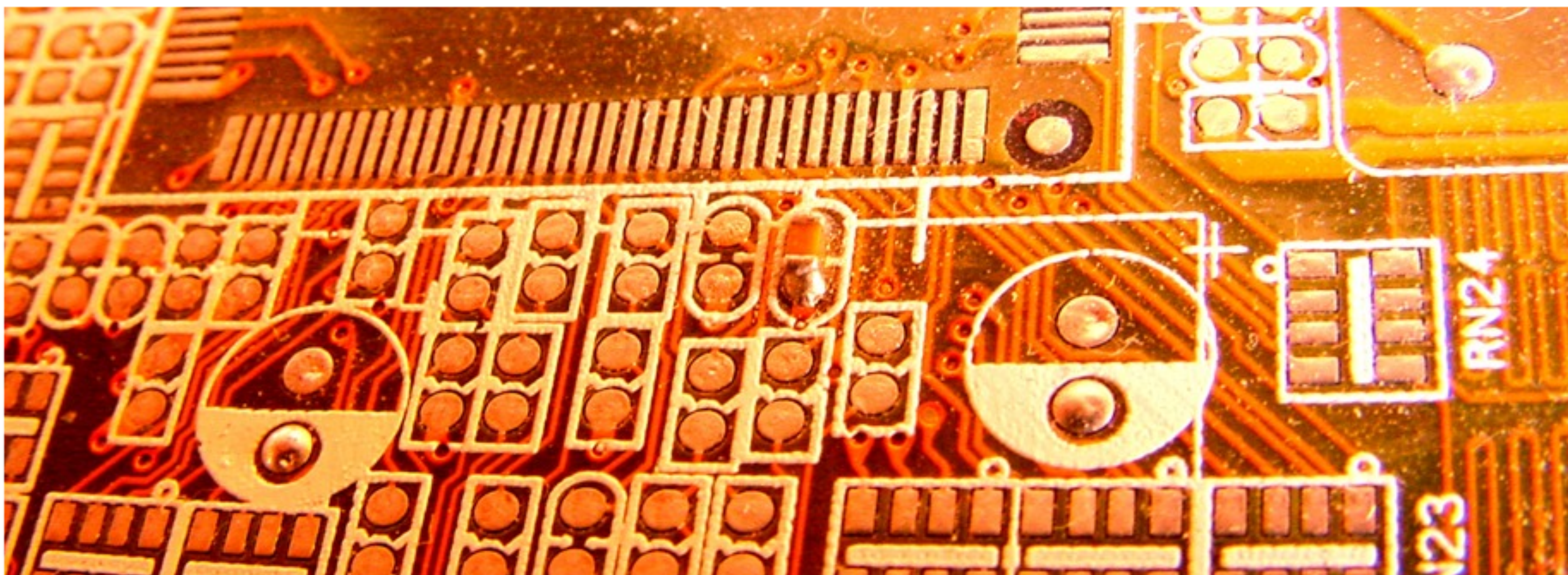
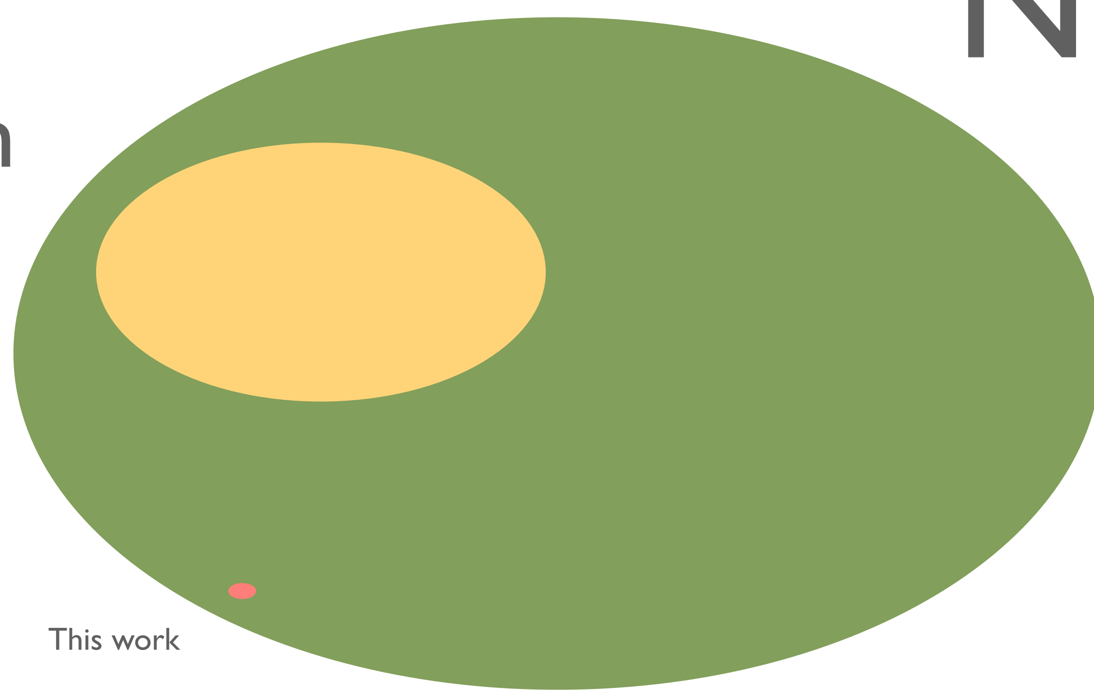
Signal Processing beyond the Gaussian Assumption

Ricardo Silveira Cabral



But really...



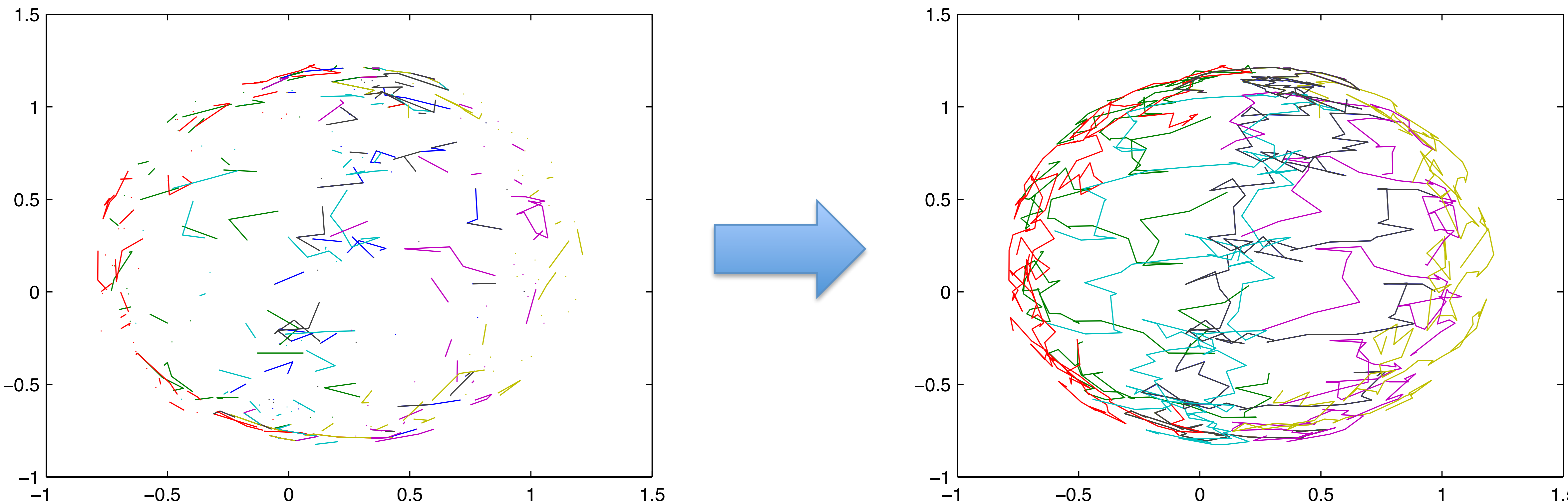
Fast Incremental Matrix Completion:
an application to trajectory correction

Motivation

- Gaussian Estimation
  - Kalman Filter
  - Extended Kalman Filter
  - Uscented Kalman Filter
- Non-Gaussian Estimation
  - Weighted Median Filter
  - Weighted Myriad Filter
- Partially unknown measurements?

Motivation

- 3D Reconstruction: Points seen in various 2D images
- Plagued by self-occlusion & tracker errors



The goal

$$\mathbf{W}^{(0)} = \begin{bmatrix} 0.2 & ? & 0.9 & ? & 1 \\ 0.7 & ? & 0.3 & ? & 0.1 \\ ? & 0.3 & ? & 0.9 & ? \\ ? & 0.1 & ? & 0.1 & ? \\ 0.5 & 0.3 & 0.8 & 0.5 & 0 \\ 0.1 & 0.9 & 0.2 & 0.8 & 1 \\ & & \vdots & & \end{bmatrix} \quad \mathbf{W}^{(1)} = \begin{bmatrix} 0.2 & ? & 0.9 & ? & 1 \\ 0.7 & ? & 0.3 & ? & 0.1 \\ 0.4 & 0.3 & 0.5 & 0.9 & 0.8 \\ 0.6 & 0.1 & 0.5 & 0.1 & 0.1 \\ 0.5 & 0.3 & 0.8 & 0.5 & 0 \\ 0.1 & 0.9 & 0.2 & 0.8 & 1 \\ & & \vdots & & \end{bmatrix} \quad \mathbf{W}^{(2)} = \begin{bmatrix} 0.2 & 1 & 0.9 & 0.3 & 1 \\ 0.7 & 0.2 & 0.3 & 0.5 & 0.1 \\ 0.4 & 0.3 & 0.5 & 0.9 & 0.8 \\ 0.6 & 0.1 & 0.5 & 0.1 & 0.1 \\ 0.5 & 0.3 & 0.8 & 0.5 & 0 \\ 0.1 & 0.9 & 0.2 & 0.8 & 1 \\ & & \vdots & & \end{bmatrix}$$

- Low-rank (rank 4)
- Gross, but sparse outliers
- Fast, incremental

State of the art

- Nuclear Norm formulation
  [Cai et al. 2008; Candès and Recht 2008;
  Lin et al. 2009; Toh and Yun 2009; Ji and Ye 2009]

$$\begin{array}{ll} \text{minimize} & ||\mathbf{A}||_* \\ \text{subject to} & \mathbf{D}_{ij} = \mathbf{A}_{ij}, \quad \forall (i,j) \in \Omega. \end{array}$$

- Interesting theoretical properties
- Slow and no incremental version

State of the art

- SPectrally Optimal Completion [Aguiar et al. 2008]

$$\mathbf{W} = \begin{bmatrix} x & \mathbf{v}^\top \\ \mathbf{u} & \mathbf{A} \end{bmatrix}, \quad \hat{\sigma}_5 = \max\left\{\sigma_5\left(\begin{bmatrix} \mathbf{v}^\top \\ \mathbf{A} \end{bmatrix}\right), \sigma_5\left(\begin{bmatrix} \mathbf{u} & \mathbf{A} \end{bmatrix}\right)\right\}$$

$$p(x) = \det(\mathbf{W}\mathbf{W}^\top - \hat{\sigma}_5^2\mathbf{I}) = 0$$

- Optimal in absence of noise
- Matrix needs to be permuted into Young Diagram

Proposed approach

- To build upon SPOC we need:
  - Estimate permutation matrices

$$\begin{bmatrix} ? & 0.3 & 0.8 & ? & 0.5 \\ ? & 0.9 & 0.2 & ? & 0.1 \\ ? & 0.7 & 0.9 & 0.5 & 1 \\ ? & 0.2 & 0.3 & 0.8 & 0.1 \\ 0.1 & 0.3 & 0.4 & 0.9 & 0.5 \\ 0.9 & 0.1 & 0.2 & 0.1 & 0.3 \end{bmatrix} \Rightarrow \begin{bmatrix} ? & ? & 0.3 & 0.8 & 0.5 \\ ? & ? & 0.9 & 0.2 & 0.1 \\ ? & 0.5 & 0.7 & 0.9 & 1 \\ ? & 0.8 & 0.2 & 0.3 & 0.1 \\ 0.1 & 0.9 & 0.3 & 0.4 & 0.5 \\ 0.9 & 0.1 & 0.1 & 0.2 & 0.3 \end{bmatrix}$$

  - Add robustness to outliers

$$\begin{bmatrix} ? & ? & 0.3 & 0.8 & 0.5 \\ ? & ? & 0.9 & 0.2 & 0.1 \\ ? & 0.5 & 0.7 & 0.9 & 1 \\ ? & 17 & 0.2 & 0.3 & 0.1 \\ 0.1 & 0.9 & 0.3 & 0.4 & 0.5 \\ 0.9 & 0.1 & 0.1 & 0.2 & 0.3 \end{bmatrix} \Rightarrow \begin{bmatrix} ? & ? & 0.3 & 0.8 & 0.5 \\ ? & ? & 0.9 & 0.2 & 0.1 \\ ? & 0.5 & 0.7 & 0.9 & 1 \\ ? & 0.8 & 0.2 & 0.3 & 0.1 \\ 0.1 & 0.9 & 0.3 & 0.4 & 0.5 \\ 0.9 & 0.1 & 0.1 & 0.2 & 0.3 \end{bmatrix}$$

Proposed approach

- Finding Permutations
- Complexity O(N)

Algorithm 1 Finding permutations for SPOC

Initialize known count $k = 0$
Initialize unknown count $u = 0$
for all points $i \in 1 : N$ **do**
  if point i is known **then**
    Increment k
    $P_{i,k} = 1$
  else
    $P_{i,N-u} = 1$
    Increment u
  end if
end for

$$\mathbf{W} = \begin{bmatrix} ??? & 0.3 & ??? & 0.8 & ??? & 0.5 \\ ??? & 0.9 & ??? & 0.2 & ??? & 0.1 \end{bmatrix}$$

$$\mathbf{WP} = \begin{bmatrix} 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \end{bmatrix}$$

$$\mathbf{P} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Proposed approach

- Finding Permutations
- Complexity O(N)

Algorithm 1 Finding permutations for SPOC

```
Initialize known count k = 0
Initialize unknown count u = 0
for all points i ∈ 1 : N do
    if point i is known then
        Increment k
        P_{i,k} = 1
    else
        P_{i,N−u} = 1
        Increment u
    end if
end for
```

$$\mathbf{W} = \begin{bmatrix} ??? & 0.3 & ??? & 0.8 & ??? & 0.5 \\ ??? & 0.9 & ??? & 0.2 & ??? & 0.1 \end{bmatrix}$$

$$\mathbf{WP} = \begin{bmatrix} ??? & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ ??? & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \end{bmatrix}$$

$$\mathbf{P} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Proposed approach

- Finding Permutations
- Complexity O(N)

Algorithm 1 Finding permutations for SPOC

Initialize known count $k = 0$
Initialize unknown count $u = 0$
for all points $i \in 1 : N$ **do**
  if point i is known **then**
    Increment k
    $P_{i,k} = 1$
  else
    $P_{i,N-u} = 1$
    Increment u
  end if
end for

$$\mathbf{W} = \begin{bmatrix} ??? & 0.3 & ??? & 0.8 & ??? & 0.5 \\ ??? & 0.9 & ??? & 0.2 & ??? & 0.1 \end{bmatrix}$$

$$\mathbf{WP} = \begin{bmatrix} ??? & 0.0 & 0.0 & 0.0 & 0.0 & 0.3 \\ ??? & 0.0 & 0.0 & 0.0 & 0.0 & 0.9 \end{bmatrix}$$

$$\mathbf{P} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Proposed approach

- Finding Permutations
- Complexity O(N)

Algorithm 1 Finding permutations for SPOC

Initialize known count $k = 0$
Initialize unknown count $u = 0$
for all points $i \in 1 : N$ **do**
  if point i is known **then**
    Increment k
    $P_{i,k} = 1$
  else
    $P_{i,N-u} = 1$
    Increment u
  end if
end for

$$\mathbf{W} = \begin{bmatrix} ??? & 0.3 & ??? & 0.8 & ??? & 0.5 \\ ??? & 0.9 & ??? & 0.2 & ??? & 0.1 \end{bmatrix}$$

$$\mathbf{WP} = \begin{bmatrix} ??? & ??? & 0.0 & 0.0 & 0.0 & 0.3 \\ ??? & ??? & 0.0 & 0.0 & 0.0 & 0.9 \end{bmatrix}$$

$$\mathbf{P} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Proposed approach

- Finding Permutations
- Complexity O(N)

Algorithm 1 Finding permutations for SPOC

```
Initialize known count k = 0
Initialize unknown count u = 0
for all points i ∈ 1 : N do
   if point i is known then
      Increment k
      P_{i,k} = 1
   else
      P_{i,N-u} = 1
      Increment u
   end if
end for
```

$$\mathbf{W} = \begin{bmatrix} ??? & 0.3 & ??? & 0.8 & ??? & 0.5 \\ ??? & 0.9 & ??? & 0.2 & ??? & 0.1 \end{bmatrix}$$

$$\mathbf{WP} = \begin{bmatrix} ??? & ??? & 0.0 & 0.0 & 0.8 & 0.3 \\ ??? & ??? & 0.0 & 0.0 & 0.2 & 0.9 \end{bmatrix}$$

$$\mathbf{P} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Proposed approach

- Finding Permutations
- Complexity O(N)

$$\mathbf{W} = \begin{bmatrix} ??? & 0.3 & ??? & 0.8 & ??? & 0.5 \\ ??? & 0.9 & ??? & 0.2 & ??? & 0.1 \end{bmatrix}$$

$$\mathbf{WP} = \begin{bmatrix} ??? & ??? & ??? & 0.0 & 0.8 & 0.3 \\ ??? & ??? & ??? & 0.0 & 0.2 & 0.9 \end{bmatrix}$$

$$\mathbf{P} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Algorithm 1 Finding permutations for SPOC

```
Initialize known count k = 0
Initialize unknown count u = 0
for all points i ∈ 1 : N do
   if point i is known then
      Increment k
      P_{i,k} = 1
   else
      P_{i,N−u} = 1
      Increment u
   end if
end for
```

Proposed approach

- Dealing with outliers: Robust PCA
  [Cai and Candès 2008; Lin et al. 2009]

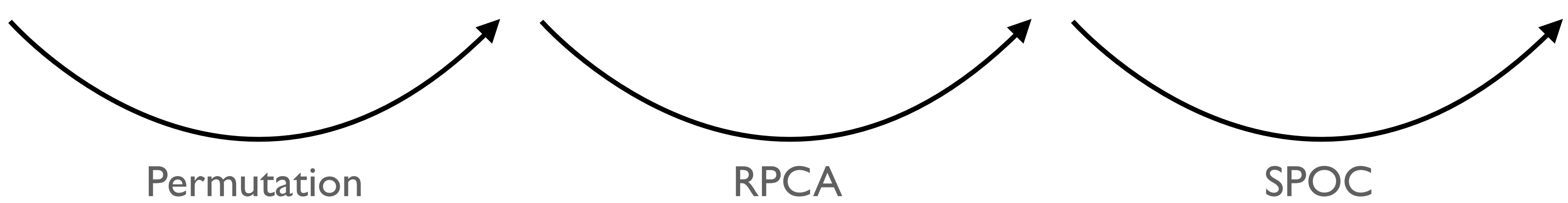
$$\begin{aligned} \text{minimize} \quad & ||\mathbf{A}||_* + \lambda ||\mathbf{E}||_1 \\ \text{subject to} \quad & \mathbf{D} = \mathbf{A} + \mathbf{E}, \end{aligned}$$

- Shares properties (and pitfalls) of Nuclear Norm completion

Proposed approach

- Final algorithm

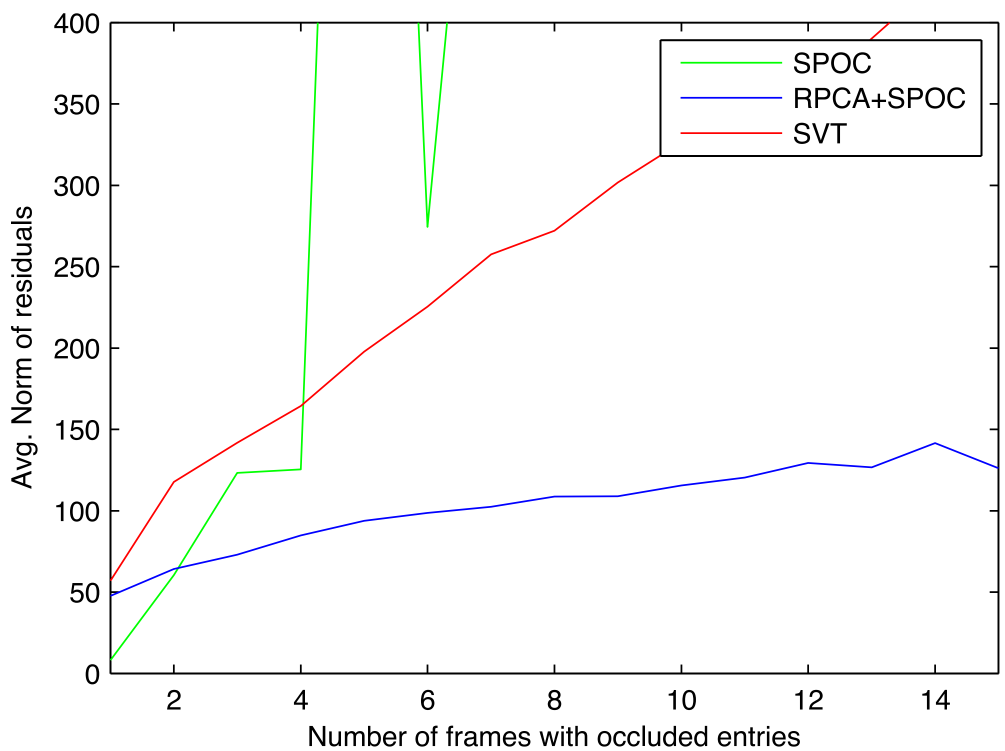
$$
\begin{bmatrix} ? & 17 & 0.8 & ? \\ ? & 23 & 0.2 & ? \\ 0.3 & 0.3 & 0.4 & 0.9 \\ 0.9 & 0.1 & 0.2 & 0.1 \\ & & \vdots & \end{bmatrix}
\begin{bmatrix} ? & ? & 17 & 0.8 \\ ? & ? & 23 & 0.2 \\ 0.3 & 0.9 & 0.3 & 0.4 \\ 0.9 & 0.1 & 0.1 & 0.2 \\ & & \vdots & \end{bmatrix}
\begin{bmatrix} ? & ? & \color{blue}{0.7} & 0.8 \\ ? & ? & \color{blue}{0.9} & 0.2 \\ 0.3 & 0.9 & 0.3 & 0.4 \\ 0.9 & 0.1 & 0.1 & 0.2 \\ & & \vdots & \end{bmatrix}
\begin{bmatrix} \color{blue}{0.1} & \color{blue}{0.5} & 0.3 & 0.8 \\ \color{blue}{0.3} & \color{blue}{0.2} & 0.9 & 0.2 \\ 0.7 & 0.9 & 0.3 & 0.4 \\ 0.9 & 0.1 & 0.1 & 0.2 \\ & & \vdots & \end{bmatrix}
$$

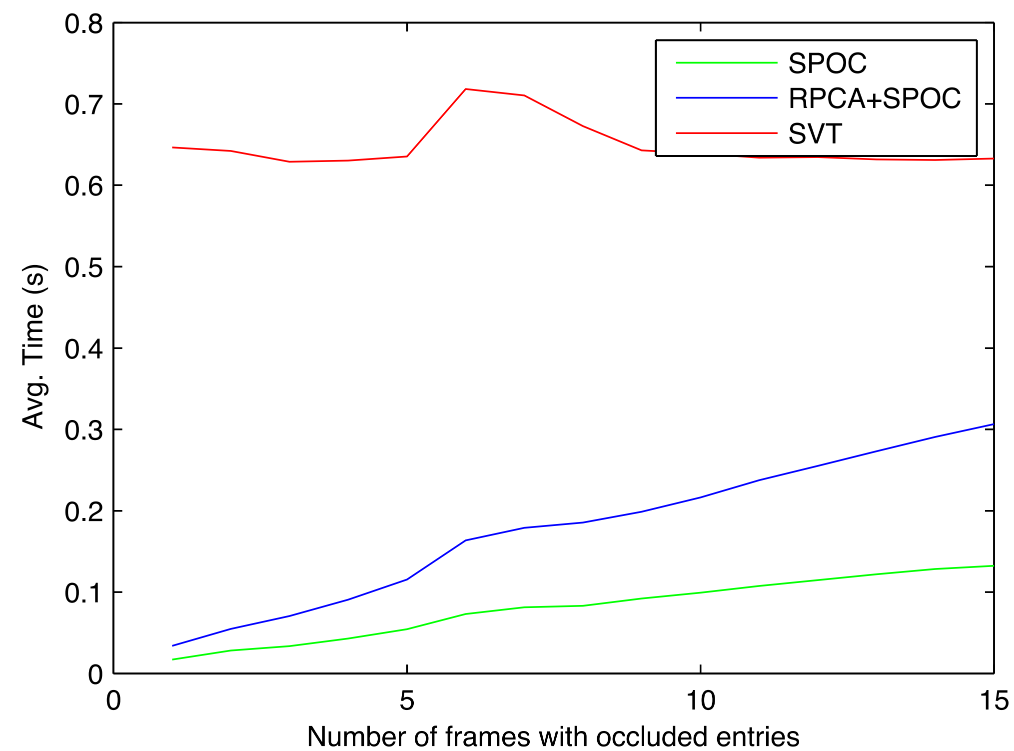


- Optimal for one frame, sub-optimal for more
- We still do SVDs, but smaller and in less number

Experiments

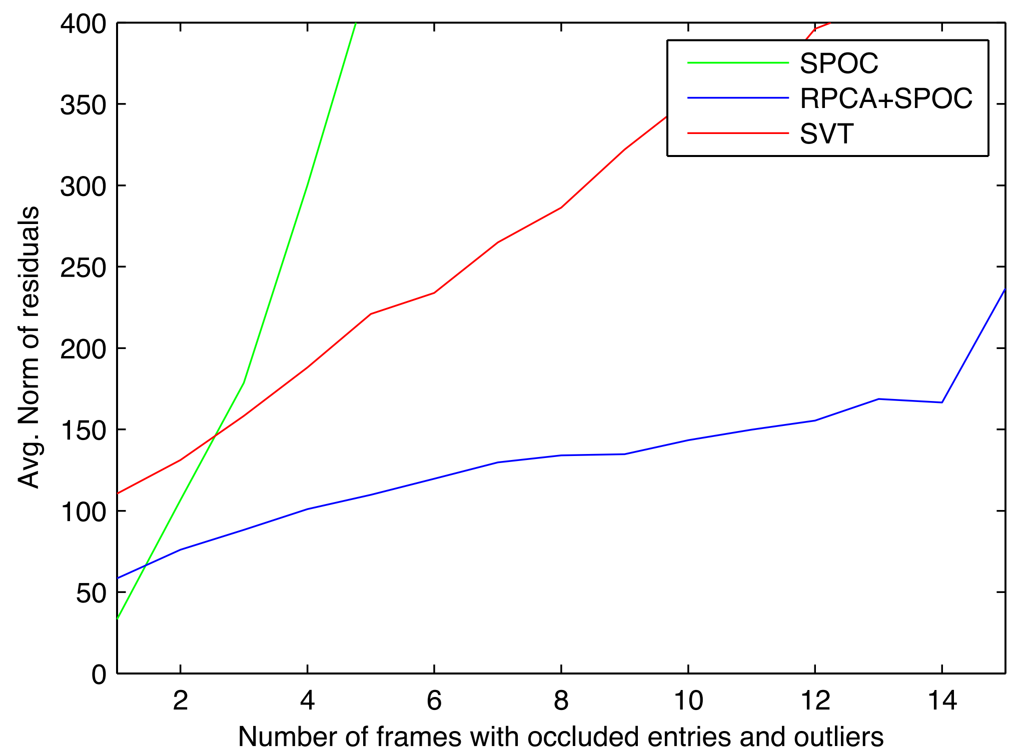
- 20 Frames; 20 Points in [-50;50];
- Trajectories on 5 frames known *a priori*
- Occlusion of 40% in new frames

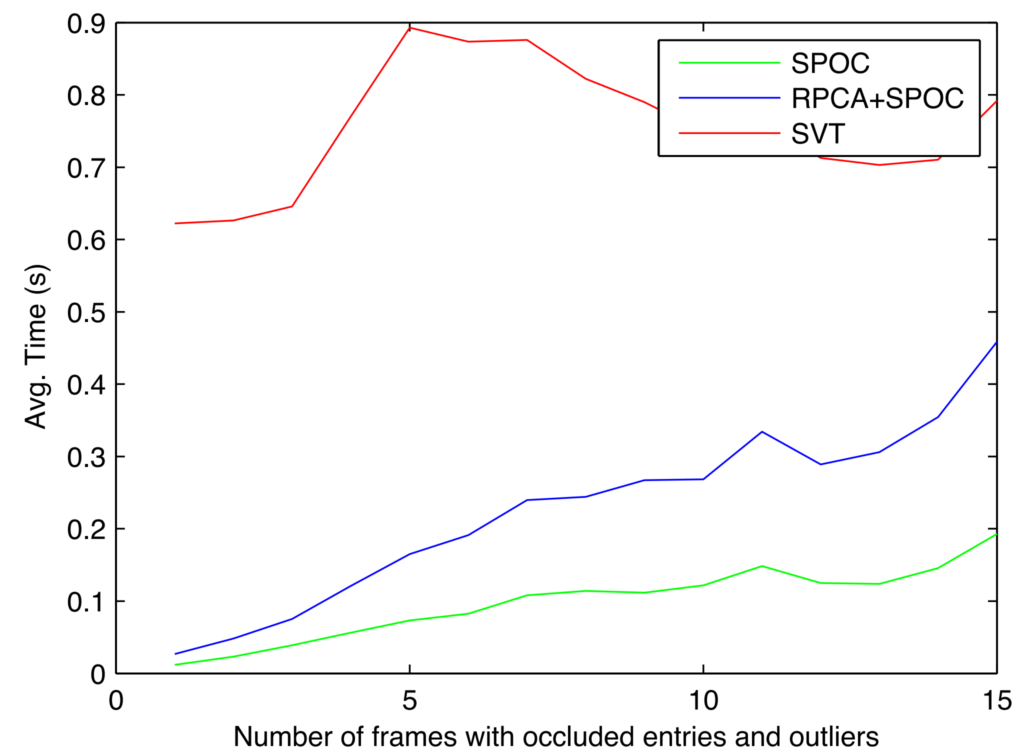




Experiments

- 5 frames known *a priori*
- Occlusion of 40% in new frames
- Outliers in10% of remaining points



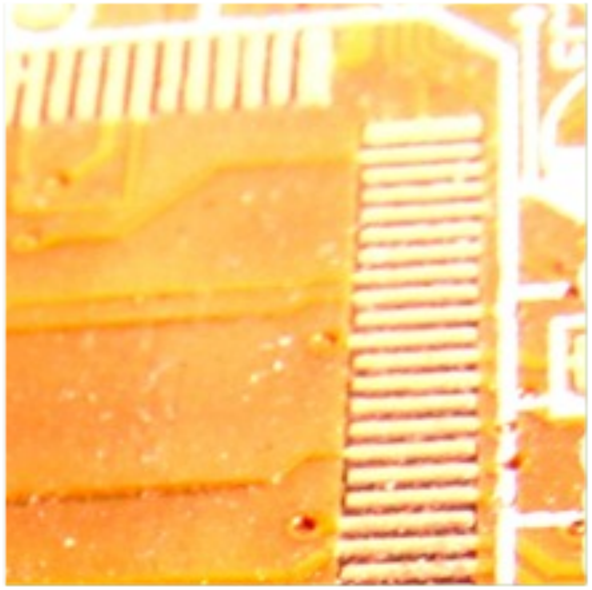
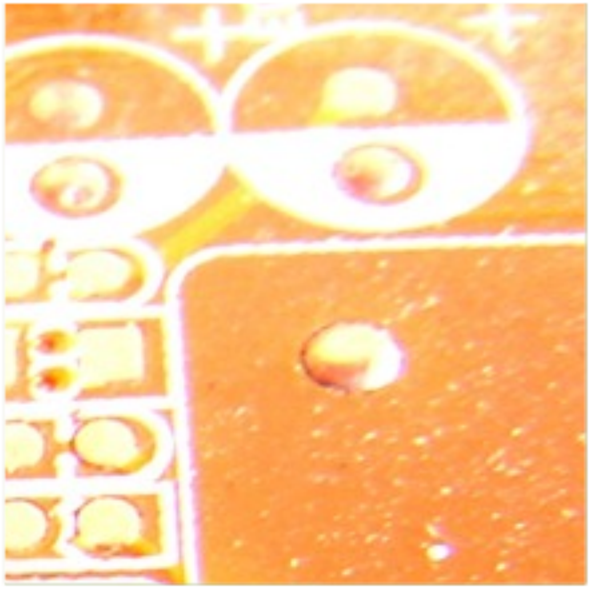
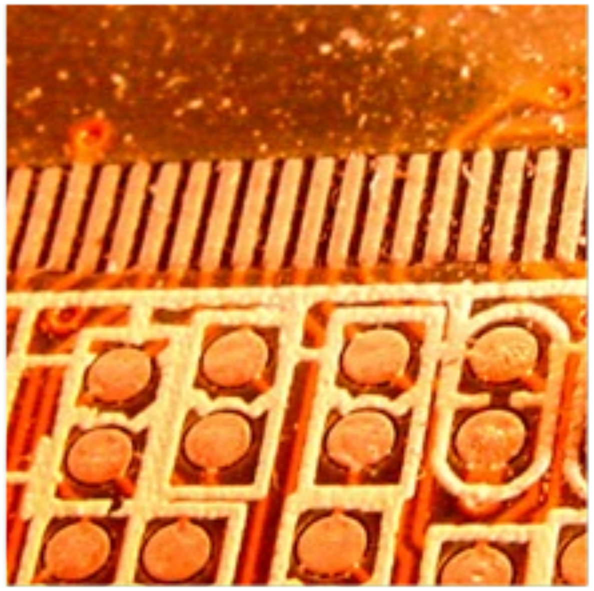


INSTITUTO SUPERIOR TÉCNICO
Universidade Técnica de Lisboa

Information and Communication Technologies Institute
Carnegie Mellon | PORTUGAL
AN INTERNATIONAL PARTNERSHIP

Conclusions

- New method for nuclear norm minimization
  - Rank has to be known *a priori*
  - Assume initial number of frames is known
  - Incremental method
  - Faster than state-of-the-art
  - Easily converted to a distributed algorithm

Thank you!