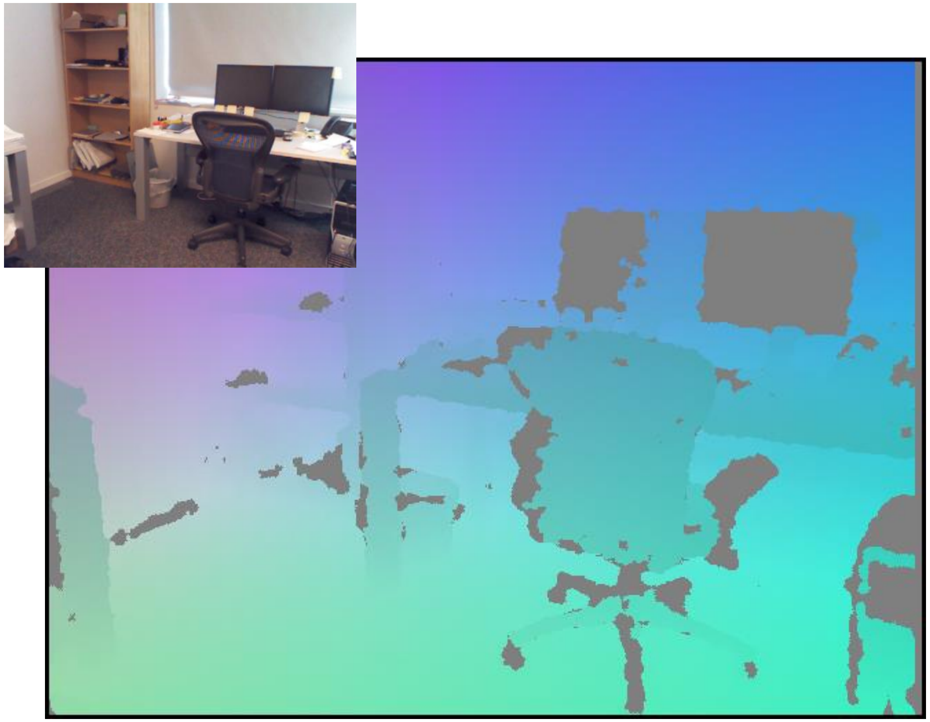
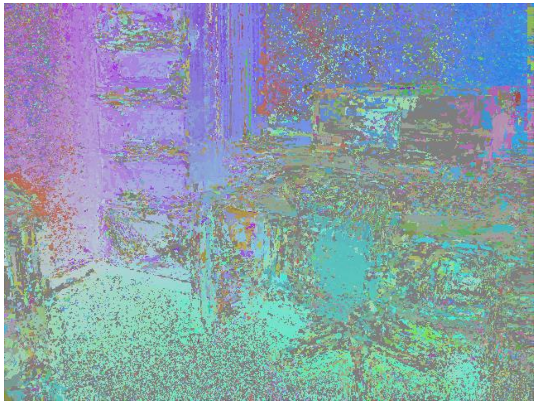
Object Coordinate Regression Results

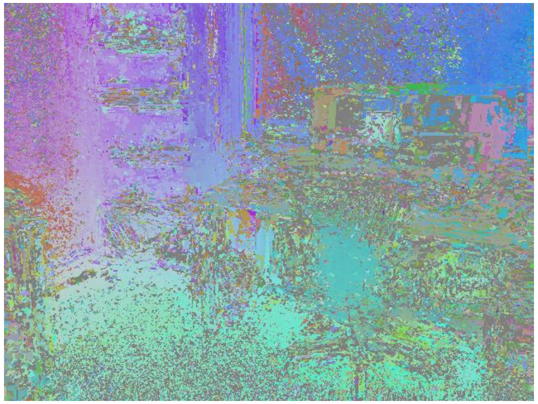


$\boldsymbol{X}^{Obj}$ (ground truth)

$\boldsymbol{X}^{Obj}$ (tree 1)



$\boldsymbol{X}^{Obj}$ (tree 2)



$\boldsymbol{X}^{Obj}$ (tree 3)

Problem: Outliers

Solution: RANSAC

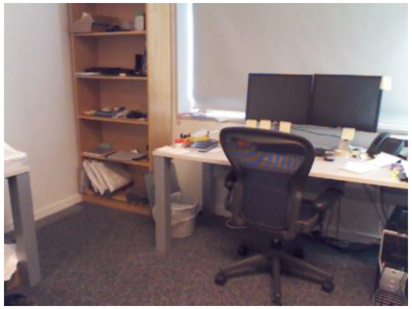
RANSAC

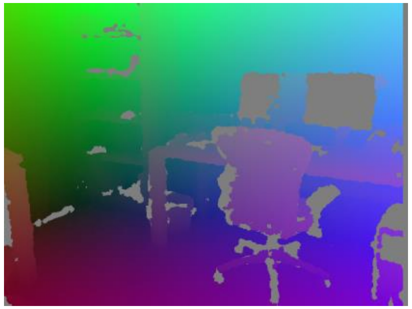
$\boldsymbol{R}^*, \boldsymbol{C}^*$
$E^* \coloneqq 0$
repeat n times:
  sample $\left\{(\boldsymbol{X}_1^{Cam}, \boldsymbol{X}_1^{Obj}), (\boldsymbol{X}_2^{Cam}, \boldsymbol{X}_2^{Obj}), (\boldsymbol{X}_3^{Cam}, \boldsymbol{X}_3^{Obj})\right\}$
  $\boldsymbol{R}, \boldsymbol{C} \coloneqq$ solve Kabsch
  $E(\boldsymbol{R}, \boldsymbol{C}) \coloneqq \sum_{i \in I} \sum_{t \in T} \mathbb{1}\left[\left\|\boldsymbol{X}_i^{Cam} - (\boldsymbol{R}\boldsymbol{X}_{i,t}^{Obj} - \boldsymbol{C})\right\| < \tau\right]$
  if $E(\boldsymbol{R}, \boldsymbol{C}) > E^*$:
    $E^* \coloneqq E(\boldsymbol{R}, \boldsymbol{C})$
    $\boldsymbol{R}^* \coloneqq \boldsymbol{R}, \boldsymbol{C}^* \coloneqq \boldsymbol{C}$
refine $\boldsymbol{R}^*, \boldsymbol{C}^*$

E	… inlier count
τ	… inlier threshold
I	… set of image pixels
T	… set of regression trees

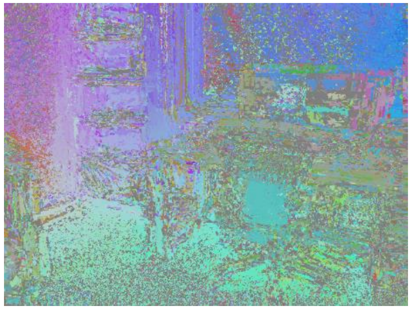
RANSAC

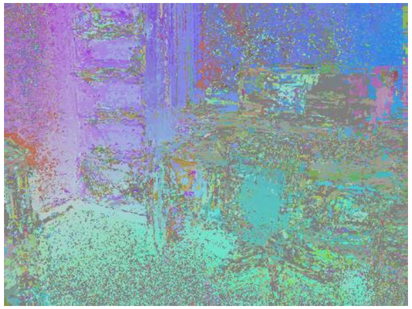
Input:

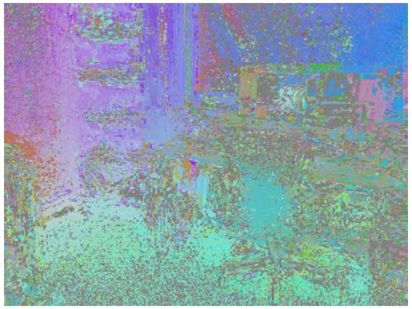

RGB-D


$\boldsymbol{X}^{Cam}$

Forest Prediction:


$\boldsymbol{X}^{Obj}$ (tree 1)


$\boldsymbol{X}^{Obj}$ (tree 2)

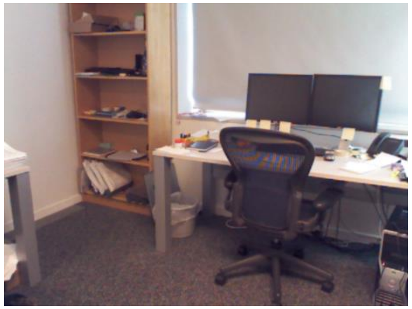

$\boldsymbol{X}^{Obj}$ (tree 3)

Inliers:

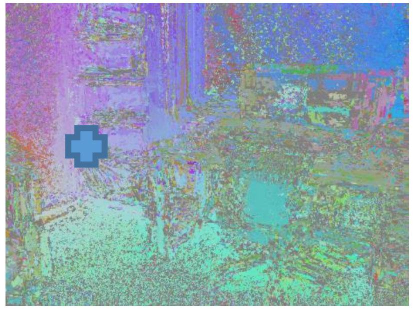
$\boldsymbol{R}^*, \boldsymbol{C}^*$
$E^* = 0$
`repeat 3 times:`
  `sample`
  $\boldsymbol{R}, \boldsymbol{C} :=$ `Kabsch`
  $E(\boldsymbol{R}, \boldsymbol{C}) = ?$
  `if` $E(\boldsymbol{R}, \boldsymbol{C}) > E^*$:
    $E^* := E(\boldsymbol{R}, \boldsymbol{C})$
    $\boldsymbol{R}^* := \boldsymbol{R}, \boldsymbol{C}^* := \boldsymbol{C}$
`refine` $\boldsymbol{R}^*, \boldsymbol{C}^*$

RANSAC

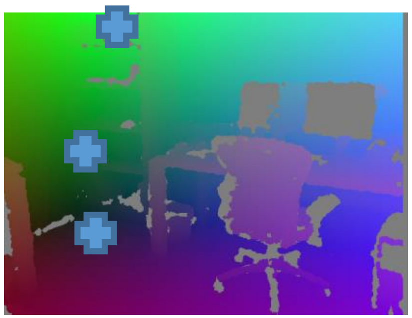
Input: Forest Prediction: Inliers:



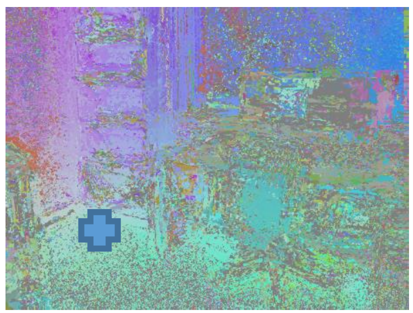
RGB-D



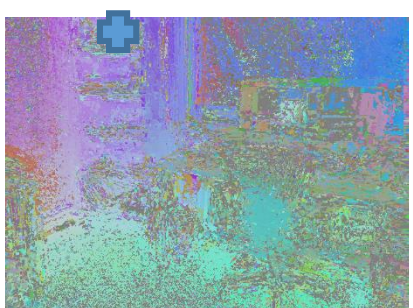
$\boldsymbol{X}^{Obj}$ (tree 1)



$\boldsymbol{X}^{Cam}$



$\boldsymbol{X}^{Obj}$ (tree 2)



$\boldsymbol{X}^{Obj}$ (tree 3)

$\boldsymbol{R}^*, \boldsymbol{C}^*$

$E^* = 0$

`repeat 3 times:`

    `sample`

    $\boldsymbol{R}, \boldsymbol{C} :=$ `Kabsch`

    $E(\boldsymbol{R}, \boldsymbol{C}) = ?$

    `if` $E(\boldsymbol{R}, \boldsymbol{C}) > E^*$ `:`

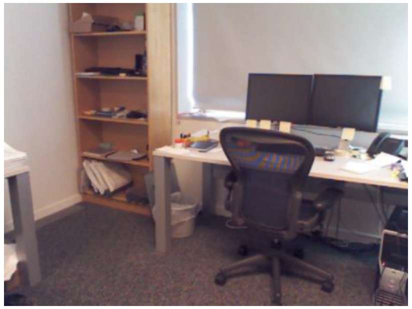
        $E^* := E(\boldsymbol{R}, \boldsymbol{C})$

        $\boldsymbol{R}^* := \boldsymbol{R}, \boldsymbol{C}^* := \boldsymbol{C}$

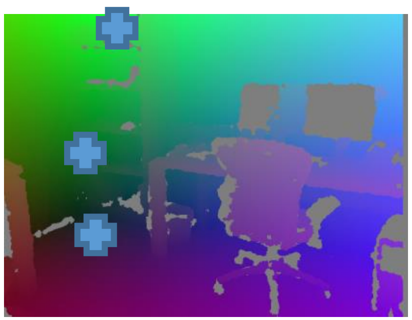
`refine` $\boldsymbol{R}^*, \boldsymbol{C}^*$

RANSAC

Input:

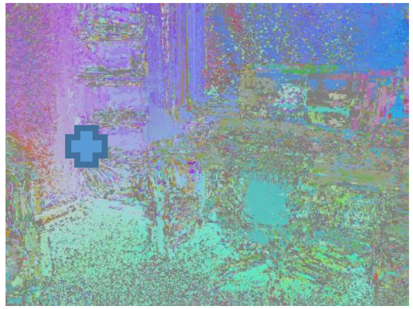


RGB-D

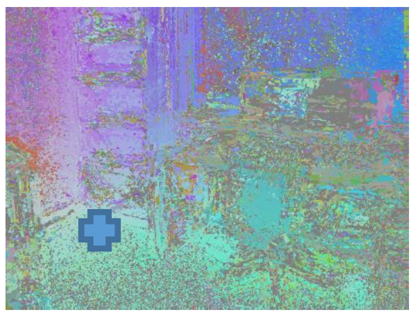


$\boldsymbol{X}^{Cam}$

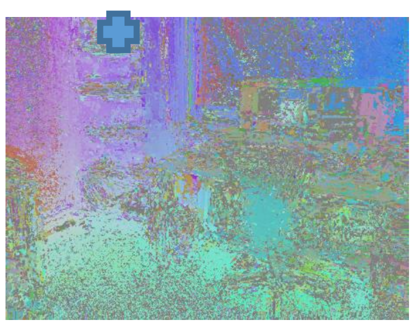
Forest Prediction:



$\boldsymbol{X}^{Obj}$ (tree 1)



$\boldsymbol{X}^{Obj}$ (tree 2)



$\boldsymbol{X}^{Obj}$ (tree 3)

Inliers:

$\boldsymbol{R}^*, \boldsymbol{C}^*$

$E^* = 0$

`repeat 3 times:`

  `sample`

  $\boldsymbol{R}_1, \boldsymbol{C}_1 :=$ `Kabsch`

  $E(\boldsymbol{R}, \boldsymbol{C}) = ?$

  `if` $E(\boldsymbol{R}, \boldsymbol{C}) > E^*$:

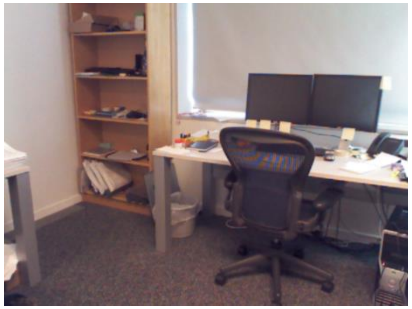
    $E^* := E(\boldsymbol{R}, \boldsymbol{C})$

    $\boldsymbol{R}^* := \boldsymbol{R}, \boldsymbol{C}^* := \boldsymbol{C}$

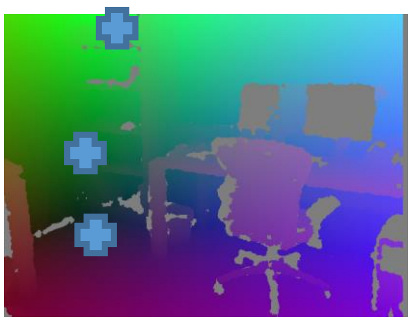
`refine` $\boldsymbol{R}^*, \boldsymbol{C}^*$

RANSAC

Input:

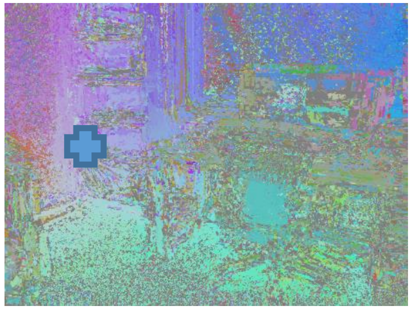


RGB-D

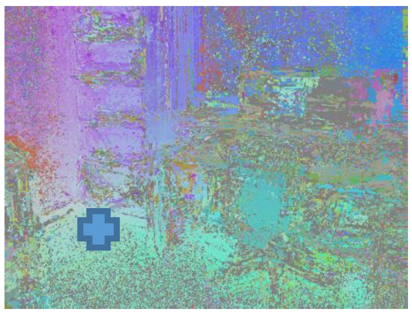


$\boldsymbol{X}^{Cam}$

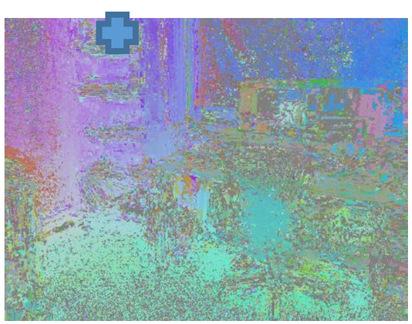
Forest Prediction:



$\boldsymbol{X}^{Obj}$ (tree 1)

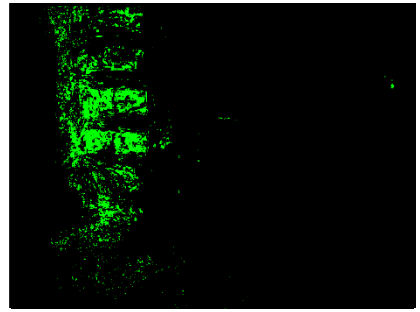


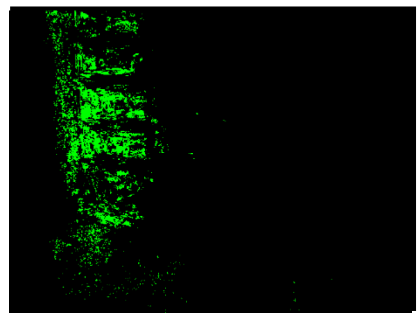
$\boldsymbol{X}^{Obj}$ (tree 2)

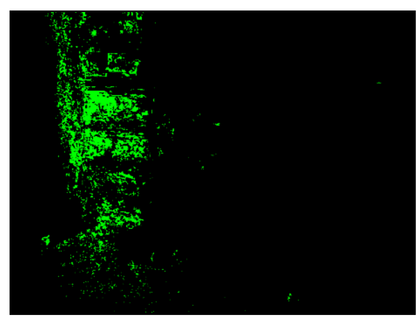


$\boldsymbol{X}^{Obj}$ (tree 3)

Inliers:







$\boldsymbol{R}^*, \boldsymbol{C}^*$

$E^* = 0$

repeat 3 times:

    sample

    $\boldsymbol{R}_1, \boldsymbol{C}_1 :=$ Kabsch

    $E(\boldsymbol{R}_1, \boldsymbol{C}_1) = 40772$

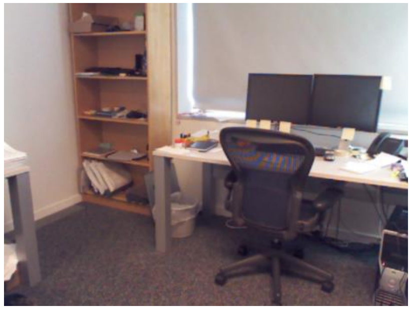
    if $E(\boldsymbol{R}, \boldsymbol{C}) > E^*$:

        $E^* := E(\boldsymbol{R}, \boldsymbol{C})$

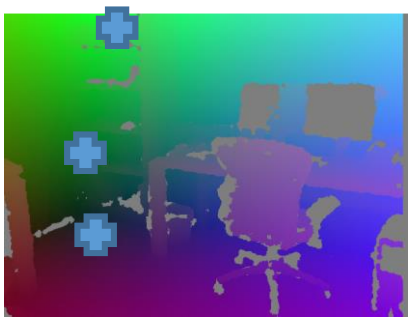
        $\boldsymbol{R}^* := \boldsymbol{R}, \boldsymbol{C}^* := \boldsymbol{C}$

refine $\boldsymbol{R}^*, \boldsymbol{C}^*$

Input:

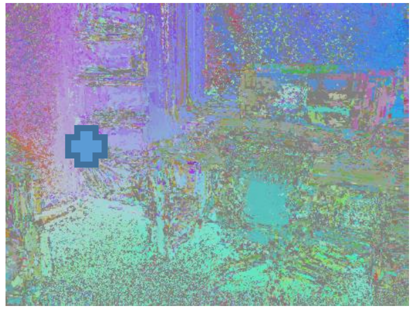


RGB-D

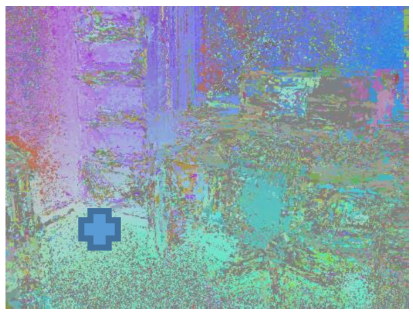


$\boldsymbol{X}^{Cam}$

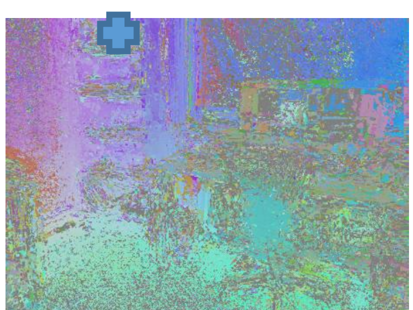
Forest Prediction:



$\boldsymbol{X}^{Obj}$ (tree 1)

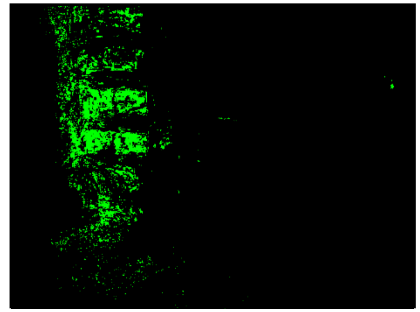


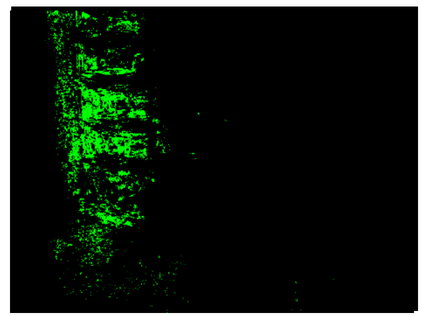
$\boldsymbol{X}^{Obj}$ (tree 2)

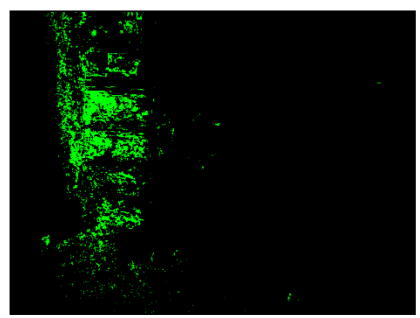


$\boldsymbol{X}^{Obj}$ (tree 3)

Inliers:







$\boldsymbol{R}^*, \boldsymbol{C}^*$

$E^* = 0$

`repeat 3 times:`

  `sample`

  $\boldsymbol{R}_1, \boldsymbol{C}_1 :=$ `Kabsch`

  $E(\boldsymbol{R}_1, \boldsymbol{C}_1) = 40772$

  `if` $E(\boldsymbol{R}_1, \boldsymbol{C}_1) > E^*$ `:`

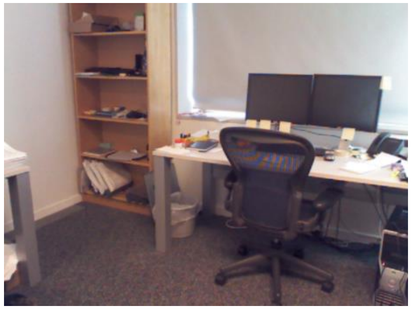
    $E^* := E(\boldsymbol{R}_1, \boldsymbol{C}_1)$

    $\boldsymbol{R}^* := \boldsymbol{R}_1, \boldsymbol{C}^* := \boldsymbol{C}_1$

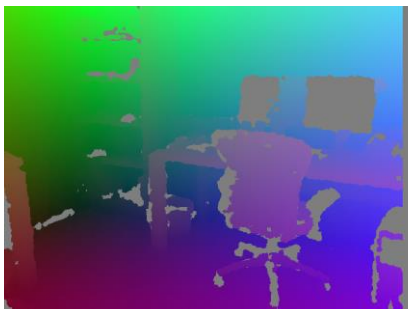
`refine` $\boldsymbol{R}^*, \boldsymbol{C}^*$

RANSAC

Input:

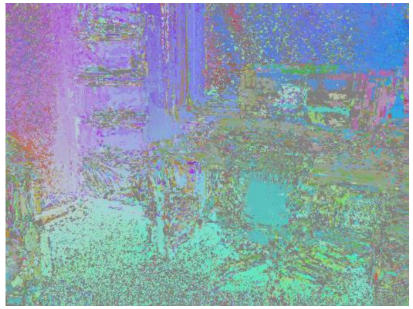


RGB-D

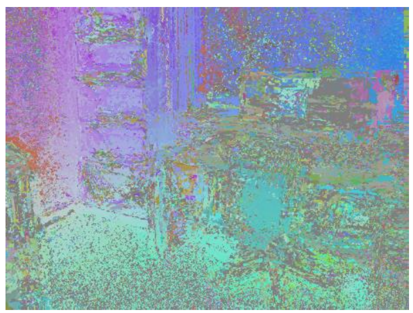


$\boldsymbol{X}^{Cam}$

Forest Prediction:



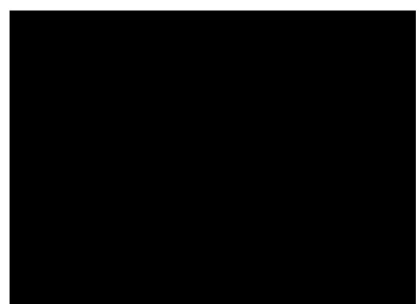
$\boldsymbol{X}^{Obj}$ (tree 1)



$\boldsymbol{X}^{Obj}$ (tree 2)

$\boldsymbol{X}^{Obj}$ (tree 3)

Inliers:

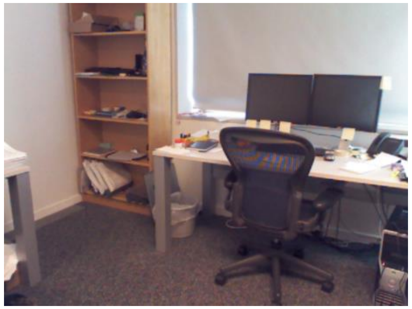


$\boldsymbol{R}^* = \boldsymbol{R}_1, \boldsymbol{C}^* = \boldsymbol{C}_1$
$E^* = 40772$

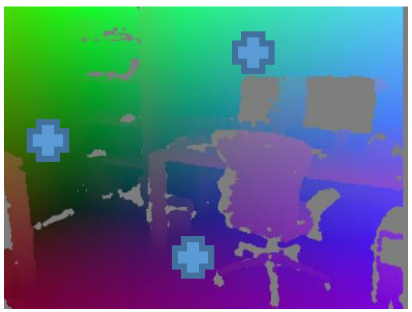
```
repeat 2 times:
   sample
```
   $\boldsymbol{R}, \boldsymbol{C} :=$ `Kabsch`
   $E(\boldsymbol{R}, \boldsymbol{C}) = ?$
   `if` $E(\boldsymbol{R}, \boldsymbol{C}) > E^*$:
      $E^* := E(\boldsymbol{R}, \boldsymbol{C})$
      $\boldsymbol{R}^* := \boldsymbol{R}, \boldsymbol{C}^* := \boldsymbol{C}$
`refine` $\boldsymbol{R}^*, \boldsymbol{C}^*$

RANSAC

Input:

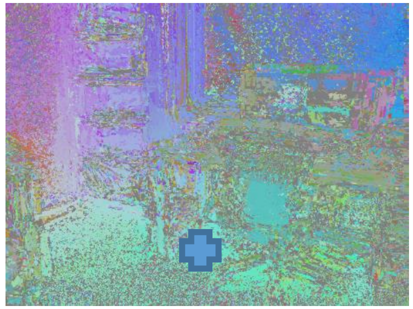


RGB-D

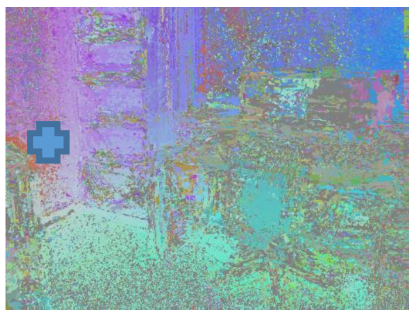


$\boldsymbol{X}^{Cam}$

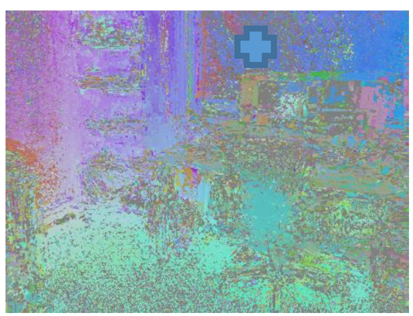
Forest Prediction:



$\boldsymbol{X}^{Obj}$ (tree 1)



$\boldsymbol{X}^{Obj}$ (tree 2)



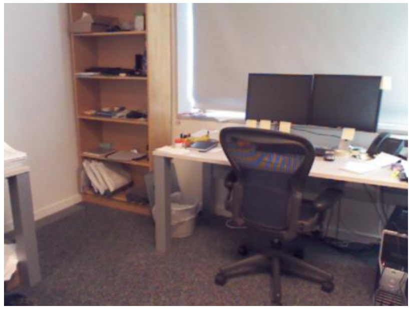
$\boldsymbol{X}^{Obj}$ (tree 3)

Inliers:

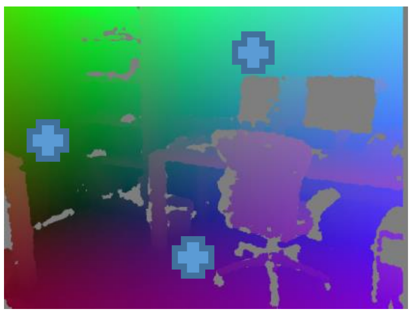
$\boldsymbol{R}^* = \boldsymbol{R}_1, \boldsymbol{C}^* = \boldsymbol{C}_1$
$E^* = 40772$
`repeat 2 times:`
    `sample`
    $\boldsymbol{R}, \boldsymbol{C} \coloneqq$ `Kabsch`
    $E(\boldsymbol{R}, \boldsymbol{C}) = ?$
    `if` $E(\boldsymbol{R}, \boldsymbol{C}) > E^*$`:`
        $E^* \coloneqq E(\boldsymbol{R}, \boldsymbol{C})$
        $\boldsymbol{R}^* \coloneqq \boldsymbol{R}, \boldsymbol{C}^* \coloneqq \boldsymbol{C}$
`refine` $\boldsymbol{R}^*, \boldsymbol{C}^*$

RANSAC

Input:

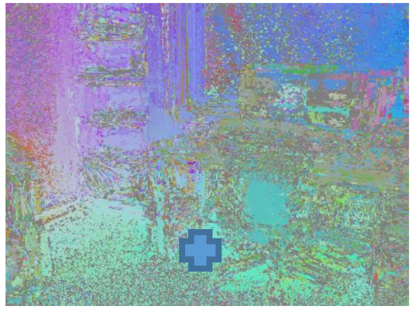


RGB-D

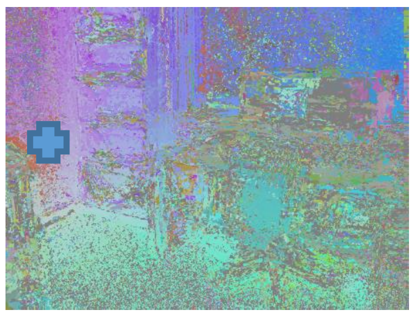


$\boldsymbol{X}^{Cam}$

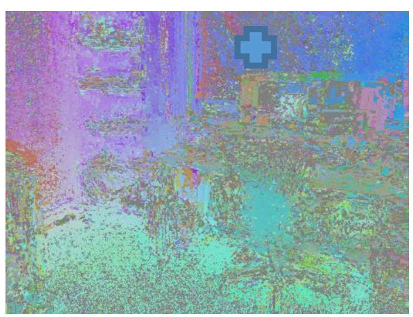
Forest Prediction:



$\boldsymbol{X}^{Obj}$ (tree 1)



$\boldsymbol{X}^{Obj}$ (tree 2)



$\boldsymbol{X}^{Obj}$ (tree 3)

Inliers:

$\boldsymbol{R}^* = \boldsymbol{R}_1, \boldsymbol{C}^* = \boldsymbol{C}_1$

$E^* = 40772$

`repeat 2 times:`

    `sample`

    $\boldsymbol{R}_2, \boldsymbol{C}_2 :=$ `Kabsch`

    $E(\boldsymbol{R}, \boldsymbol{C}) = ?$

    `if` $E(\boldsymbol{R}, \boldsymbol{C}) > E^*$:

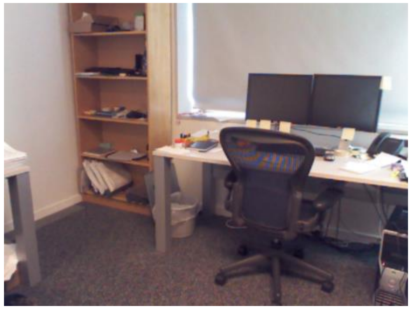
        $E^* := E(\boldsymbol{R}, \boldsymbol{C})$

        $\boldsymbol{R}^* := \boldsymbol{R}, \boldsymbol{C}^* := \boldsymbol{C}$

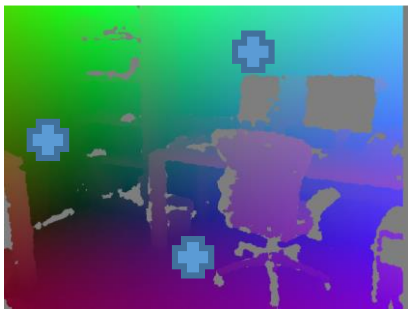
`refine` $\boldsymbol{R}^*, \boldsymbol{C}^*$

RANSAC

Input:

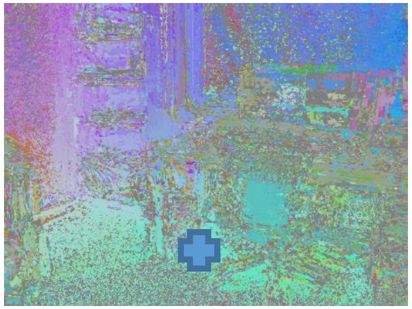


RGB-D

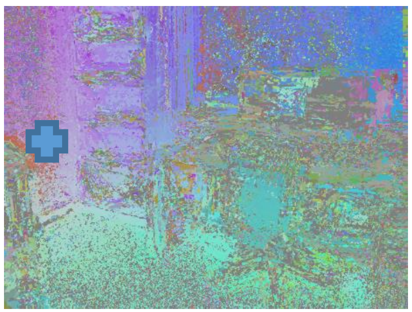


$\boldsymbol{X}^{Cam}$

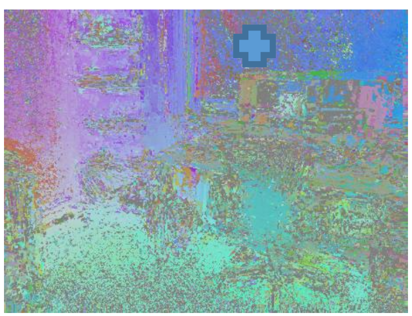
Forest Prediction:



$\boldsymbol{X}^{Obj}$ (tree 1)

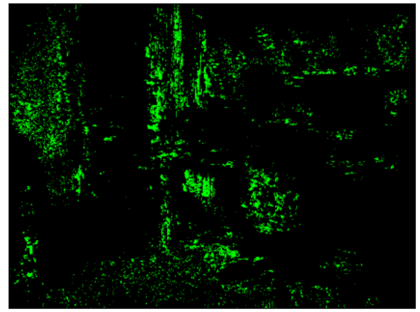


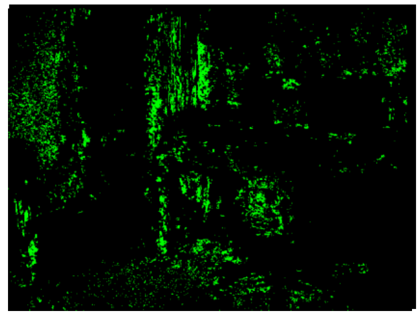
$\boldsymbol{X}^{Obj}$ (tree 2)

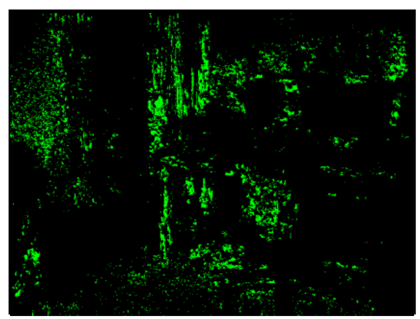


$\boldsymbol{X}^{Obj}$ (tree 3)

Inliers:







$\boldsymbol{R}^* = \boldsymbol{R}_1, \boldsymbol{C}^* = \boldsymbol{C}_1$

$E^* = 40772$

```
repeat 2 times:
   sample
```

   $\boldsymbol{R}_2, \boldsymbol{C}_2 :=$ `Kabsch`

   $E(\boldsymbol{R}_2, \boldsymbol{C}_2) = 54679$

   `if` $E(\boldsymbol{R}, \boldsymbol{C}) > E^*$:

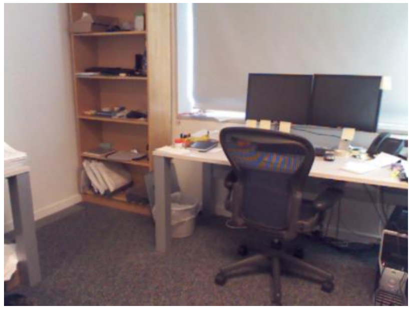
      $E^* := E(\boldsymbol{R}, \boldsymbol{C})$

      $\boldsymbol{R}^* := \boldsymbol{R}, \boldsymbol{C}^* := \boldsymbol{C}$

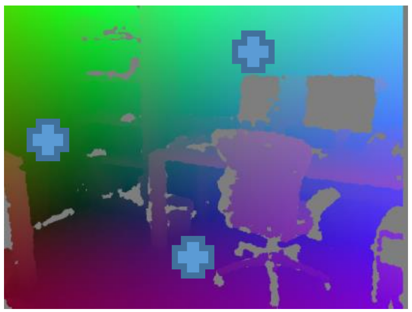
`refine` $\boldsymbol{R}^*, \boldsymbol{C}^*$

RANSAC

Input:

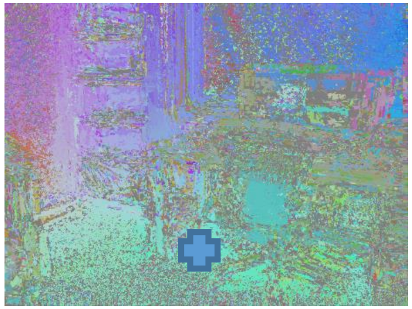


RGB-D

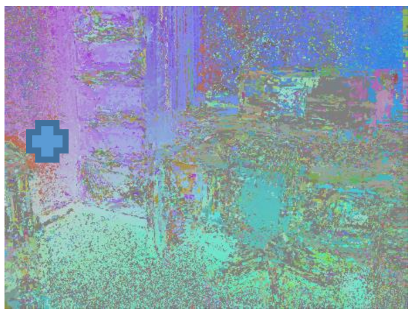


$\boldsymbol{X}^{Cam}$

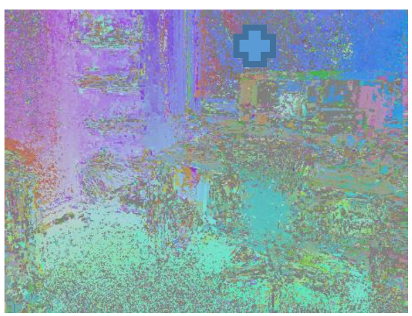
Forest Prediction:



$\boldsymbol{X}^{Obj}$ (tree 1)

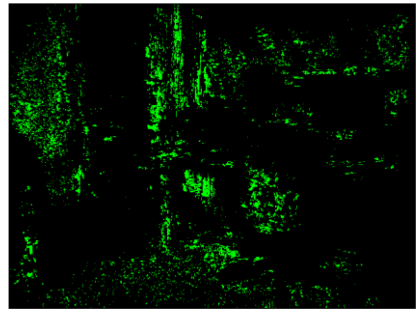


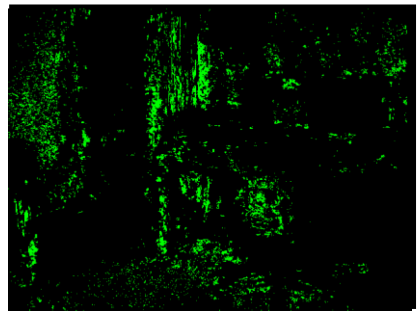
$\boldsymbol{X}^{Obj}$ (tree 2)

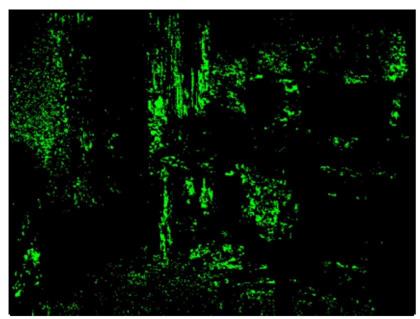


$\boldsymbol{X}^{Obj}$ (tree 3)

Inliers:







$\boldsymbol{R}^* = \boldsymbol{R}_1, \boldsymbol{C}^* = \boldsymbol{C}_1$

$E^* = 40772$

repeat 2 times:

  sample

  $\boldsymbol{R}_2, \boldsymbol{C}_2 :=$ Kabsch

  $E(\boldsymbol{R}_2, \boldsymbol{C}_2) = 54679$

  if $E(\boldsymbol{R}_2, \boldsymbol{C}_2) > E^*$:

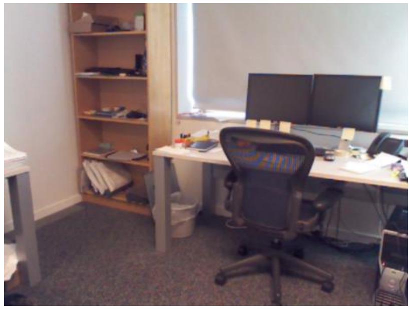
    $E^* := E(\boldsymbol{R}_2, \boldsymbol{C}_2)$

    $\boldsymbol{R}^* := \boldsymbol{R}_2, \boldsymbol{C}^* := \boldsymbol{C}_2$

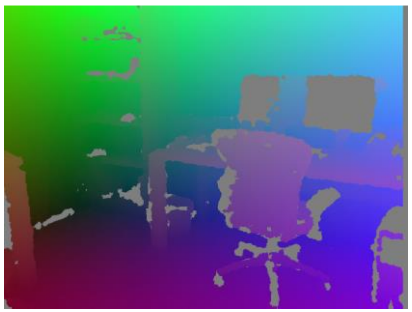
refine $\boldsymbol{R}^*, \boldsymbol{C}^*$

RANSAC

Input:

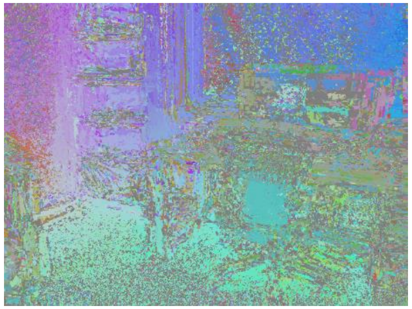


RGB-D

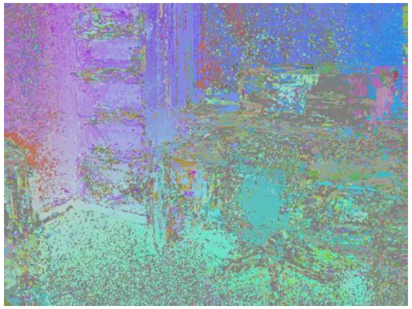


$\boldsymbol{X}^{Cam}$

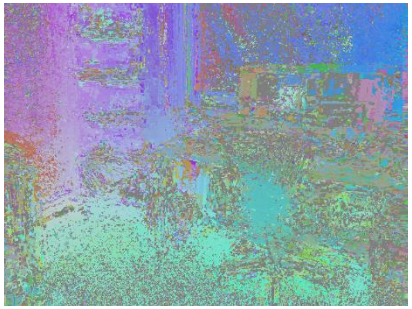
Forest Prediction:



$\boldsymbol{X}^{Obj}$ (tree 1)



$\boldsymbol{X}^{Obj}$ (tree 2)



$\boldsymbol{X}^{Obj}$ (tree 3)

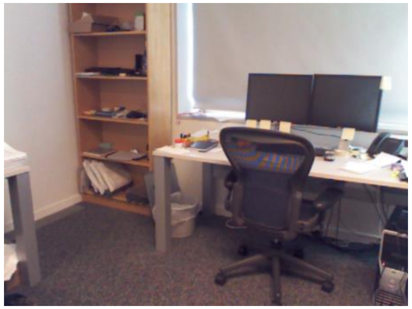
Inliers:

$\boldsymbol{R}^* = \boldsymbol{R}_2, \boldsymbol{C}^* = \boldsymbol{C}_2$

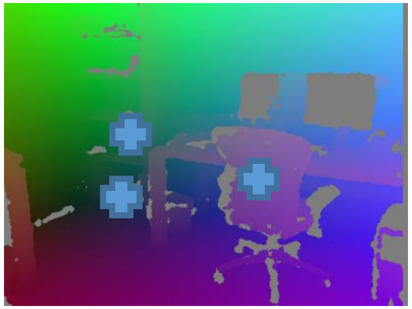
$E^* = 54679$

```
repeat 1 times:
   sample
   R,C := Kabsch
   E(R,C) =?
   if E(R,C) > E*:
      E* := E(R,C)
      R* := R, C* := C
refine R*,C*
```

Input:

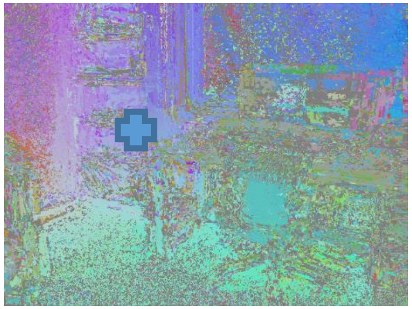


RGB-D

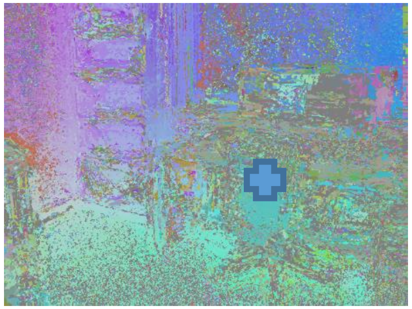


$\boldsymbol{X}^{Cam}$

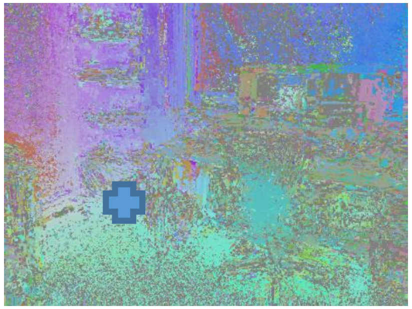
Forest Prediction:



$\boldsymbol{X}^{Obj}$ (tree 1)



$\boldsymbol{X}^{Obj}$ (tree 2)



$\boldsymbol{X}^{Obj}$ (tree 3)

Inliers:

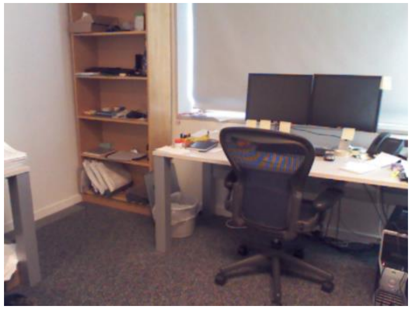
$\boldsymbol{R}^* = \boldsymbol{R}_2, \boldsymbol{C}^* = \boldsymbol{C}_2$

$E^* = 54679$

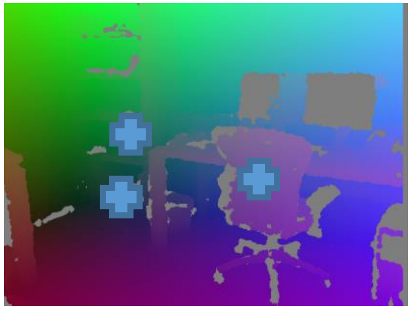
```
repeat 1 times:
   sample
   R, C := Kabsch
   E(R, C) =?
   if E(R, C) > E*:
      E* := E(R, C)
      R* := R, C* := C
refine R*, C*
```

RANSAC

Input:

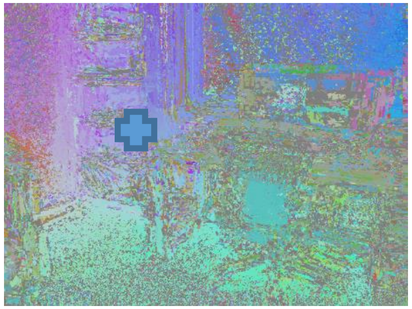


RGB-D

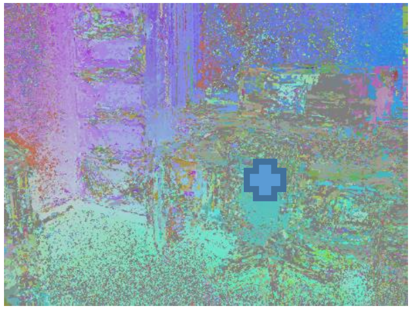


$\boldsymbol{X}^{Cam}$

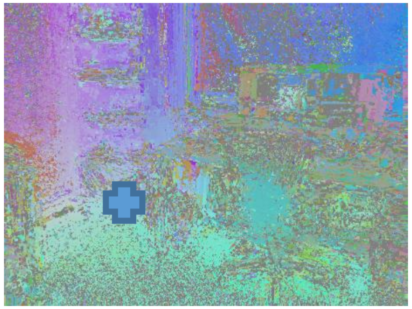
Forest Prediction:



$\boldsymbol{X}^{Obj}$ (tree 1)



$\boldsymbol{X}^{Obj}$ (tree 2)



$\boldsymbol{X}^{Obj}$ (tree 3)

Inliers:

$\boldsymbol{R}^* = \boldsymbol{R}_2, \boldsymbol{C}^* = \boldsymbol{C}_2$

$E^* = 54679$

```
repeat 1 times:
   sample
```

   $\boldsymbol{R}_3, \boldsymbol{C}_3 :=$ `Kabsch`

   $E(\boldsymbol{R}, \boldsymbol{C}) = ?$

   `if` $E(\boldsymbol{R}, \boldsymbol{C}) > E^*$:

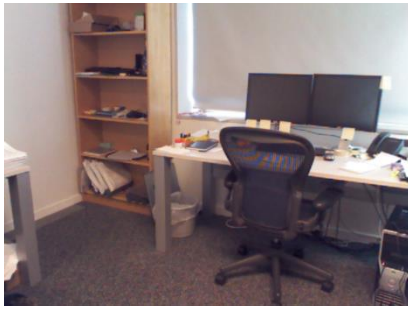
      $E^* := E(\boldsymbol{R}, \boldsymbol{C})$

      $\boldsymbol{R}^* := \boldsymbol{R}, \boldsymbol{C}^* := \boldsymbol{C}$

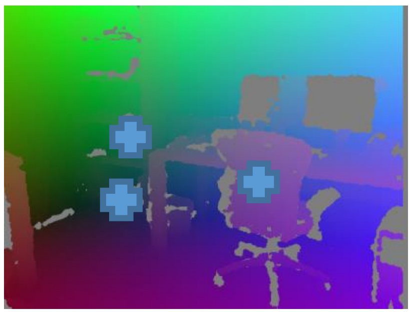
`refine` $\boldsymbol{R}^*, \boldsymbol{C}^*$

RANSAC

Input:

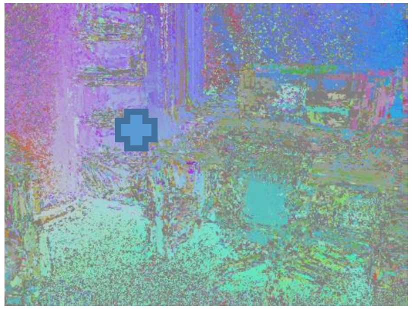


RGB-D

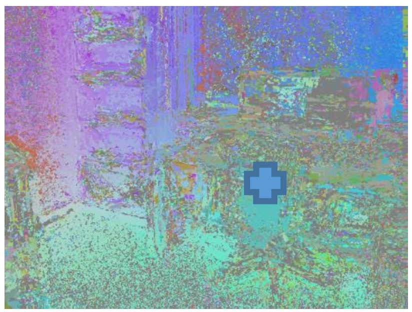


$\boldsymbol{X}^{Cam}$

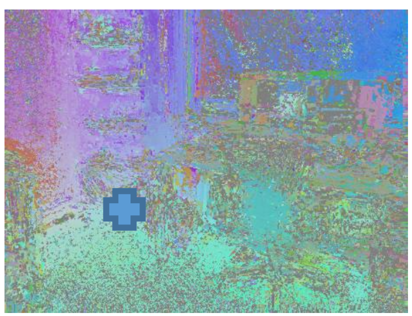
Forest Prediction:



$\boldsymbol{X}^{Obj}$ (tree 1)

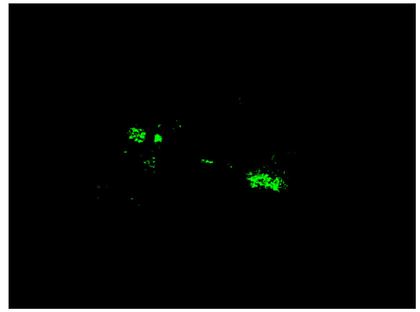


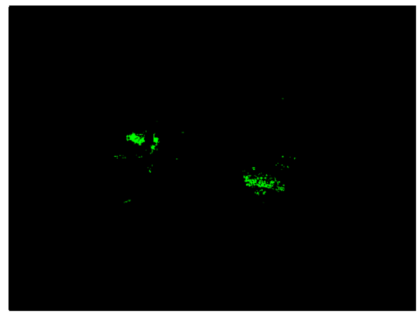
$\boldsymbol{X}^{Obj}$ (tree 2)

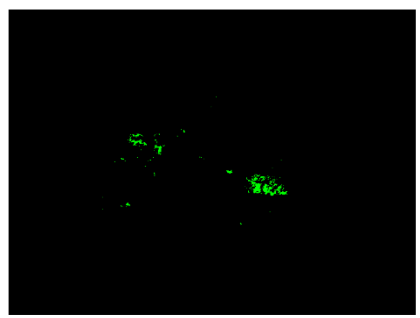


$\boldsymbol{X}^{Obj}$ (tree 3)

Inliers:



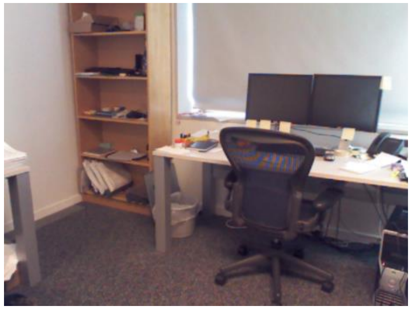




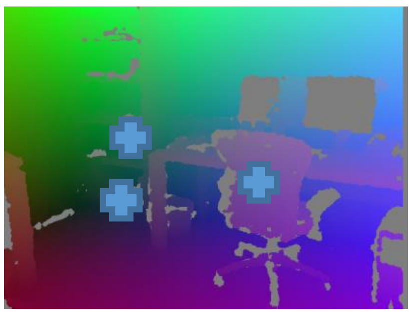
$\boldsymbol{R}^* = \boldsymbol{R}_2, \boldsymbol{C}^* = \boldsymbol{C}_2$
$E^* = 54679$
repeat 1 times:
  sample
  $\boldsymbol{R}_3, \boldsymbol{C}_3 :=$ Kabsch
  $E(\boldsymbol{R}_3, \boldsymbol{C}_3) = 3727$
  if $E(\boldsymbol{R}, \boldsymbol{C}) > E^*$:
    $E^* := E(\boldsymbol{R}, \boldsymbol{C})$
    $\boldsymbol{R}^* := \boldsymbol{R}, \boldsymbol{C}^* := \boldsymbol{C}$
refine $\boldsymbol{R}^*, \boldsymbol{C}^*$

RANSAC

Input:

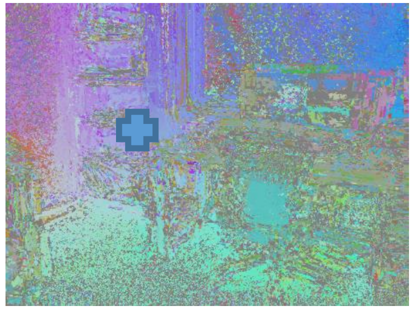


RGB-D

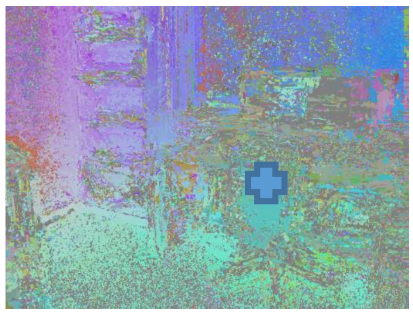


$\boldsymbol{X}^{Cam}$

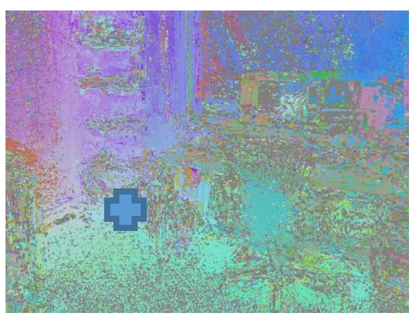
Forest Prediction:



$\boldsymbol{X}^{Obj}$ (tree 1)

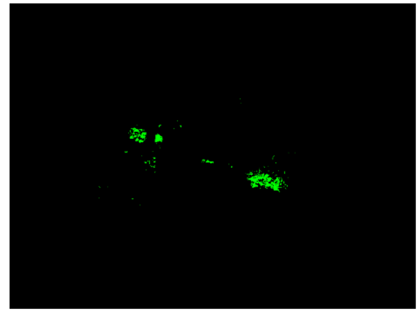


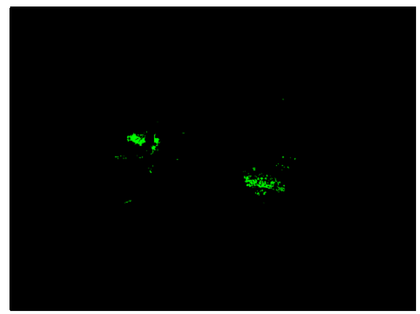
$\boldsymbol{X}^{Obj}$ (tree 2)

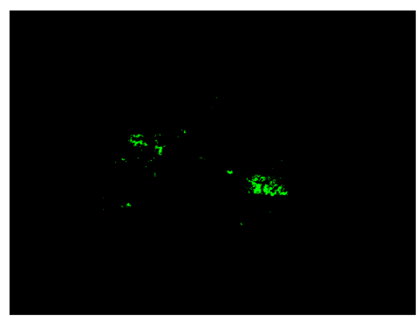


$\boldsymbol{X}^{Obj}$ (tree 3)

Inliers:







$\boldsymbol{R}^* = \boldsymbol{R}_2, \boldsymbol{C}^* = \boldsymbol{C}_2$
$E^* = 54679$

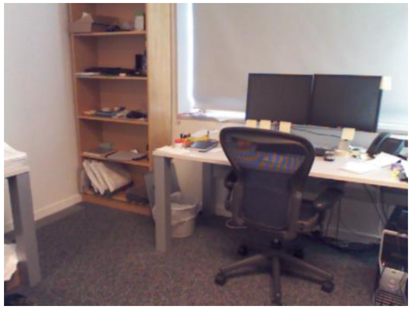
```
repeat 1 times:
   sample
```

$\quad \boldsymbol{R}_3, \boldsymbol{C}_3 \coloneqq$ `Kabsch`
$\quad E(\boldsymbol{R}_3, \boldsymbol{C}_3) = 3727$
$\quad$ `if` $E(\boldsymbol{R}_3, \boldsymbol{C}_3) > E^*$:
$\qquad E^* \coloneqq E(\boldsymbol{R}, \boldsymbol{C})$
$\qquad \boldsymbol{R}^* \coloneqq \boldsymbol{R}, \boldsymbol{C}^* \coloneqq \boldsymbol{C}$

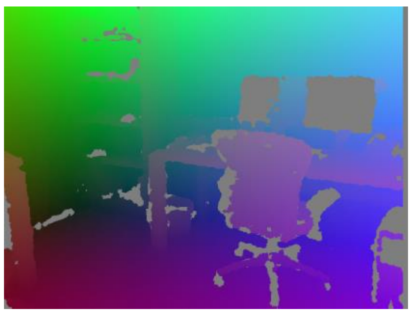
`refine` $\boldsymbol{R}^*, \boldsymbol{C}^*$

RANSAC

Input:

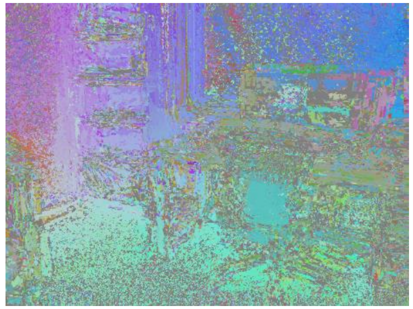


RGB-D

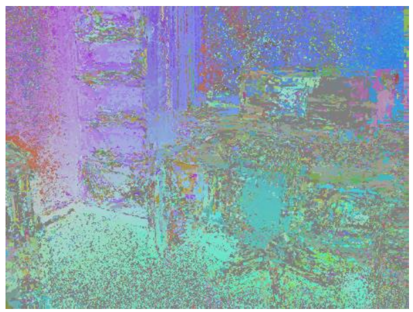


$\boldsymbol{X}^{Cam}$

Forest Prediction:



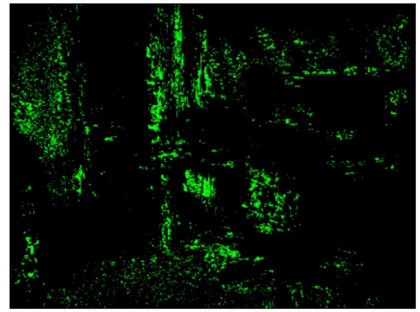
$\boldsymbol{X}^{Obj}$ (tree 1)

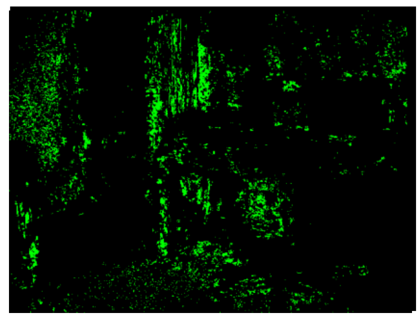


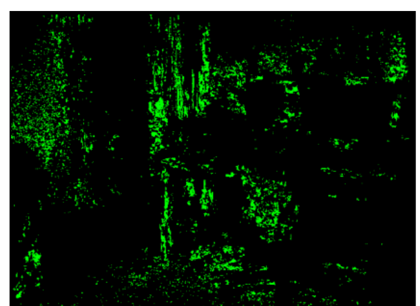
$\boldsymbol{X}^{Obj}$ (tree 2)

$\boldsymbol{X}^{Obj}$ (tree 3)

Inliers:







$\boldsymbol{R}^* = \boldsymbol{R}_2, \boldsymbol{C}^* = \boldsymbol{C}_2$

$E^* = 54679$

repeat 0 times:

  sample

    $\boldsymbol{R}, \boldsymbol{C} :=$ Kabsch

    $E(\boldsymbol{R}, \boldsymbol{C}) = ?$

    if $E(\boldsymbol{R}, \boldsymbol{C}) > E^*$:

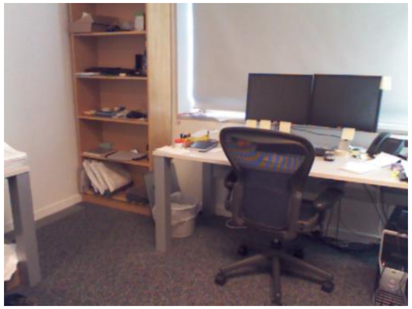
      $E^* := E(\boldsymbol{R}, \boldsymbol{C})$

      $\boldsymbol{R}^* := \boldsymbol{R}, \boldsymbol{C}^* := \boldsymbol{C}$

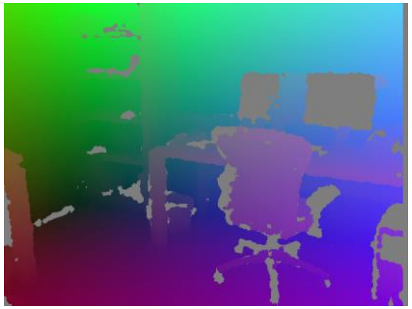
refine $\boldsymbol{R}^*, \boldsymbol{C}^*$

RANSAC

Input:

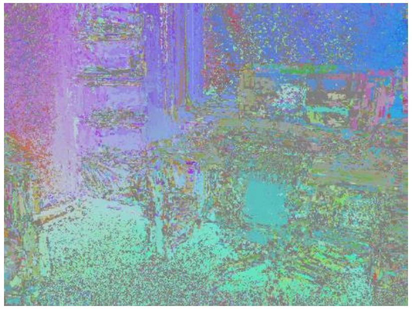


RGB-D

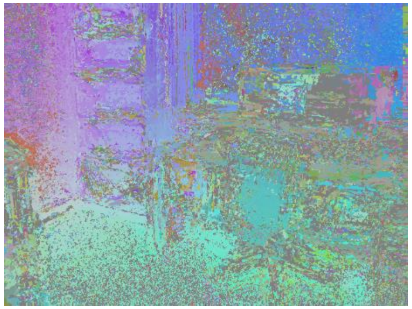


$\boldsymbol{X}^{Cam}$

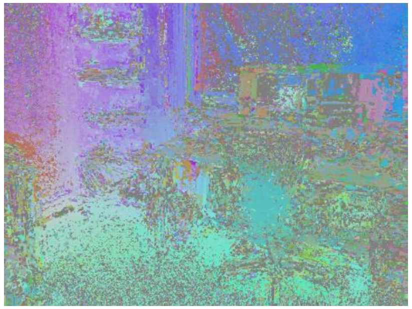
Forest Prediction:



$\boldsymbol{X}^{Obj}$ (tree 1)

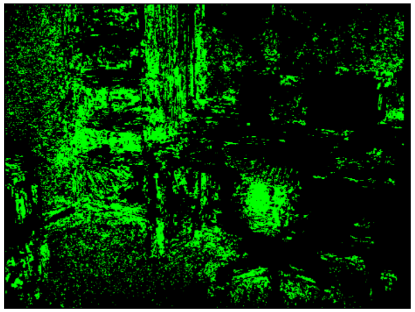


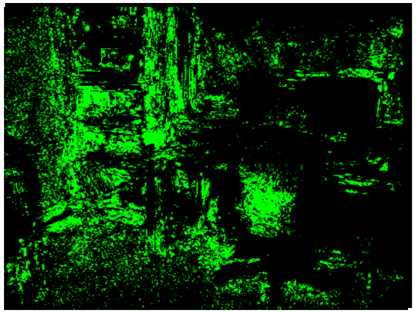
$\boldsymbol{X}^{Obj}$ (tree 2)

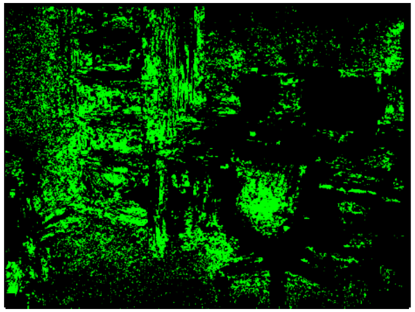


$\boldsymbol{X}^{Obj}$ (tree 3)

Inliers:







$\boldsymbol{R}^* = \boldsymbol{R}_2, \boldsymbol{C}^* = \boldsymbol{C}_2$

$E^* = 54679$

`repeat 0 times:`

  `sample`

    $\boldsymbol{R}, \boldsymbol{C} \coloneqq$ `Kabsch`

    $E(\boldsymbol{R}, \boldsymbol{C}) = ?$

    `if` $E(\boldsymbol{R}, \boldsymbol{C}) > E^*$:

      $E^* \coloneqq E(\boldsymbol{R}, \boldsymbol{C})$

      $\boldsymbol{R}^* \coloneqq \boldsymbol{R}, \boldsymbol{C}^* \coloneqq \boldsymbol{C}$

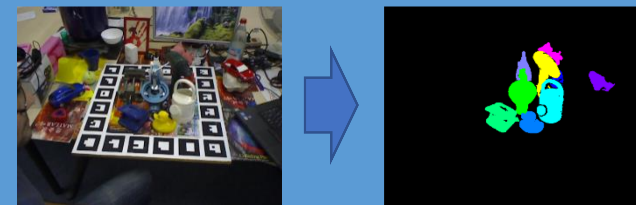
`refine` $\boldsymbol{R}^*, \boldsymbol{C}^*$

Final Inlier Count:

$E(\boldsymbol{R}^*, \boldsymbol{C}^*) = \mathbf{185414}$

Case Study: Pose Estimation and Pose Tracking

Image Segmentation:
- Decision Forest
- Features, Training Recap



Camera Re-Localization:
- Regression Forest
- Split-Criteria, Kabsch, RANSAC

One-Shot Pose Estimation:
- Decision-Regression Forest
- Hypothesis Sampling, Hypotheses Energy

One-Shot Pose Estimation

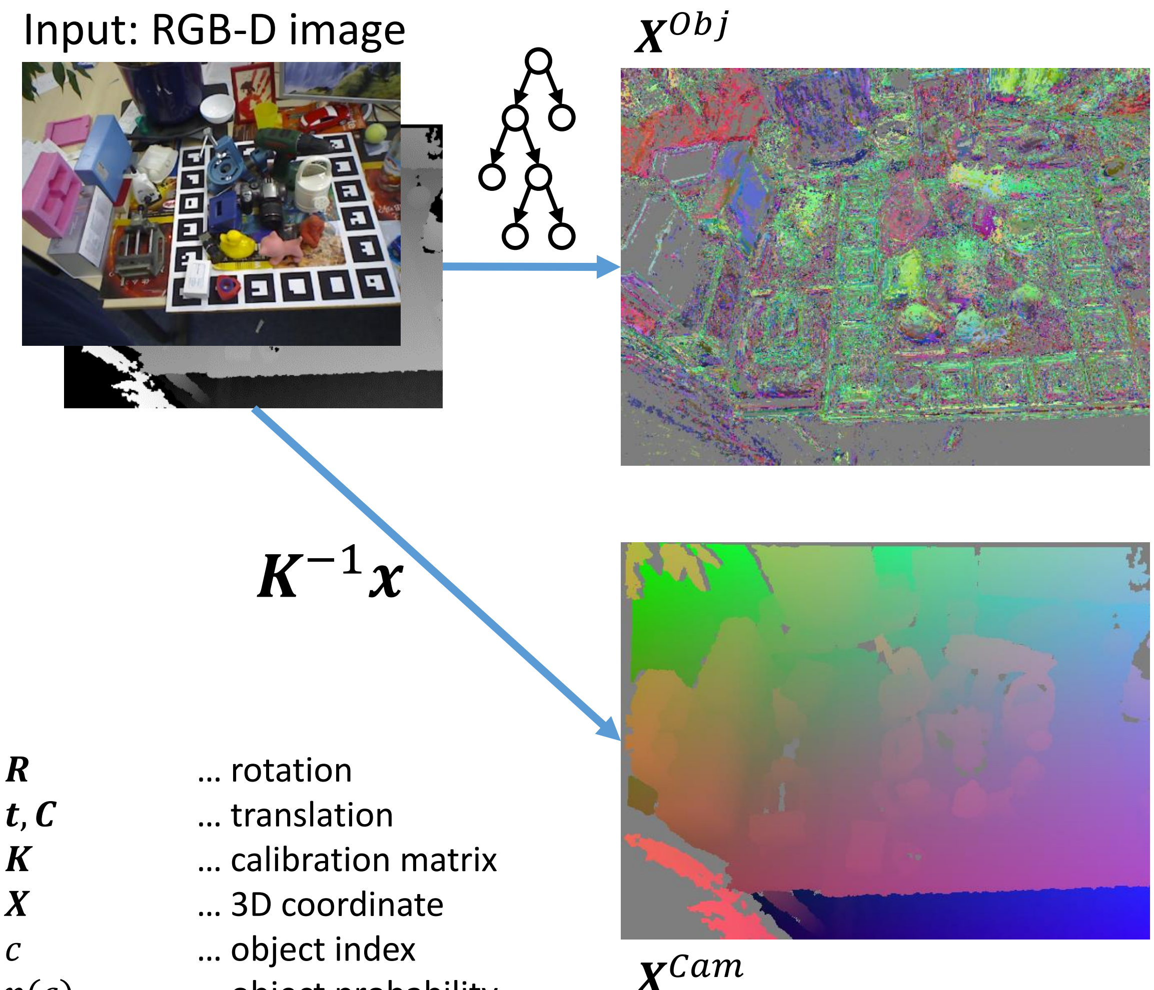
Input: RGB-D image

Output: 6DoF Object Pose

Dataset of Hinterstoisser et al., Model Based Training, Detection and Pose Estimation of Texture-Less 3D Objects in Heavily Cluttered Scenes, ACCV 12
Method of Brachmann et al., Learning 6d object pose estimation using 3d object coordinates, ECCV 14

Why not use camera re-localization?

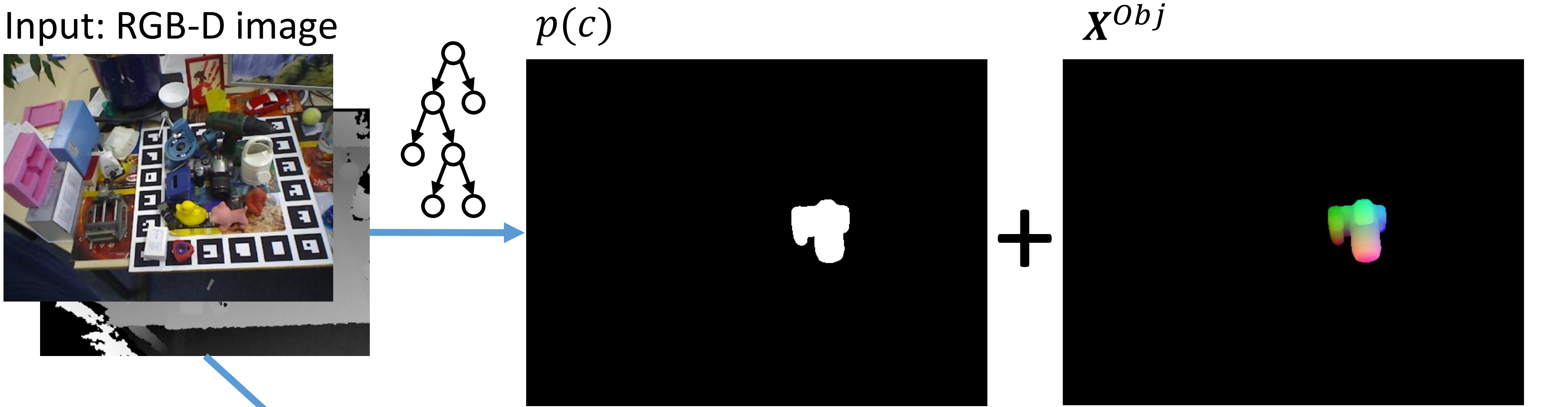
- $\boldsymbol{R}_{\boldsymbol{Obj}}(\boldsymbol{I}_{3\times3}| - \tilde{\boldsymbol{t}}_{\boldsymbol{Obj}}) = (\boldsymbol{R}_{\boldsymbol{Cam}}(\boldsymbol{I}_{3\times3}| - \tilde{\boldsymbol{C}}_{\boldsymbol{Cam}}))^{-1}$

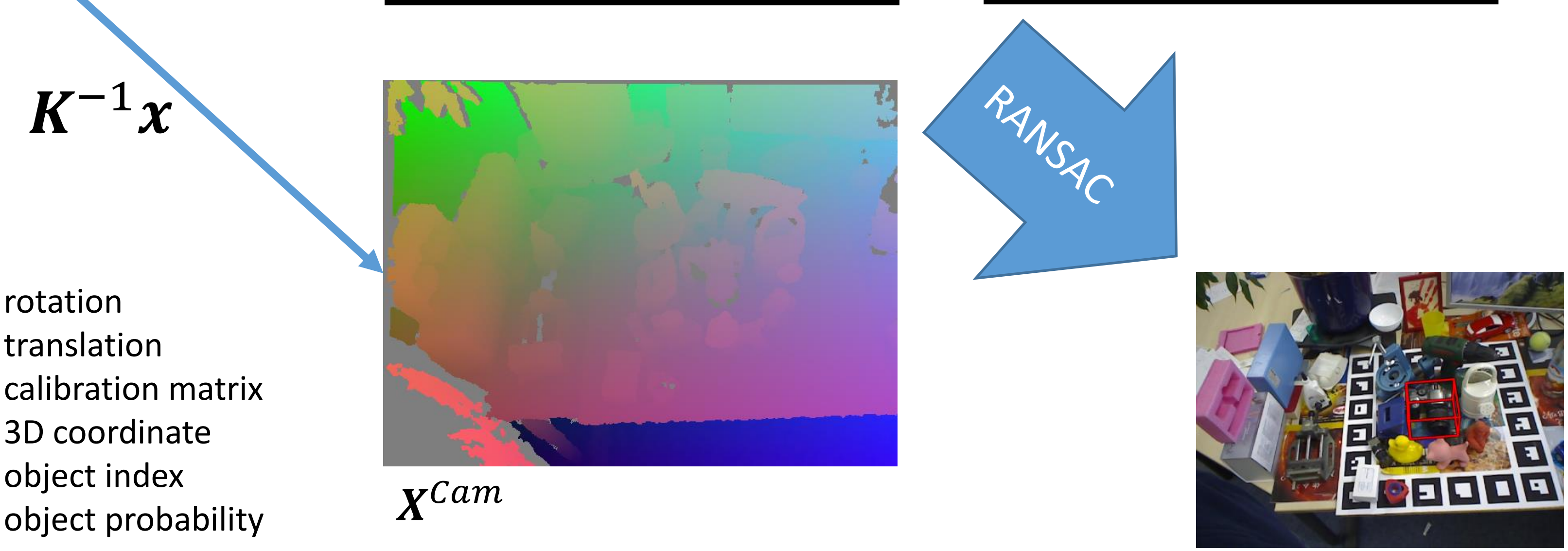


$\boldsymbol{R}$	… rotation
$\boldsymbol{t}, \boldsymbol{C}$	… translation
$\boldsymbol{K}$	… calibration matrix
$\boldsymbol{X}$	… 3D coordinate
c	… object index
$p(c)$	… object probability

Why not use camera re-localization?

- $\boldsymbol{R_{Obj}}(\boldsymbol{I}_{3\times3}| - \tilde{\boldsymbol{t}}_{\boldsymbol{Obj}}) = (\boldsymbol{R_{Cam}}(\boldsymbol{I}_{3\times3}| - \tilde{\boldsymbol{C}}_{\boldsymbol{Cam}}))^{-1}$





$\boldsymbol{R}$	… rotation
$\boldsymbol{t}, \boldsymbol{C}$	… translation
$\boldsymbol{K}$	… calibration matrix
$\boldsymbol{X}$	… 3D coordinate
c	… object index
$p(c)$	… object probability

Regression-Decision Forest

- What has to be defined?
  - Features: Features: f^{DA-D}, f^{DA-RGB}
  - Split score: Proxy classes

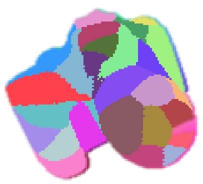
Ape

Class 1-125

Bench Vise

Class 126-250

Camera

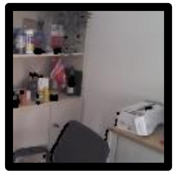


Class 251-375

Can

Class 376-500

Background

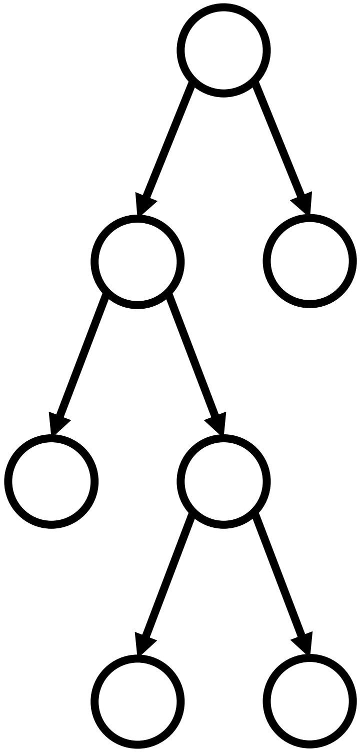


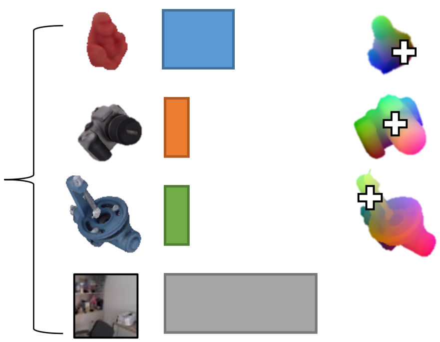
Class 0

  - Stopping criterium: Max. tree depth
  - Leaf distributions: ?

Leaf Distributions

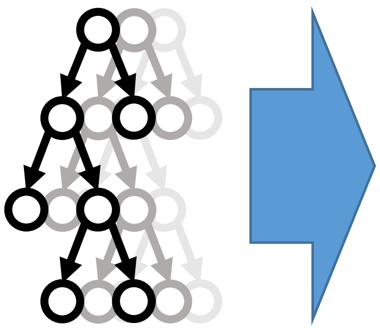
- For each object:
  - Store object probability
  - Find modes
  - Store mean of largest mode





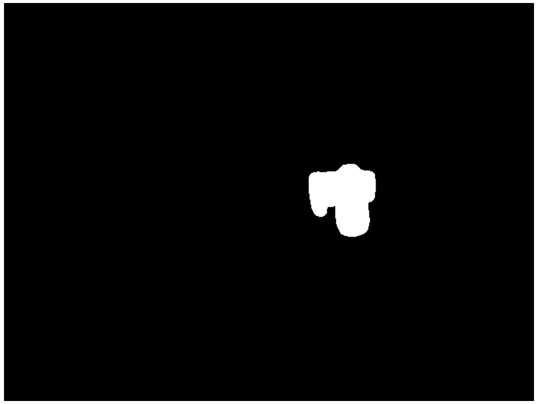
$p(c)$ $\quad$ $\boldsymbol{X}^{Obj}$ $\quad$ Formally: $p(\boldsymbol{X}^{Obj}|c)$

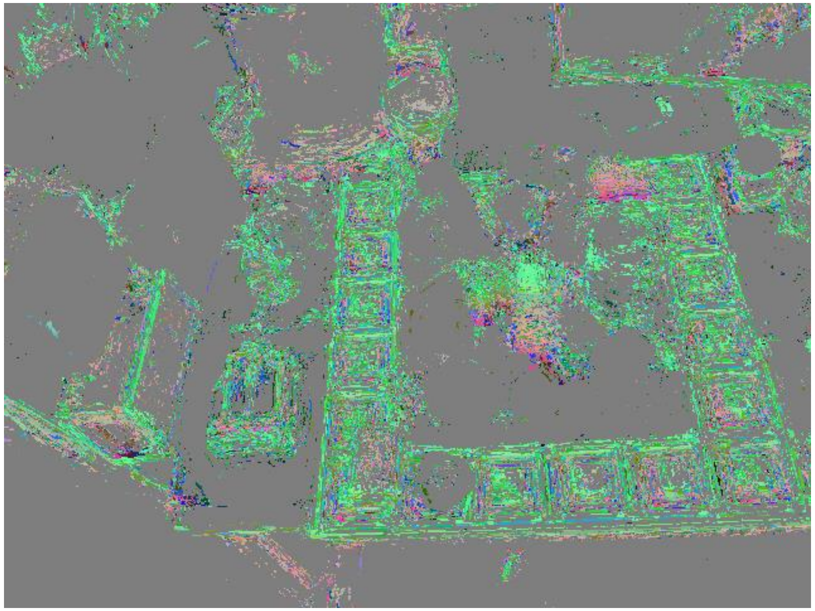
Forest Prediction

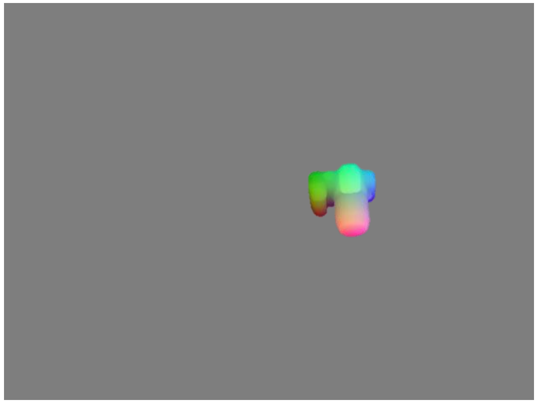


$p(c)$

Ground Truth







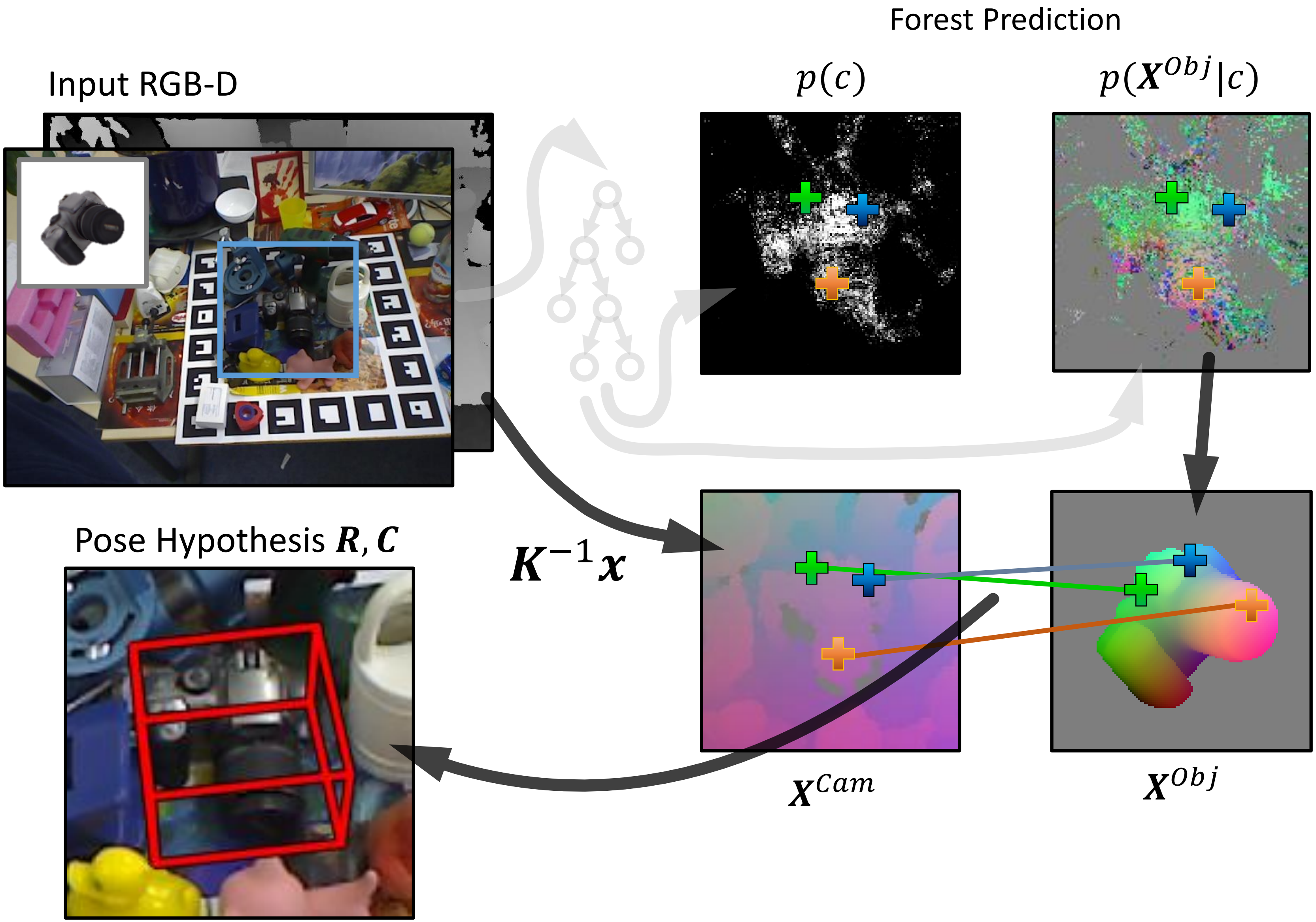
$p(\boldsymbol{X}^{Obj})$ (tree 1)

RANSAC

$\boldsymbol{R}^*, \boldsymbol{C}^*$
$E_c^* \coloneqq 0$
repeat n times:
    sample $\left\{(\boldsymbol{X}_1^{Cam}, \boldsymbol{X}_1^{Obj}), (\boldsymbol{X}_2^{Cam}, \boldsymbol{X}_2^{Obj}), (\boldsymbol{X}_3^{Cam}, \boldsymbol{X}_3^{Obj})\right\}$
    $\boldsymbol{R}, \boldsymbol{C} \coloneqq$ solve Kabsch
    calculate $E_c(\boldsymbol{R}, \boldsymbol{C})$
    if $E_c(\boldsymbol{R}, \boldsymbol{C}) < E_c^*$:
        $E_c^* \coloneqq E_c(\boldsymbol{R}, \boldsymbol{C})$
        $\boldsymbol{R}^* \coloneqq \boldsymbol{R}, \boldsymbol{C}^* \coloneqq \boldsymbol{C}$
refine $\boldsymbol{R}^*, \boldsymbol{C}^*$

c … object index
E_c … energy
τ … inlier threshold
I … set of image pixels
T … set of regression trees

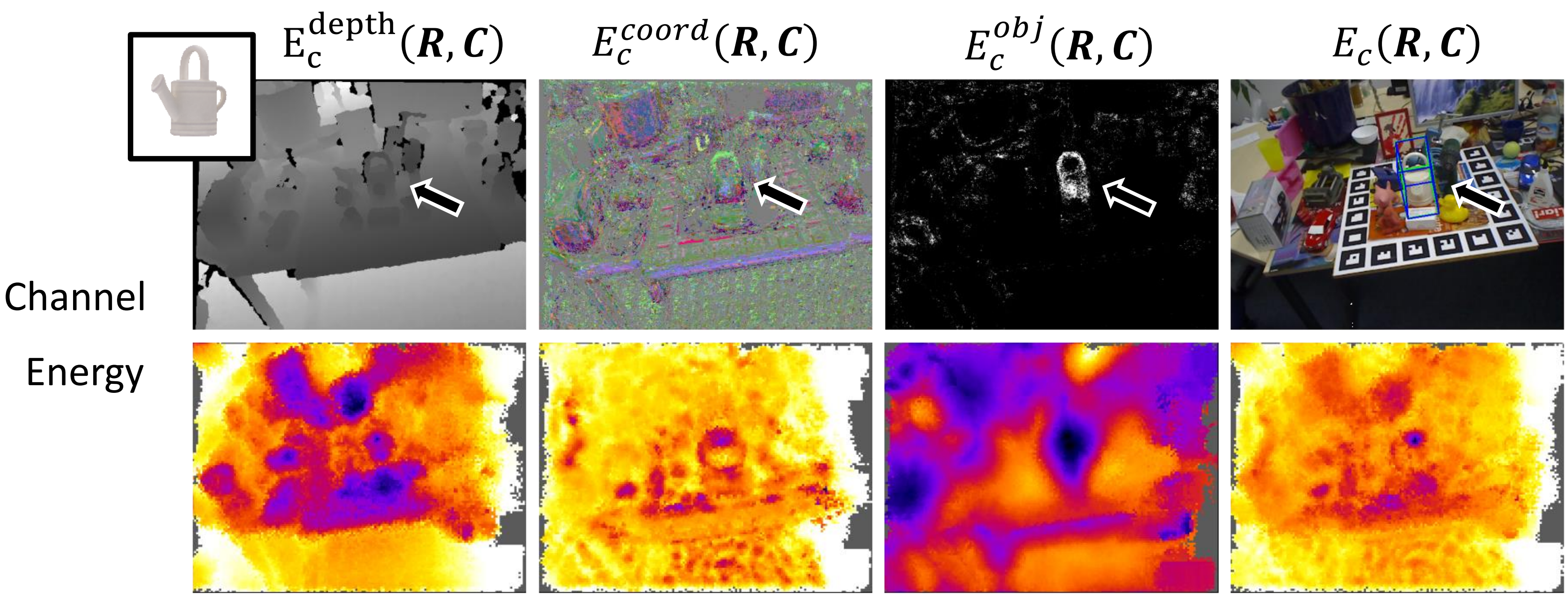
Sampling a Pose Hypothesis



Energy Formulation

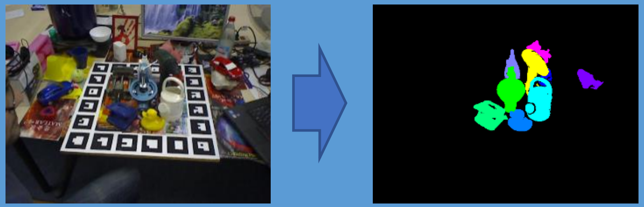
- Arbitrary 6DoF pose hypotheses $\boldsymbol{R}, \boldsymbol{C}$ is scored according to:

$$E_c(\boldsymbol{R}, \boldsymbol{C}) = \underbrace{\lambda^{\mathrm{depth}} \mathrm{E}_{\mathrm{c}}^{\mathrm{depth}}(\boldsymbol{R}, \boldsymbol{C})}_{\text{Comparison of depth}} + \underbrace{\lambda^{coord} E_c^{coord}(\boldsymbol{R}, \boldsymbol{C}) + \lambda^{obj} E_c^{obj}(\boldsymbol{R}, \boldsymbol{C})}_{\text{Comparison to forest prediction}}$$

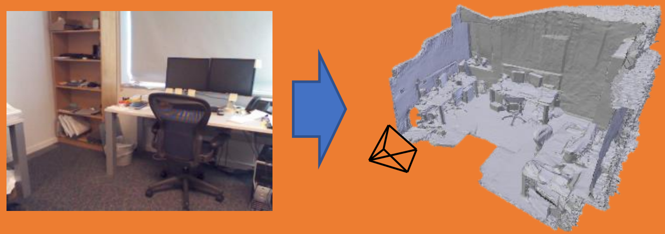


Case Study: Pose Estimation and Pose Tracking

Image Segmentation:
- Decision Forest
- Features, Training Recap

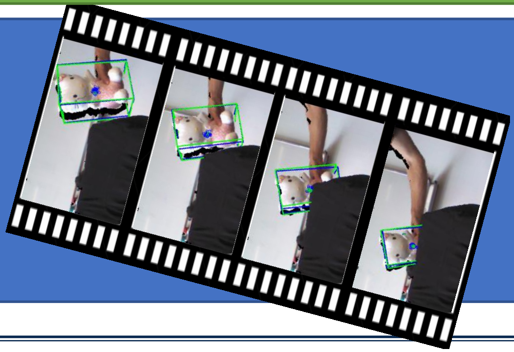


Camera Re-Localization:
- Regression Forest
- Split-Criteria, Kabsch, RANSAC



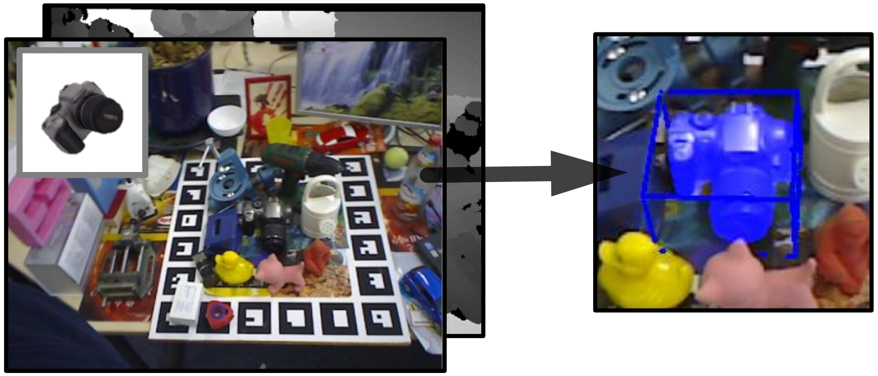
One-Shot Pose Estimation:
- Decision-Regression Forest
- Hypothesis Sampling, Hypotheses Energy

Object Pose Tracking
- Particle Filter
- Importance Sampling



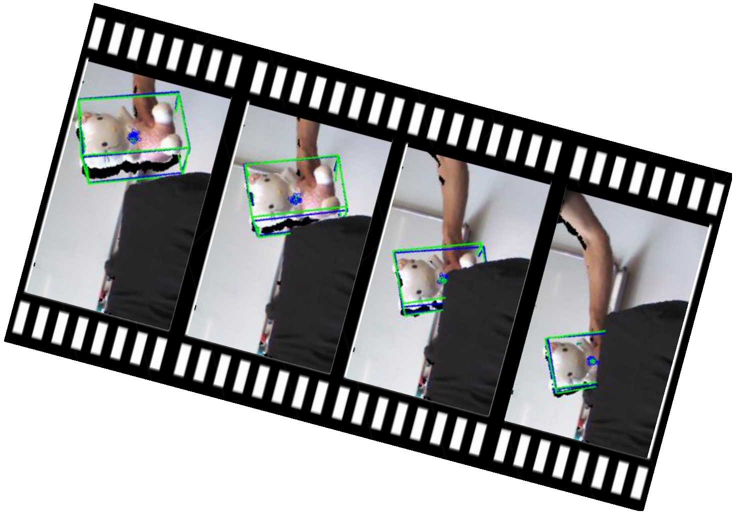
Pose Estimation vs. Pose Tracking

One Shot Pose Estimation [1]



- Estimate 6D Pose from **single** RGB-D image
- Use Object Coordinate Regression

Pose Tracking [2]



- **Stream** of RDB-D images
- Use information from previous frames:
  - Realtime
  - Increase robustness, accuracy

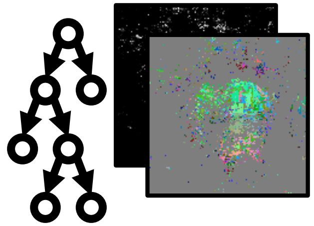
[1] Brachmann, E., Krull, A., Michel, F., Shotton, J., Gumhold, S., Rother, C.: Learning 6d object pose estimation using 3d object coordinates, ECCV (2014)

[2] Krull, A., Michel, F., Brachmann, E., Gumhold, S., Ihrke, S., Rother, C.: 6-DOF Model Based Tracking via Object Coordinate Regression, ACCV (2014)

How to Adapt it for Pose Tracking?

Ingredients:

Object Coordinate Regression



Particle Filter

Observation Model:

Energy

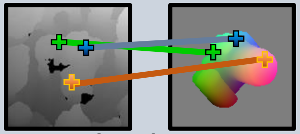
Motion Model:

- Assume continous motion

Proposal Distribution:

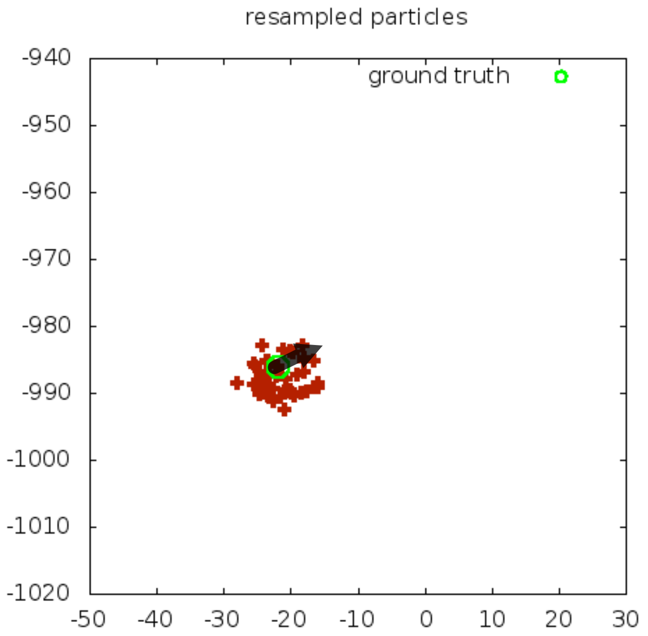
- Find rough estimate using

Efficient Search

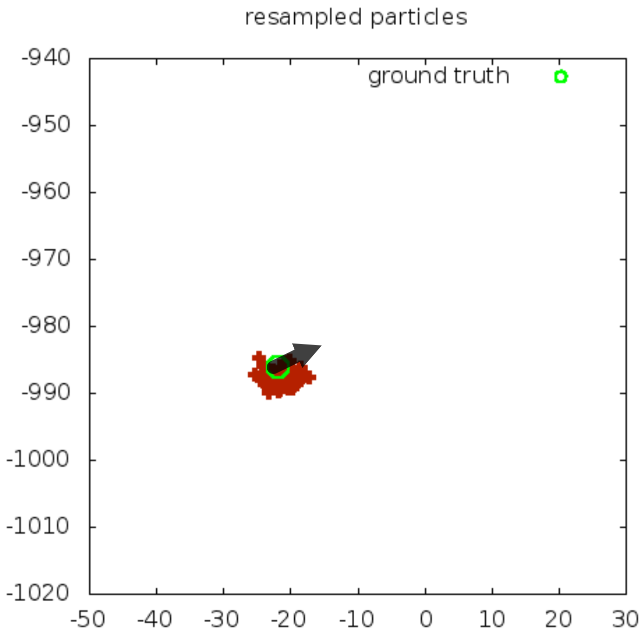


- Concentrate samples around rough estimate

Filtering with Object Coordinates

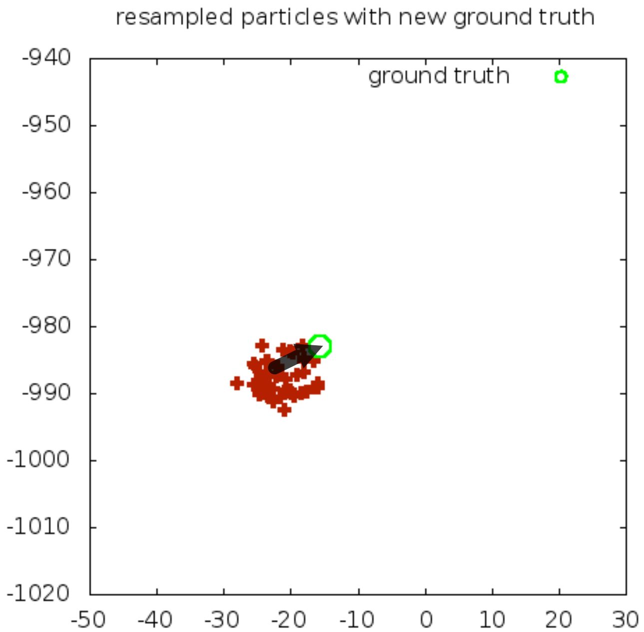


Sampling from Motion Model

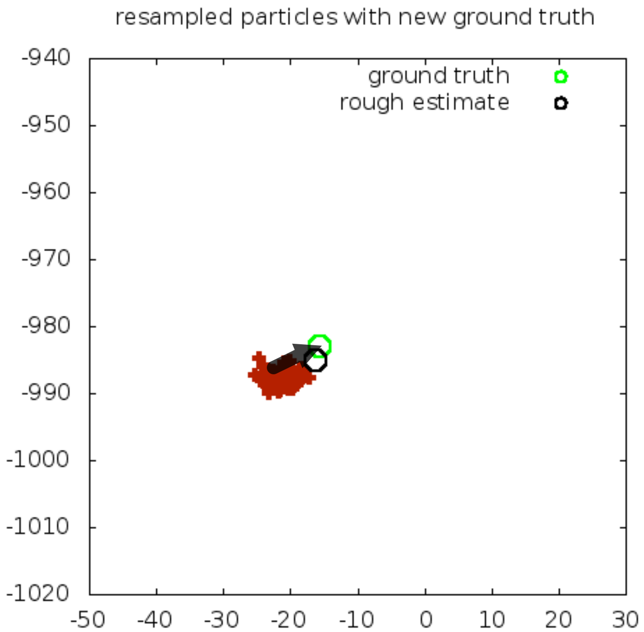


Sampling from Proposal Distribution

Filtering with Object Coordinates

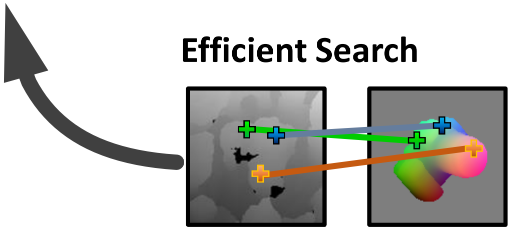


Sampling from Motion Model

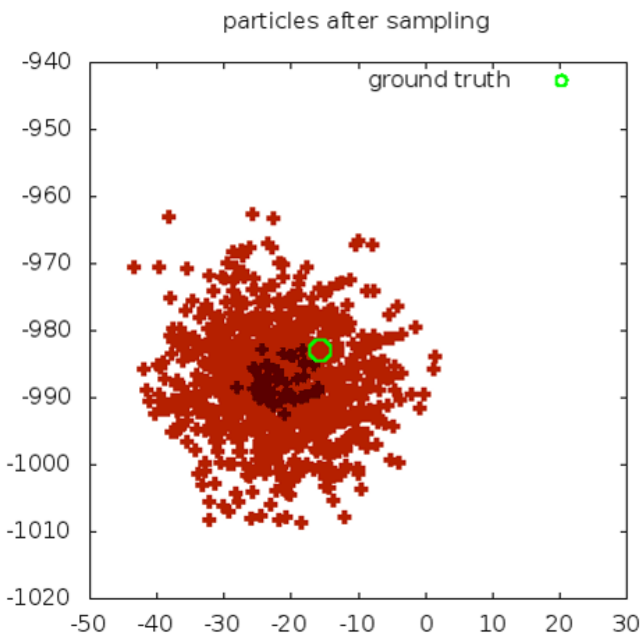


Sampling from Proposal Distribution

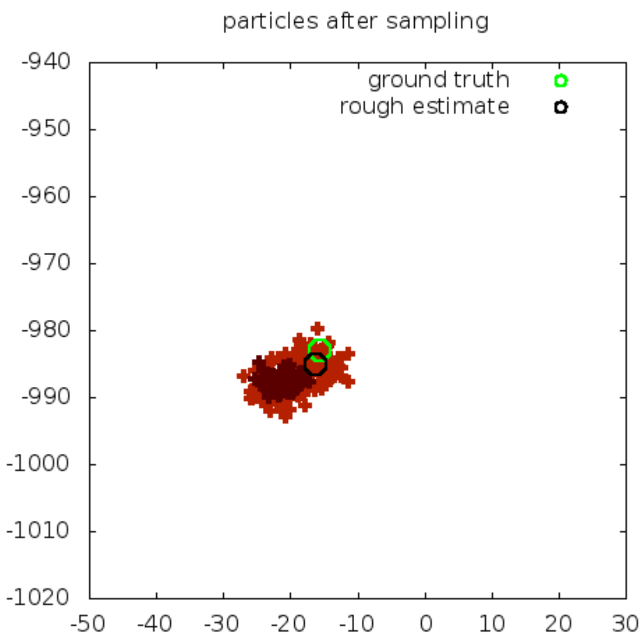
- Find a rough estimate for current pose



Filtering with Object Coordinates



Sampling from Motion Model

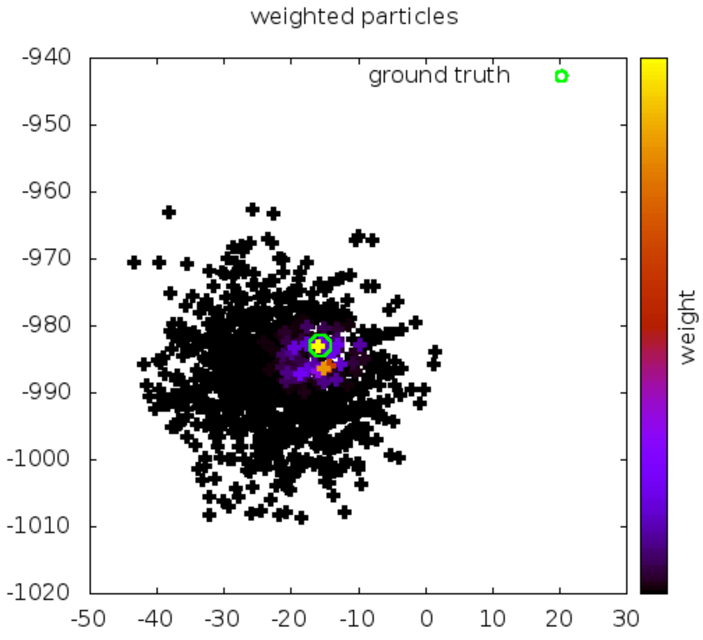


Sampling from Proposal Distribution

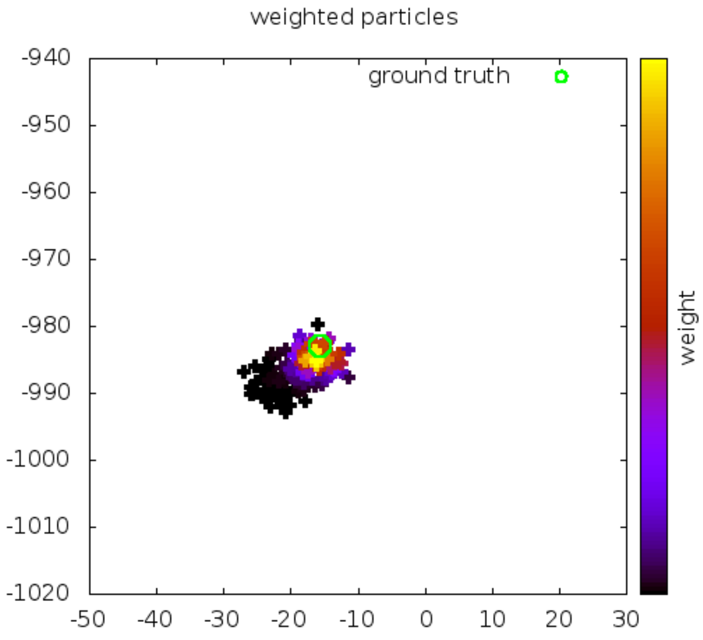
$p(x_{t+1}|x_t)$

$q(x_{t+1}|x_t)$

Filtering with Object Coordinates



Sampling from Motion Model



Sampling from Proposal Distribution

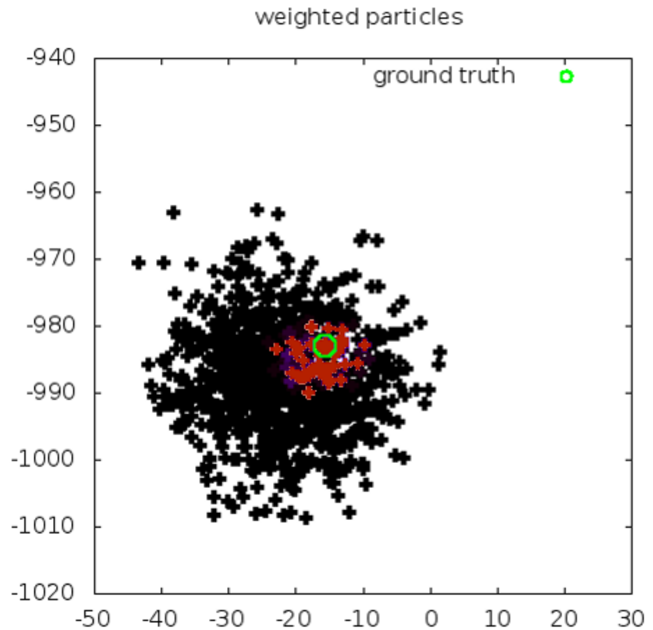
- Most samples have a very low weight

$$w_t^i = p(z_t|x_t = \tilde{x}_t^i)$$

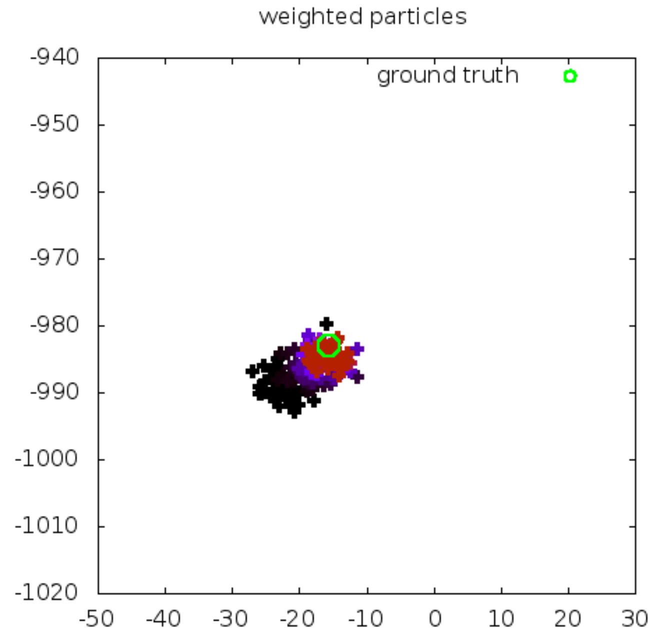
- Few samples have very low weight

$$w_t^i = p(z_t|x_t = \tilde{x}_t^i)\frac{p\left(x_t = \tilde{x}_t^i\ |x_{t-1} = \tilde{x}_{t-1}^i\right)}{q\left(x_t = \tilde{x}_t^i\ |x_{t-1} = \tilde{x}_{t-1}^i\right)}$$

Filtering with Object Coordinates



Sampling from Motion Model



Sampling from Proposal Distribution

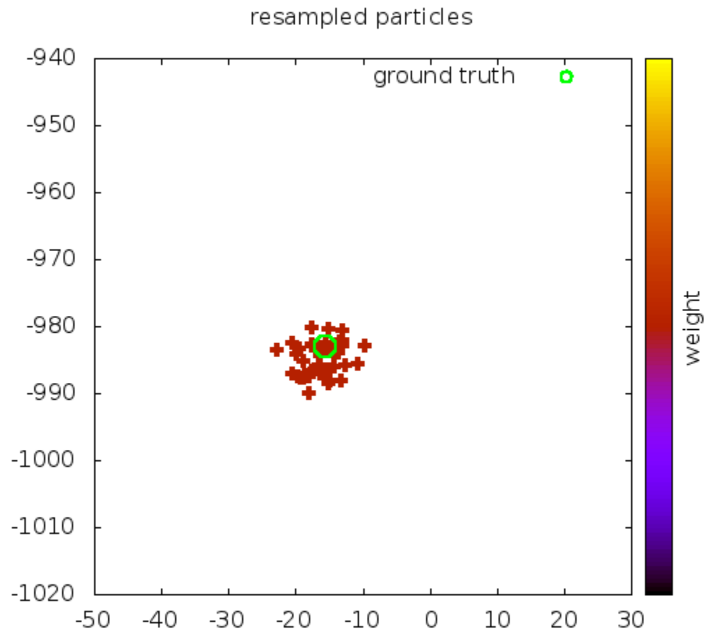
- Most samples have a very low weight

$$w_t^i = p(z_t | x_t = \tilde{x}_t^i)$$

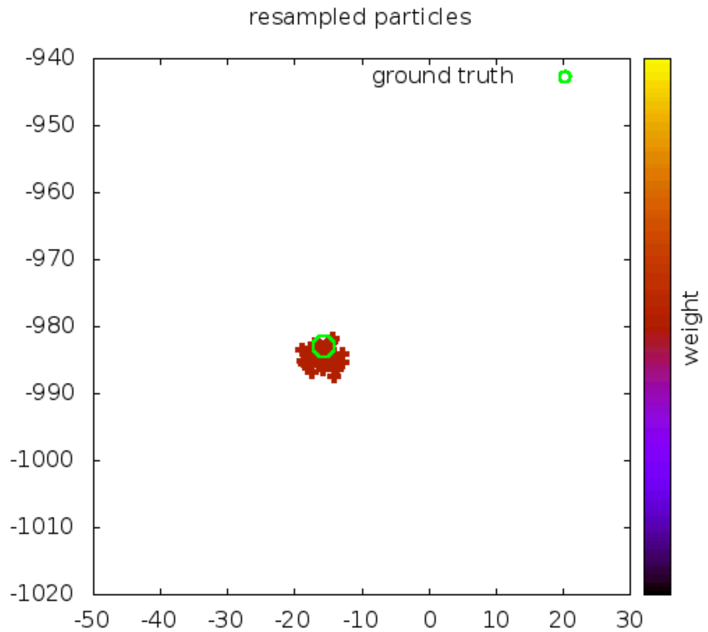
- Few samples have very low weight

$$w_t^i = p(z_t | x_t = \tilde{x}_t^i) \frac{p(x_{t+1} | x_t = \hat{x}_t^i)}{q(x_{t+1} | x_t = \hat{x}_t^i)}$$

Filtering with Object Coordinates



Sampling from Motion Model

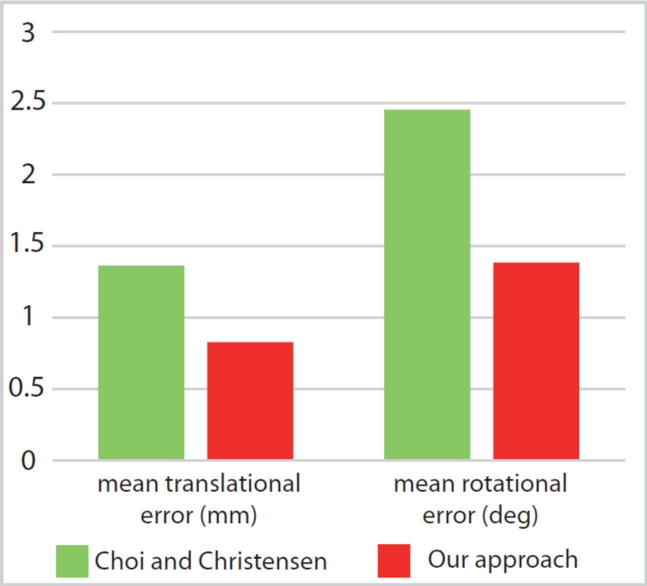


Sampling from Proposal Distribution

- More efficient
- Number of Particles can be reduced
- Frame rate can be increased

Evaluation: Choi and Christensen's Dataset [2]

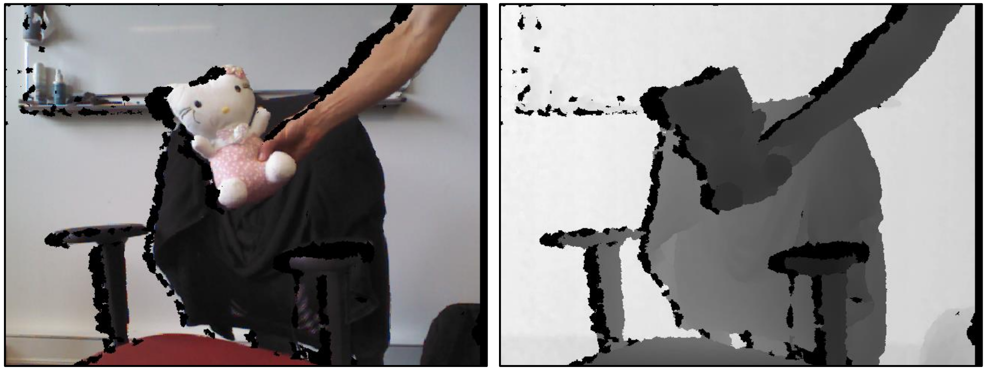
- A total of 4 synthetic sequences with 4 objects
- Objects placed in static environment
- Camera moving around object
- Partial occlusion
- Very exact ground truth



[2] Changhyun Choi, Henrik I. Christensen, RGB-D Object Tracking: A Particle Filter Approach on GPU, IROS, 2013

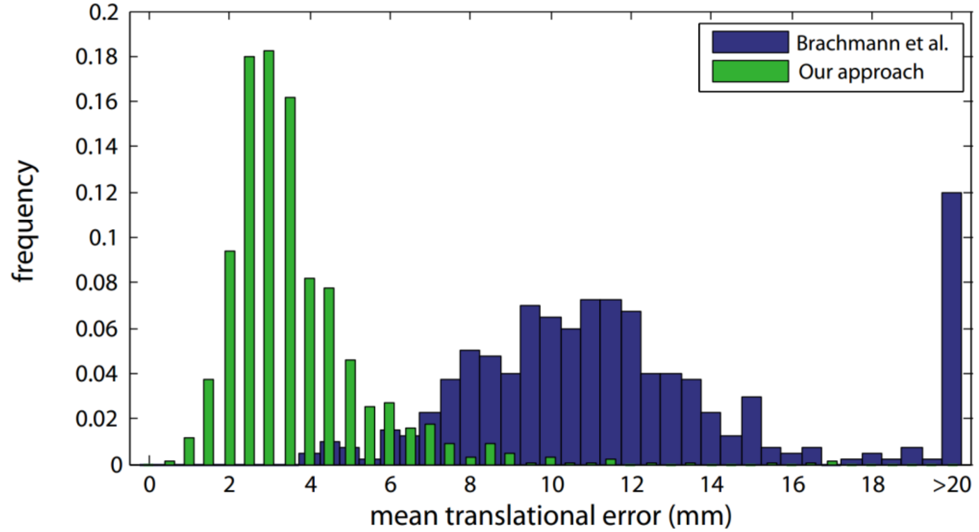
Evaluation: Our Dataset

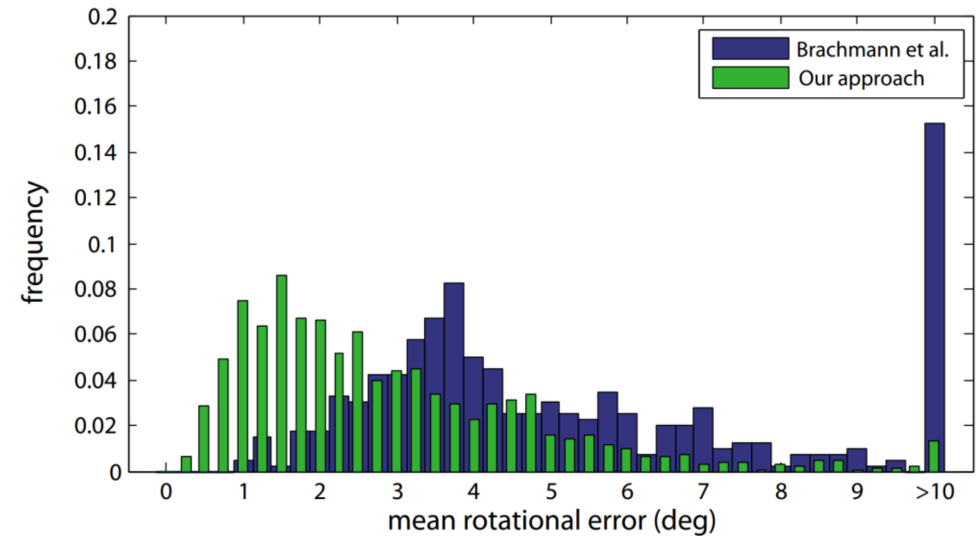
- A total of 6 captured sequences of with 3 objects
- Manually annotated ground truth
- Moving objects in front of dynamic background
- Fast erratic movement
- Strong occlusions



Evaluation: Our Dataset

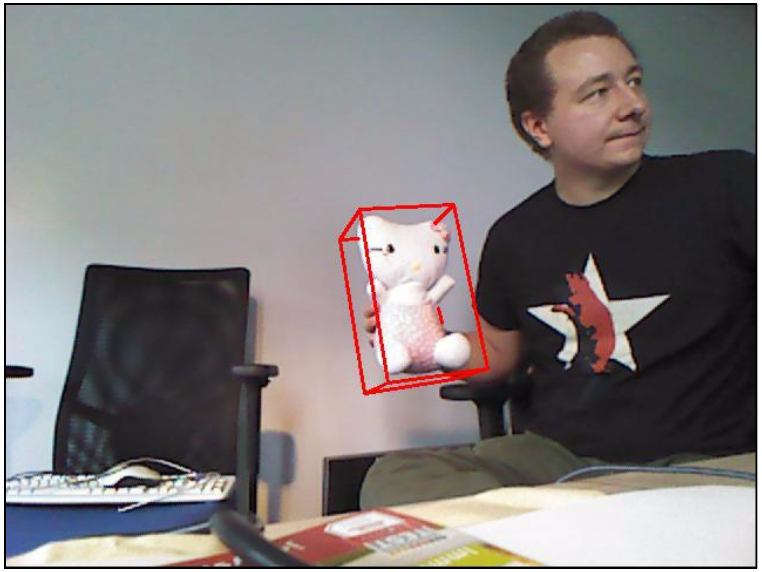
- Compared to [1] applied to each frame separately
- Lower average error
- Almost no outliers





Conclusion

- We have adapted the system from [1] for real time pose tracking
- We have designed a proposal distribution making efficient use of object coordinates
- Our method is robust against:
  - Quick motion
  - Strong occlusion
  - Shadows and changing light conditions
  - Deformation



[1] Brachmann, E., Krull, A., Michel, F., Shotton, J., Gumhold, S., Rother, C.: Learning 6d Object Pose Estimation Using 3d Object Coordinates, ECCV '14 (2014)

Recent Developments

- Articulated objects

- Learning the energy function

- Pose estimation from RGB