$$y_i^{k+1} = \operatorname{prox}_{f_i, \rho_i^k} (\bar{x}_i^{k+1} - \frac{v_i^k}{\rho_i^k})$$

$$v_i^{k+1} = v_i^k + \rho_i^k (y_i^{k+1} - \bar{x}_i^{k+1}).$$

New algorithm

PARSDMM:

projection adaptive-relaxed simultaneous direction method of multipliers

Iterations for our problem: (equivalent to SDMM + over/under relaxation)

$$
\begin{aligned}
x^{k+1} &= \Big[\sum_{i=1}^{p} (\rho_i^k A_i^\top A_i) + \rho_{p+1}^k I_N \Big]^{-1} \sum_{i=1}^{p+1} \Big[A_i^\top (\rho_i^k y_i^k + v_i^k) \Big] \\
\bar{x}_i^{k+1} &= \gamma_i^k A_i x_i^{k+1} + (1 - \gamma_i^k) y_i^k \\
y_i^{k+1} &= \operatorname{prox}_{f_i, \rho_i^k} (\bar{x}_i^{k+1} - \frac{v_i^k}{\rho_i^k}) \\
v_i^{k+1} &= v_i^k + \rho_i^k (y_i^{k+1} - \bar{x}_i^{k+1}).
\end{aligned}
$$

system-mat always pos-def,

New algorithm

PARSDMM:

projection adaptive-relaxed simultaneous direction method of multipliers

Iterations for our problem: (equivalent to SDMM + over/under relaxation)

$$x^{k+1} = \Big[\sum_{i=1}^{p} (\rho_i^k A_i^\top A_i) + \rho_{p+1}^k I_N \Big]^{-1} \sum_{i=1}^{p+1} \Big[A_i^\top (\rho_i^k y_i^k + v_i^k) \Big]$$

$$\bar{x}_i^{k+1} = \gamma_i^k A_i x_i^{k+1} + (1 - \gamma_i^k) y_i^k$$

$$y_i^{k+1} = \text{prox}_{f_i, \rho_i^k} (\bar{x}_i^{k+1} - \frac{v_i^k}{\rho_i^k})$$

$$v_i^{k+1} = v_i^k + \rho_i^k (y_i^{k+1} - \bar{x}_i^{k+1}).$$

we can decide on how many CG iterations we need for the sub-problem

New algorithm

PARSDMM:

projection adaptive-relaxed simultaneous direction method of multipliers

Iterations for our problem: (equivalent to SDMM + over/under relaxation)

$$x^{k+1} = \Big[\sum_{i=1}^{p} (\rho_i^k A_i^\top A_i) + \rho_{p+1}^k I_N \Big]^{-1} \sum_{i=1}^{p+1} \Big[A_i^\top (\rho_i^k y_i^k + v_i^k) \Big]$$

$$\bar{x}_i^{k+1} = \gamma_i^k A_i x_i^{k+1} + (1 - \gamma_i^k) y_i^k$$

$$y_i^{k+1} = \text{prox}_{f_i, \rho_i^k} (\bar{x}_i^{k+1} - \frac{v_i^k}{\rho_i^k})$$

$$v_i^{k+1} = v_i^k + \rho_i^k (y_i^{k+1} - \bar{x}_i^{k+1}).$$

in parallel for every index i

New algorithm

PARSDMM:

projection adaptive-relaxed simultaneous direction method of multipliers

Iterations for our problem: (equivalent to SDMM + over/under relaxation)

$$x^{k+1} = \Big[\sum_{i=1}^{p}(\rho_i^k A_i^\top A_i) + \rho_{p+1}^k I_N\Big]^{-1} \sum_{i=1}^{p+1} \Big[A_i^\top(\rho_i^k y_i^k + v_i^k)\Big]$$

$$\bar{x}_i^{k+1} = \gamma_i^k A_i x_i^{k+1} + (1-\gamma_i^k) y_i^k \longrightarrow$$ over/under relaxation

$$y_i^{k+1} = \text{prox}_{f_i,\rho_i^k}(\bar{x}_i^{k+1} - \frac{v_i^k}{\rho_i^k})$$

$$v_i^{k+1} = v_i^k + \rho_i^k(y_i^{k+1} - \bar{x}_i^{k+1}).$$

SLIM

New algorithm

PARSDMM:

projection adaptive-relaxed simultaneous direction method of multipliers

Iterations for our problem: (equivalent to SDMM + over/under relaxation)

$$
\begin{aligned}
x^{k+1} &= \Big[\sum_{i=1}^{p} (\rho_i^k A_i^\top A_i) + \rho_{p+1}^k I_N \Big]^{-1} \sum_{i=1}^{p+1} \Big[A_i^\top (\rho_i^k y_i^k + v_i^k) \Big] \\
\bar{x}_i^{k+1} &= \gamma_i^k A_i x_i^{k+1} + (1 - \gamma_i^k) y_i^k \\
y_i^{k+1} &= \operatorname{prox}_{f_i, \rho_i^k} (\bar{x}_i^{k+1} - \frac{v_i^k}{\rho_i^k}) \\
v_i^{k+1} &= v_i^k + \rho_i^k (y_i^{k+1} - \bar{x}_i^{k+1}).
\end{aligned}
$$

simple projection onto set:
norm-ball/bounds/cardinality/rank
(all closed-form solutions)

i=p: prox for distance-squared
(closed form)

Iterations are just iterations…

For a fast algorithms we also need:

- stopping conditions
- adaptive parameter selection
- hybrid coarse-fine parallel implementation
- multilevel acceleration
- use multithreaded compressed diagonal MVPs for banded matrices

- couple more things…

Projections onto the generalized Minkowski set

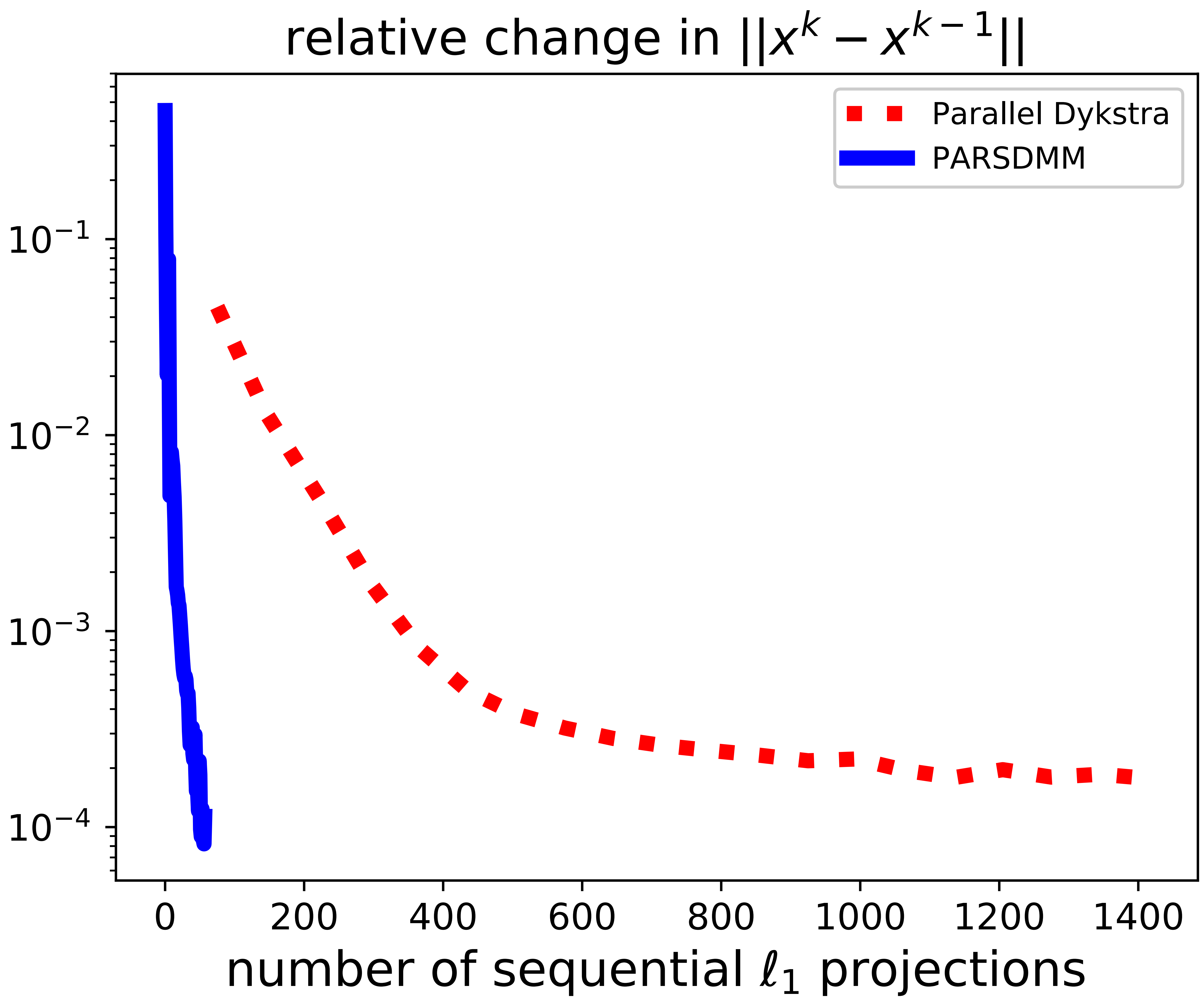
$$\mathcal{M} \equiv \{m = u + v \,|\, u \in \bigcap_{i=1}^{p} \mathcal{D}_i, \; v \in \bigcap_{j=1}^{q} \mathcal{E}_j, \; m \in \bigcap_{k=1}^{r} \mathcal{F}_k\}$$

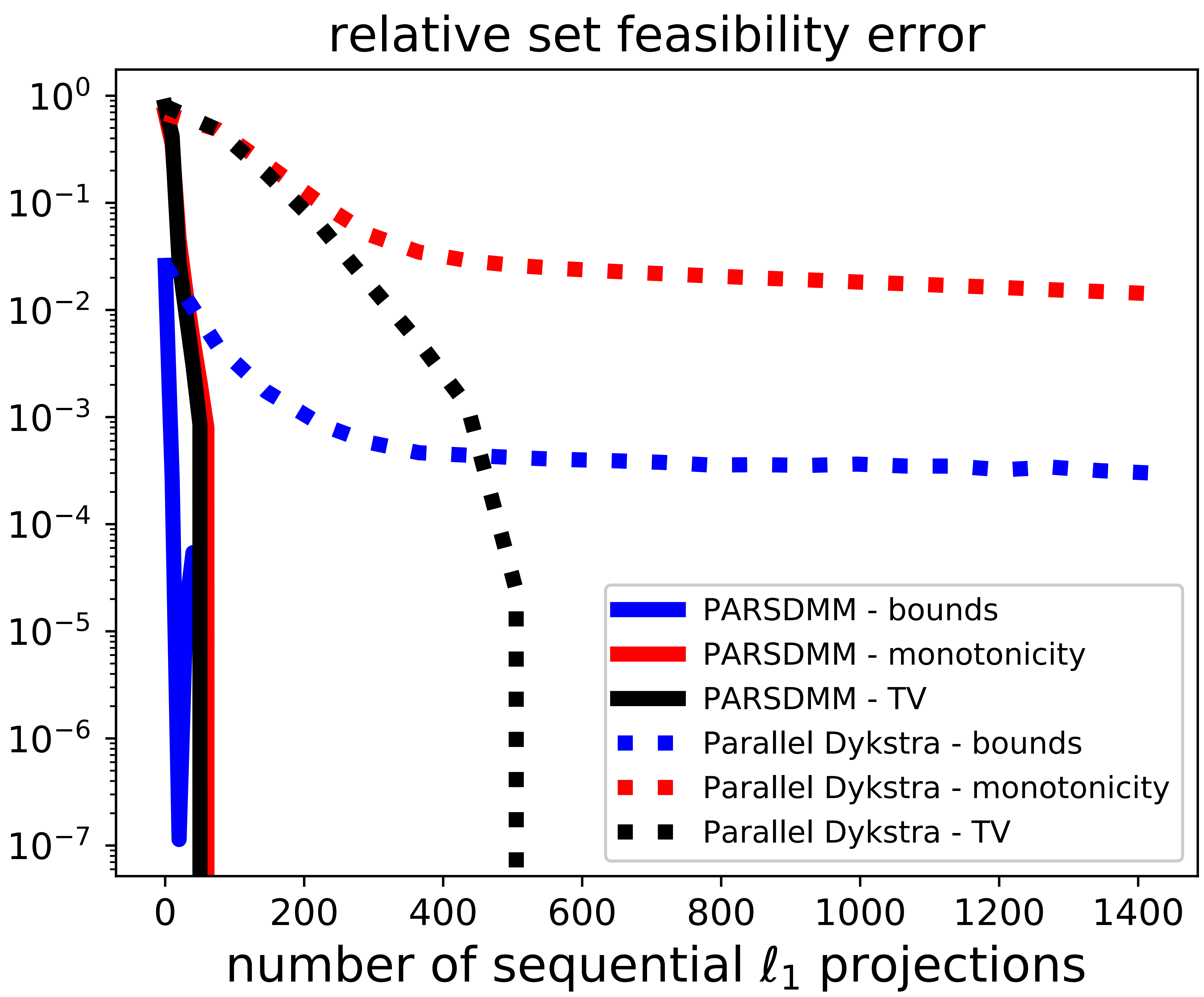
Projection:

$$\underset{u,v,w}{\arg\min} \frac{1}{2}\|w-m\|_2^2 + \sum_{i=1}^{p} \iota_{\mathcal{D}_i}(A_i u) + \sum_{j=1}^{q} \iota_{\mathcal{E}_j}(B_j v) + \sum_{k=1}^{r} \iota_{\mathcal{F}_k}(C_k w) + \iota_{w=u+v}(w, u, v)$$

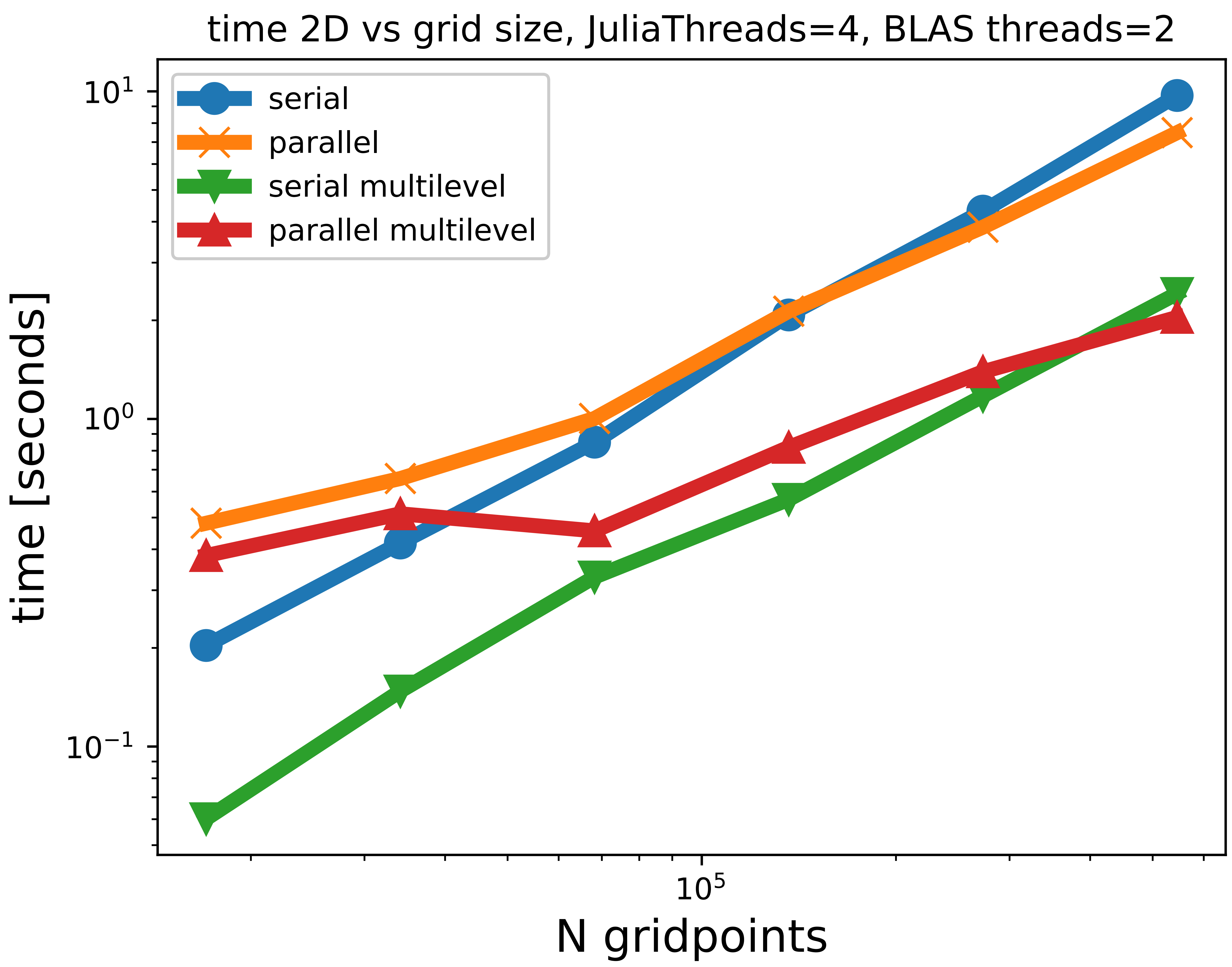
- follow same recipe as for intersections
- matrices -> block-structured linear systems
- same algorithm in the end, different inputs

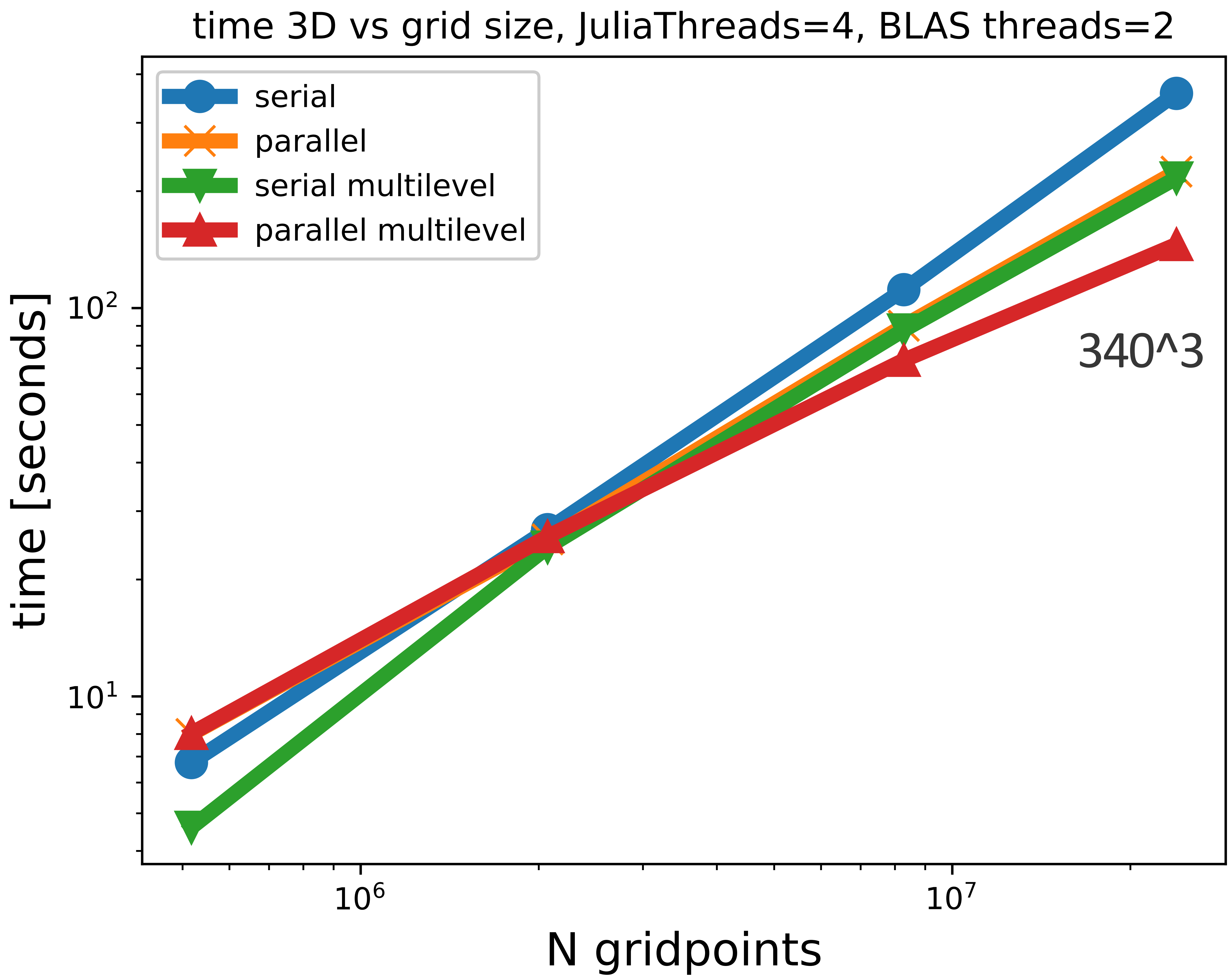
Parallel Dykstra-ADMM vs PARSDMM





Bounds & lateral smoothness & vertical monotonicity constraints





Example: video segmentation

Decompose video $T \in \mathbb{R}^{n_x \times n_y \times n_t}$ into background and anomaly

Proposed constrained formulation:

$$\min_x \frac{1}{2}\|x - \text{vec}(T)\|_2^2 \text{ s.t. } x \in \mathcal{F}_1 \bigcap \left(\bigcap_{i=1}^{2} \mathcal{D}_i + \bigcap_{j=1}^{4} \mathcal{E}_j\right) \qquad x = \begin{pmatrix} u \\ v \end{pmatrix}$$

- constraints on tensor
- simultaneously apply constraints to time-frames
- simultaneously apply constraints to other tensor slices

Example: video segmentation

$$\min_{x} \frac{1}{2}\|x - \mathrm{vec}(T)\|_2^2 \ \text{ s.t. } \ x \in \mathcal{F}_1 \bigcap \Big(\bigcap_{i=1}^{2} \mathcal{D}_i + \bigcap_{j=1}^{4} \mathcal{E}_j \Big) \qquad x = \begin{pmatrix} u \\ v \end{pmatrix}$$

Constraints on background ($\mathcal{D}$) :

- every time slice is an element of the subspace spanned by the last 20 frames (there are no people in those)
- bounds per fiber along time-axis

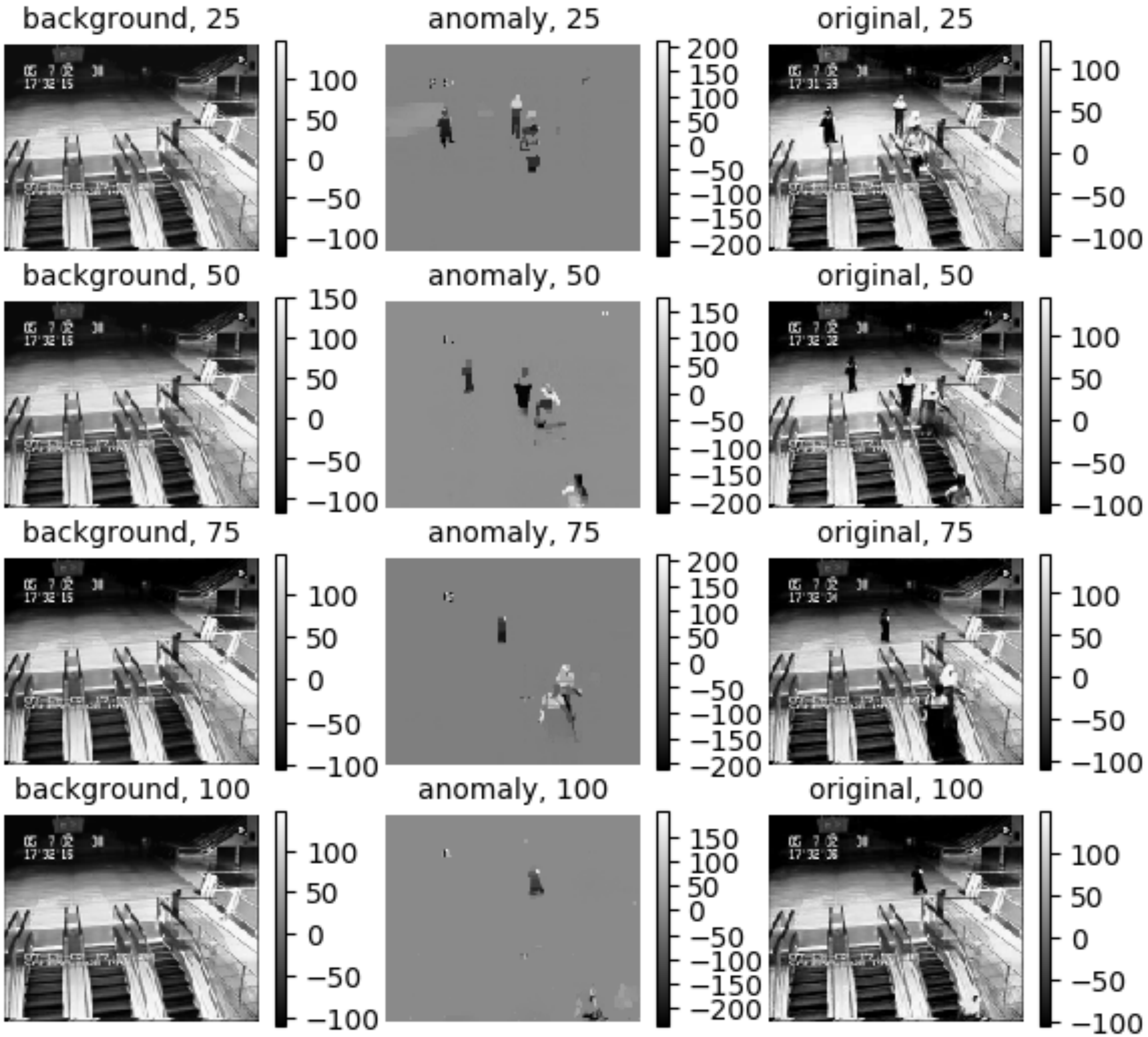
Constraints on sum ($\mathcal{F}$) :

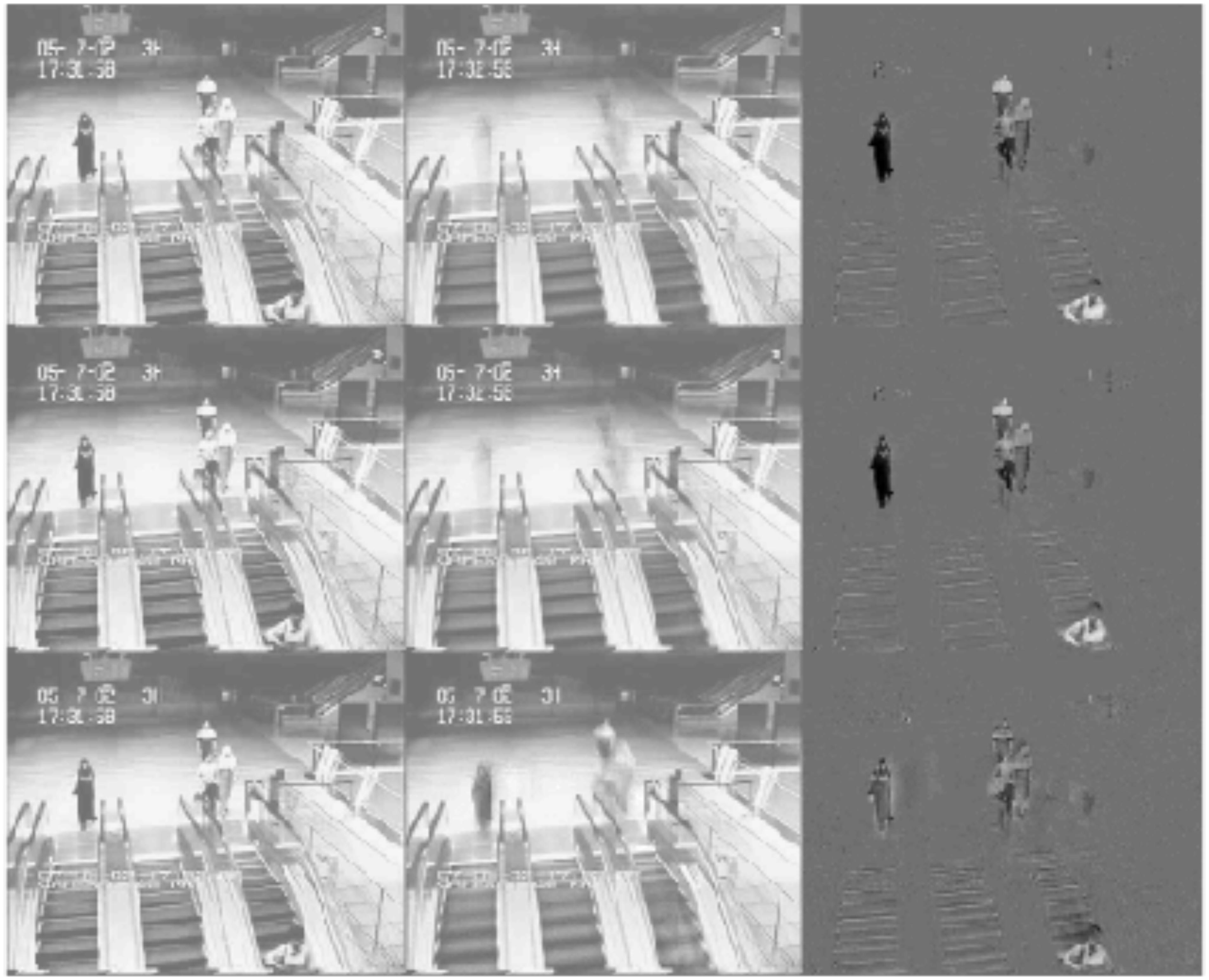
- bounds (grayscale)

Constraints on anomaly ($\mathcal{E}$) :

- bounds on sum minus bounds on background
- cardinality on each time-frame and on horizontal, vertical derivative

Example:


background, 25
anomaly, 25
original, 25
background, 50
anomaly, 50
original, 50
background, 75
anomaly, 75
original, 75
background, 100
anomaly, 100
original, 100



(a) X,L, and S matrices found by (from top to bottom) Split-SPCP, LagQN, and GoDec.

(b) X,L, and S matrices found by (from top to bottom) FPCP, IALM, and LMaFit.

[Comparison of RPCA algorithms by Driggs et al., 2017]

Conclusions

Intersections of sets are particularly suitable in case of many constraint sets.

Generalized Minkowski sets allow for more convenient use of prior info.

Dedicated algorithm suitable for 2D and larger 3D videos and models.

A larger number of constraint sets does not increase computational cost much.

Code: SetIntersectionProjection.jl

Software available for julia

Algorithms and software for projections onto intersections of convex and non-convex sets with applications to inverse problems

[B. Peters & F.J. Herrmann (2019) , arXiv preprint arXiv:1902.09699]

https://github.com/slimgroup/SetIntersectionProjection.jl/

Generalized Minkowski sets for the regularization of inverse problems

[B. Peters & F.J. Herrmann (2019) , arXiv preprint arXiv:1903.03942]

https://petersbas.github.io/GeneralizedMinkowskiSetDocs/