Sequential Monte Carlo for probabilistic object detection in images

Tim White

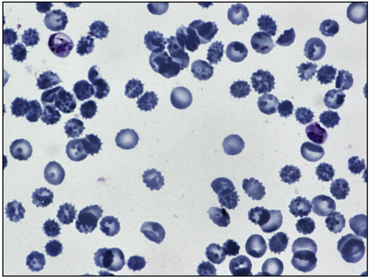
Joint work with Jeffrey Regier

Department of Statistics, University of Michigan

Joint Statistical Meetings
August 5th, 2024

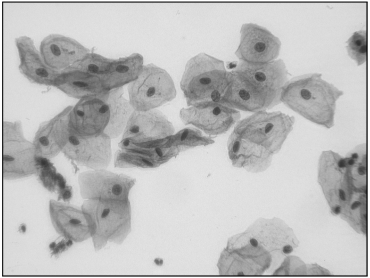
Introduction

Small object detection in biology



Malaria-infected blood cells

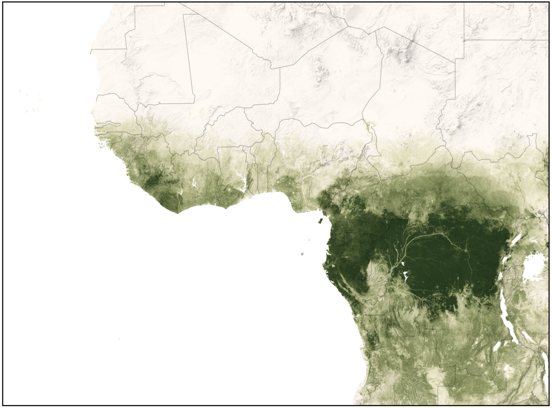
Broad Bioimage Benchmark Collection, BBBC041
Ljosa et al., 2012



Cancerous cervical cells

Cervix93 cytology dataset
Phoulady and Mouton, 2018

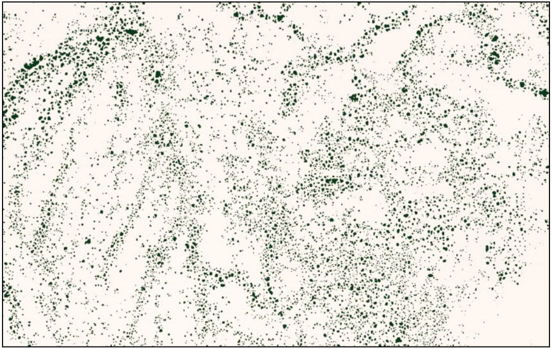
Small object detection in remote sensing



Forest cover in West Africa

Brandt, et al. An unexpectedly large count of trees in the West African Sahara and Sahel. Nature, 2020.

Small object detection in remote sensing



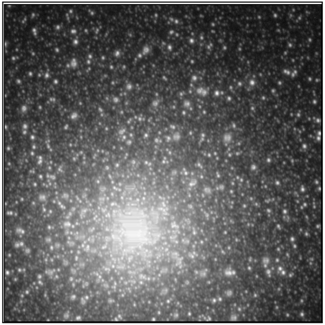
Forest cover in West Africa

Brandt, et al. An unexpectedly large count of trees in the West African Sahara and Sahel. Nature, 2020.

Small object detection in astronomy

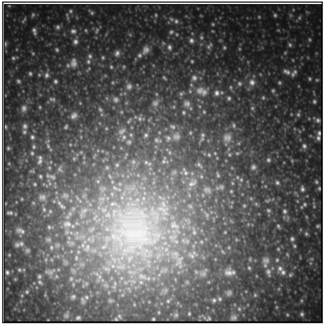
The Messier 15 globular cluster, imaged by Hubble/SDSS

Small object detection in astronomy



The Messier 15 globular cluster, imaged by Hubble/SDSS

Small object detection in astronomy

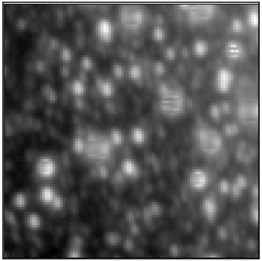


The Messier 15 globular cluster, imaged by Hubble/SDSS

- **Astronomical cataloging** is the task of inferring the properties of stars, galaxies, and other objects in astronomical images

Challenges of astronomical cataloging

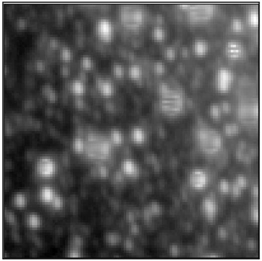
- **Massive amount of data**, typically with no ground truth



100 x 100 pixel subregion of Messier 15

Challenges of astronomical cataloging

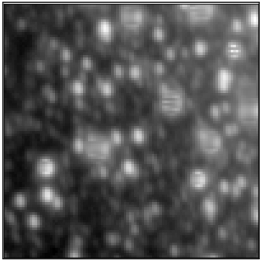
- **Massive amount of data**, typically with no ground truth
- Objects may be **faint** and might **visually overlap** with one another



100 x 100 pixel subregion of Messier 15

Challenges of astronomical cataloging

- **Massive amount of data**, typically with no ground truth
- Objects may be **faint** and might **visually overlap** with one another
- Requires **transdimensional** inference
  - True number of objects is unknown
  - Properties are ambiguous



100 x 100 pixel subregion of Messier 15

Existing approaches to object detection

Non-probabilistic

- Use deterministic algorithm to make single-catalog estimates
- ~~Calibrated uncertainty~~

Methods:

- Threshold + watershed
- (Many) CNN-based methods
- Source Extractor

Existing approaches to object detection

Non-probabilistic

- Use deterministic algorithm to make single-catalog estimates
- ~~Calibrated uncertainty~~

Methods:

- Threshold + watershed
- (Many) CNN-based methods
- Source Extractor

Probabilistic

- Infer a posterior distribution over all possible catalogs
- Calibrated uncertainty

Methods:

- Markov chain Monte Carlo
  Sample catalogs from the posterior
- Variational inference
  Optimize an approximate posterior

Our contribution

- We propose a probabilistic method for small object detection based on **sequential Monte Carlo (SMC)**

Our contribution

- We propose a probabilistic method for small object detection based on **sequential Monte Carlo (SMC)**

  Our algorithm...

  - runs in parallel on **tiles** (i.e., subregions) of an image

Our contribution

- We propose a probabilistic method for small object detection based on **sequential Monte Carlo (SMC)**

  Our algorithm...

  - runs in parallel on **tiles** (i.e., subregions) of an image
  - leverages GPUs to efficiently evaluate latent variable catalogs

Our contribution

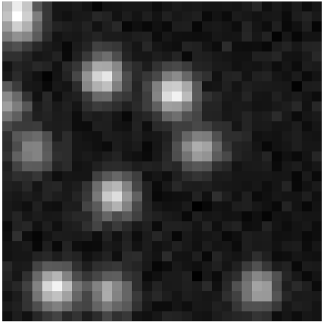
- We propose a probabilistic method for small object detection based on **sequential Monte Carlo (SMC)**

  Our algorithm...

  - runs in parallel on **tiles** (i.e., subregions) of an image
  - leverages GPUs to efficiently evaluate latent variable catalogs
  - performs transdimensional inference without transdimensional sampling

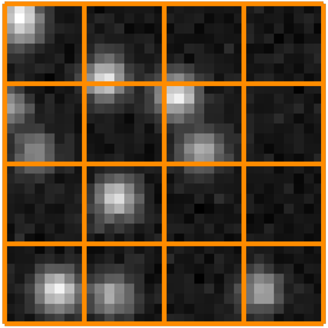
Our contribution

- Tasks for the remainder of this talk:



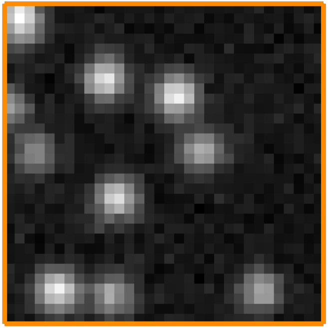
Our contribution

- Tasks for the remainder of this talk:
  1. Introduce an SMC sampler for detecting objects in each tile



Our contribution

- Tasks for the remainder of this talk:
  1. Introduce an SMC sampler for detecting objects in each tile
  2. Combine the tile-level catalogs via divide-and-conquer SMC



An SMC sampler for one tile

Model

- **Image** x with a height of H pixels and a width of W pixels

Model

- **Image** x with a height of H pixels and a width of W pixels
- Prior
  - **Number of objects** $s \sim \text{Uniform}\{0, 1, 2, \ldots, s_{\text{max}}\}$
  - Given s, **locations** $\ell_1, \ldots, \ell_s \overset{\text{iid}}{\sim} \text{Uniform}\big([0, H] \times [0, W]\big)$

    features $f_1, \ldots, f_s \overset{\text{iid}}{\sim} \mathcal{F}(\cdot)$
  - **Catalog** $z = \{s\} \cup \{\ell_j, f_j\}_{j=1}^{s}$

Model

- **Image** x with a height of H pixels and a width of W pixels
- Prior
  - **Number of objects** $s \sim \text{Uniform}\{0, 1, 2, \ldots, s_{\max}\}$
  - Given s, **locations** $\ell_1, \ldots, \ell_s \overset{\text{iid}}{\sim} \text{Uniform}\big([0, H] \times [0, W]\big)$

    features $f_1, \ldots, f_s \overset{\text{iid}}{\sim} \mathcal{F}(\cdot)$
  - **Catalog** $z = \{s\} \cup \{\ell_j, f_j\}_{j=1}^{s}$
- Likelihood
  - **Intensity** at pixel (h, w) is $x_{hw} \mid z \sim \text{Poisson}(\lambda_{hw}(z))$
    - $\lambda_{hw}(z) = \text{background intensity} + \text{function of features at } (h, w)$

Model

- **Image** x with a height of H pixels and a width of W pixels
- Prior
  - **Number of objects** $s \sim \text{Uniform}\{0, 1, 2, \dots, s_{\text{max}}\}$
  - Given s, **locations** $\ell_1, \dots, \ell_s \overset{\text{iid}}{\sim} \text{Uniform}\big([0, H] \times [0, W]\big)$

    features $f_1, \dots, f_s \overset{\text{iid}}{\sim} \mathcal{F}(\cdot)$
  - **Catalog** $z = \{s\} \cup \{\ell_j, f_j\}_{j=1}^{s}$
- Likelihood
  - **Intensity** at pixel (h, w) is $x_{hw} \mid z \sim \text{Poisson}(\lambda_{hw}(z))$
    - $\lambda_{hw}(z) = \text{background intensity} + \text{function of features at } (h, w)$
- Posterior $p(z \mid x) \propto p(z)p(x \mid z)$

SMC sampler

Ingredient #1: Likelihood tempering

- Cannot directly sample catalogs from $p(z \mid x)$
- Define a sequence of auxiliary distributions $p(z)p(x \mid z)^{\tau}$
  - Increase temperature τ from 0 (prior) to 1 (posterior)

SMC sampler

Ingredient #1: Likelihood tempering

- Cannot directly sample catalogs from $p(z \mid x)$
- Define a sequence of auxiliary distributions $p(z)p(x \mid z)^{\tau}$
  - Increase temperature τ from 0 (prior) to 1 (posterior)
- Procedure:

SMC sampler

Ingredient #1: Likelihood tempering

- Cannot directly sample catalogs from $p(z \mid x)$
- Define a sequence of auxiliary distributions $p(z)p(x \mid z)^\tau$
  - Increase temperature τ from 0 (prior) to 1 (posterior)
- Procedure:
  - Sample N catalogs $z^{1:N} \overset{\text{iid}}{\sim} p(z)$ and initialize weights $w^{1:N} = \frac{1}{N}$

SMC sampler

Ingredient #1: Likelihood tempering

- Cannot directly sample catalogs from $p(z \mid x)$
- Define a sequence of auxiliary distributions $p(z)p(x \mid z)^\tau$
  - Increase temperature τ from 0 (prior) to 1 (posterior)
- Procedure:
  - Sample N catalogs $z^{1:N} \overset{\text{iid}}{\sim} p(z)$ and initialize weights $w^{1:N} = \frac{1}{N}$

  While $\tau < 1$:

SMC sampler

Ingredient #1: Likelihood tempering

- Cannot directly sample catalogs from $p(z \mid x)$
- Define a sequence of auxiliary distributions $p(z)p(x \mid z)^\tau$
  - Increase temperature τ from 0 (prior) to 1 (posterior)
- Procedure:
  - Sample N catalogs $z^{1:N} \overset{\text{iid}}{\sim} p(z)$ and initialize weights $w^{1:N} = \frac{1}{N}$

  While $\tau < 1$:

  - **Resample** the catalogs to avoid weight degeneracy

SMC sampler

Ingredient #1: Likelihood tempering

- Cannot directly sample catalogs from $p(z \mid x)$
- Define a sequence of auxiliary distributions $p(z)p(x \mid z)^\tau$
  - Increase temperature τ from 0 (prior) to 1 (posterior)
- Procedure:
  - Sample N catalogs $z^{1:N} \overset{\text{iid}}{\sim} p(z)$ and initialize weights $w^{1:N} = \frac{1}{N}$

  While $\tau < 1$:
  - **Resample** the catalogs to avoid weight degeneracy
  - **Mutate** the catalogs using an MCMC kernel

SMC sampler

Ingredient #1: Likelihood tempering

- Cannot directly sample catalogs from $p(z \mid x)$
- Define a sequence of auxiliary distributions $p(z)p(x \mid z)^\tau$
  - Increase temperature τ from 0 (prior) to 1 (posterior)
- Procedure:
  - Sample N catalogs $z^{1:N} \overset{\text{iid}}{\sim} p(z)$ and initialize weights $w^{1:N} = \frac{1}{N}$

  While $\tau < 1$:
  - **Resample** the catalogs to avoid weight degeneracy
  - **Mutate** the catalogs using an MCMC kernel
  - **Increase temperature**: $\tau \leftarrow \tau_{\text{previous}} + \delta$ for a suitable $\delta > 0$

SMC sampler

Ingredient #1: Likelihood tempering

- Cannot directly sample catalogs from $p(z \mid x)$
- Define a sequence of auxiliary distributions $p(z)p(x \mid z)^{\tau}$
  - Increase temperature τ from 0 (prior) to 1 (posterior)
- Procedure:
  - Sample N catalogs $z^{1:N} \overset{\text{iid}}{\sim} p(z)$ and initialize weights $w^{1:N} = \frac{1}{N}$

  While $\tau < 1$:
  - **Resample** the catalogs to avoid weight degeneracy
  - **Mutate** the catalogs using an MCMC kernel
  - **Increase temperature**: $\tau \leftarrow \tau_{\text{previous}} + \delta$ for a suitable $\delta > 0$
  - **Reweight**: $\forall n, \quad w^n \propto p(x \mid z^n)^{\tau - \tau_{\text{previous}}}$

SMC sampler

Ingredient #1: Likelihood tempering

- Cannot directly sample catalogs from $p(z \mid x)$
- Define a sequence of auxiliary distributions $p(z)p(x \mid z)^{\tau}$
  - Increase temperature τ from 0 (prior) to 1 (posterior)
- Procedure:
  - Sample N catalogs $z^{1:N} \overset{\text{iid}}{\sim} p(z)$ and initialize weights $w^{1:N} = \frac{1}{N}$

  While $\tau < 1$:
  - **Resample** the catalogs to avoid weight degeneracy
  - **Mutate** the catalogs using an MCMC kernel
  - **Increase temperature**: $\tau \leftarrow \tau_{\text{previous}} + \delta$ for a suitable $\delta > 0$
  - **Reweight**: $\forall n, \quad w^n \propto p(x \mid z^n)^{\tau - \tau_{\text{previous}}}$

  Output: **Weighted catalogs** $\{w^n, z^n\}_{n=1}^N \sim p(z \mid x)$

SMC sampler

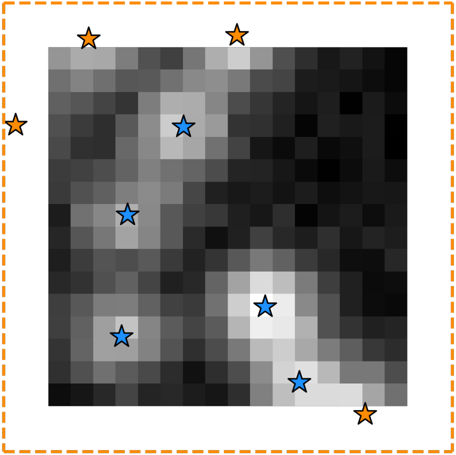
Ingredient #2: Stratification by number of objects

- Resample and mutate separately among catalogs with the same number of objects s
- Number of catalogs corresponding to each $s \in \{0, 1, ..., s_{\text{max}}\}$ remains fixed through the algorithm

SMC sampler

Ingredient #3: Padded tiles

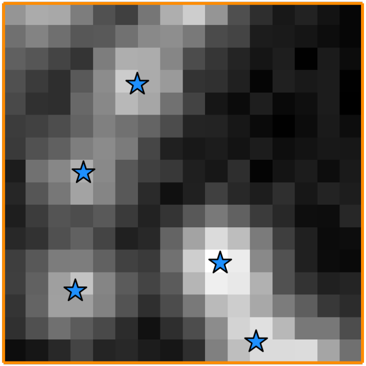
- Run sampler to generate $\{w_n, \widetilde{z}_n\}_{n=1}^N \sim p(\widetilde{z} \mid x)$, where $\widetilde{z} = z \cup z^+$
- Resample to obtain $\{1, \widetilde{z}_n\}_{n=1}^N \sim p(\widetilde{z} \mid x)$



SMC sampler

Ingredient #3: Padded tiles

- Run sampler to generate $\{w_n, \widetilde{z}_n\}_{n=1}^N \sim p(\widetilde{z} \mid x)$, where $\widetilde{z} = z \cup z^+$
- Resample to obtain $\{1, \widetilde{z}_n\}_{n=1}^N \sim p(\widetilde{z} \mid x)$
- Discard detections z^+ in the padded region to obtain $\{1, z_n\}_{n=1}^N \sim p(z \mid x)$



Case study: Crowded starfields

- 1,000 synthetic images (16 pixels $\times$ 16 pixels)
  - Up to 8 stars in each image

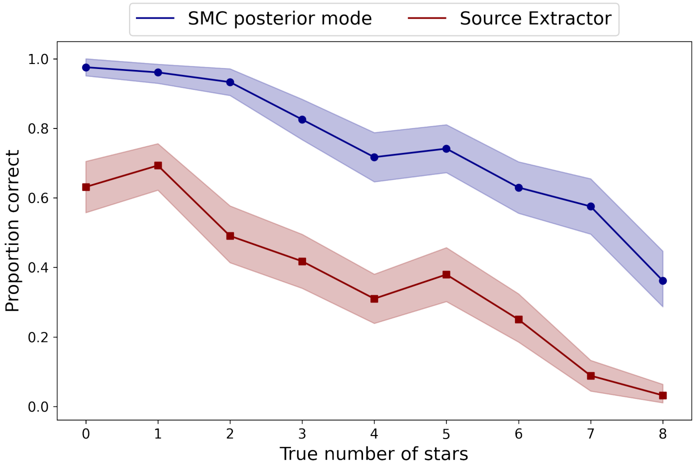
Case study: Crowded starfields

- 1,000 synthetic images (16 pixels $\times$ 16 pixels)
  - Up to 8 stars in each image
- Compare **SMC** and **Source Extractor** in terms of estimated number of stars and estimated total flux

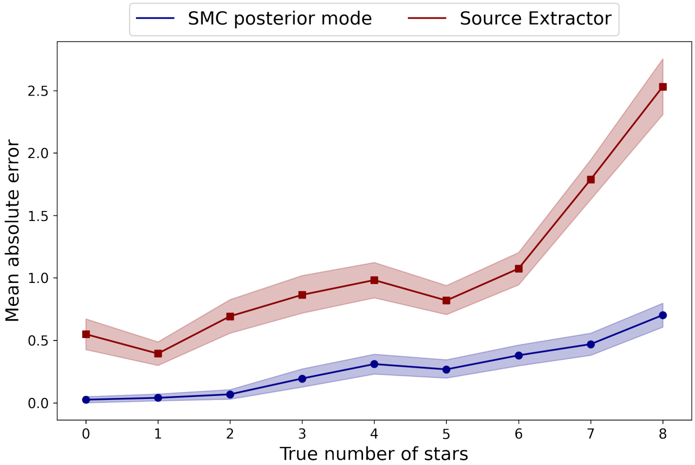
Case study: Crowded starfields

- 1,000 synthetic images (16 pixels $\times$ 16 pixels)
  - Up to 8 stars in each image
- Compare **SMC** and **Source Extractor** in terms of estimated number of stars and estimated total flux
- SMC settings:
  - 2-pixel-wide padded margin
  - Make up to 10 detections per padded image
  - 2,000 catalogs for each $s \in \{0, 1, ..., 10\}$

Accuracy of estimated number of stars



MAE of estimated number of stars



Accuracy of estimated total flux

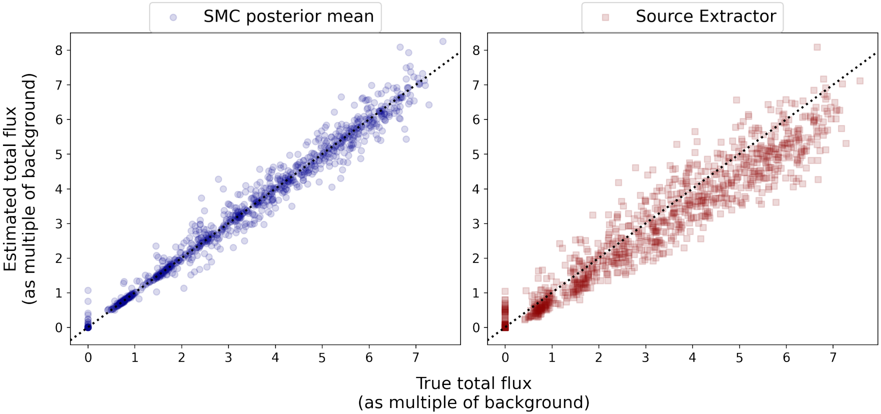


Image reconstructions

Image | One SMC catalog | Source Extractor

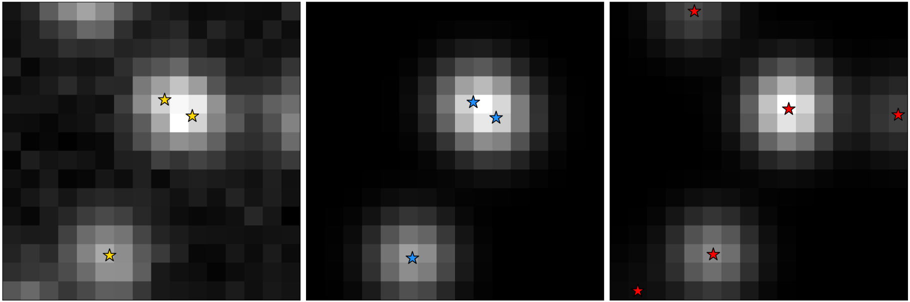
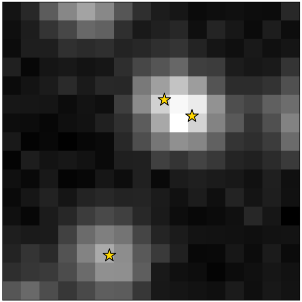
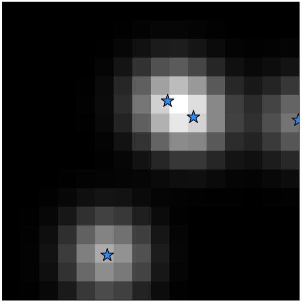


Image reconstructions

Image



One SMC catalog



Source Extractor

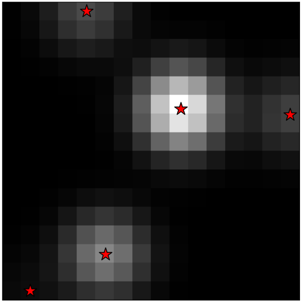
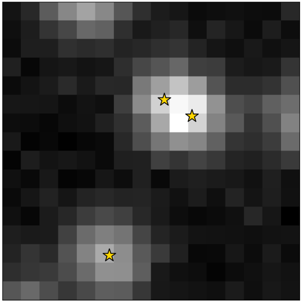
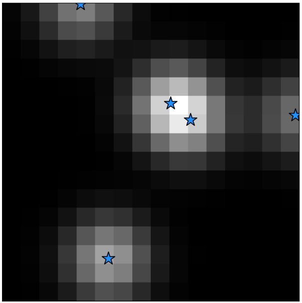


Image reconstructions

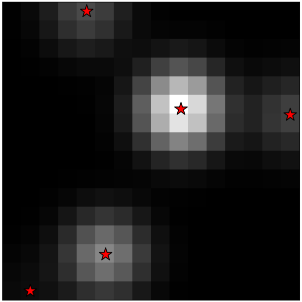
Image



One SMC catalog



Source Extractor



Combining tiles with divide-and-conquer SMC

Divide-and-conquer SMC

JOURNAL OF COMPUTATIONAL AND GRAPHICAL STATISTICS
2017, VOL. 26, NO. 2, 445–458
http://dx.doi.org/10.1080/10618600.2016.1237363

Taylor & Francis
Taylor & Francis Group

Divide-and-Conquer With Sequential Monte Carlo

F. Lindsten[a], A. M. Johansen[b], C. A. Naesseth[c], B. Kirkpatrick[d], T. B. Schön[e], J. A. D. Aston[f], and A. Bouchard-Côté[g]

[a]Division of Systems and Control, Department of Information Technology, Uppsala University, Uppsala, Sweden; [b]Department of Statistics, University of Warwick, Coventry, United Kingdom; [c]Department of Electrical Engineering, Linköping University, Linköping, Sweden; [d]Intrepid Net Computing; [e]Department of Information Technology, Uppsala University, Uppsala, Sweden; [f]University of Cambridge, Cambridge, United Kingdom; [g]Statistics, University of British Columbia, Vancouver, Canada

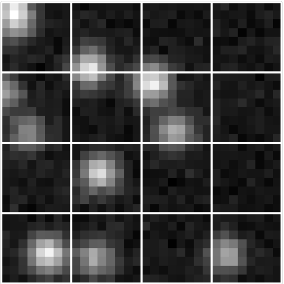
ABSTRACT

We propose a novel class of Sequential Monte Carlo (SMC) algorithms, appropriate for inference in probabilistic graphical models. This class of algorithms adopts a divide-and-conquer approach based upon an auxiliary tree-structured decomposition of the model of interest, turning the overall inferential task into a collection of recursively solved subproblems. The proposed method is applicable to a broad class of probabilistic graphical models, *including* models with loops. Unlike a standard SMC sampler, the proposed divide-and-conquer SMC employs multiple independent populations of weighted particles, which are resampled, merged, and propagated as the method progresses. We illustrate empirically that this approach can outperform standard methods in terms of the accuracy of the posterior expectation and marginal likelihood approximations. Divide-and-conquer SMC also opens up novel parallel implementation options and the possibility of concentrating the computational effort on the most challenging subproblems. We demonstrate its performance on a Markov random field and on a hierarchical logistic regression problem. Supplementary materials including proofs and additional numerical results are available online.

ARTICLE HISTORY
Received June 2015
Revised June 2016

KEYWORDS
Bayesian methods; Graphical models; Hierarchical models; Particle filters

Tree of tile-level target distributions



$$p(\widetilde{z}_{1:16} \mid x_{1:16})$$

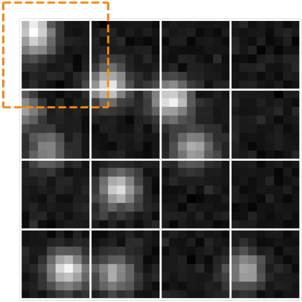
$$p(\widetilde{z}_{1:8} \mid x_{1:8}) \qquad p(\widetilde{z}_{9:16} \mid x_{9:16})$$

$$p(\widetilde{z}_{1:4} \mid x_{1:4}) \qquad p(\widetilde{z}_{5:8} \mid x_{5:8}) \qquad \cdots \qquad \cdots$$

$$p(\widetilde{z}_{1:2} \mid x_{1:2}) \qquad p(\widetilde{z}_{3:4} \mid x_{3:4}) \qquad p(\widetilde{z}_{5:6} \mid x_{5:6}) \qquad p(\widetilde{z}_{7:8} \mid x_{7:8}) \qquad \cdots$$

$$p(\widetilde{z}_1 \mid x_1) \quad p(\widetilde{z}_2 \mid x_2) \quad p(\widetilde{z}_3 \mid x_3) \quad p(\widetilde{z}_4 \mid x_4) \quad p(\widetilde{z}_5 \mid x_5) \quad p(\widetilde{z}_6 \mid x_6) \quad p(\widetilde{z}_7 \mid x_7) \quad p(\widetilde{z}_8 \mid x_8) \quad \cdots$$

Run SMC sampler in parallel on 16 tiles



$$p(\widetilde{z}_{1:16} \mid x_{1:16})$$

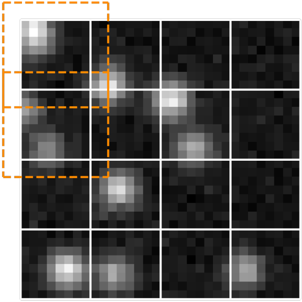
$$p(\widetilde{z}_{1:8} \mid x_{1:8}) \qquad p(\widetilde{z}_{9:16} \mid x_{9:16})$$

$$p(\widetilde{z}_{1:4} \mid x_{1:4}) \qquad p(\widetilde{z}_{5:8} \mid x_{5:8}) \qquad \cdots \qquad \cdots$$

$$p(\widetilde{z}_{1:2} \mid x_{1:2}) \qquad p(\widetilde{z}_{3:4} \mid x_{3:4}) \qquad p(\widetilde{z}_{5:6} \mid x_{5:6}) \qquad p(\widetilde{z}_{7:8} \mid x_{7:8}) \qquad \cdots$$

$$p(\widetilde{z}_{1} \mid x_{1}) \quad p(\widetilde{z}_{2} \mid x_{2}) \quad p(\widetilde{z}_{3} \mid x_{3}) \quad p(\widetilde{z}_{4} \mid x_{4}) \quad p(\widetilde{z}_{5} \mid x_{5}) \quad p(\widetilde{z}_{6} \mid x_{6}) \quad p(\widetilde{z}_{7} \mid x_{7}) \quad p(\widetilde{z}_{8} \mid x_{8}) \quad \cdots$$

Run SMC sampler in parallel on 16 tiles



$$p(\widetilde{z}_{1:16} \mid x_{1:16})$$

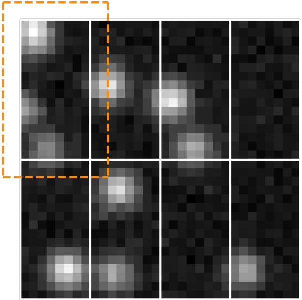
$$p(\widetilde{z}_{1:8} \mid x_{1:8}) \qquad p(\widetilde{z}_{9:16} \mid x_{9:16})$$

$$p(\widetilde{z}_{1:4} \mid x_{1:4}) \qquad p(\widetilde{z}_{5:8} \mid x_{5:8}) \qquad \cdots \qquad \cdots$$

$$p(\widetilde{z}_{1:2} \mid x_{1:2}) \qquad p(\widetilde{z}_{3:4} \mid x_{3:4}) \qquad p(\widetilde{z}_{5:6} \mid x_{5:6}) \qquad p(\widetilde{z}_{7:8} \mid x_{7:8}) \qquad \cdots$$

$$p(\widetilde{z}_1 \mid x_1) \quad p(\widetilde{z}_2 \mid x_2) \quad p(\widetilde{z}_3 \mid x_3) \quad p(\widetilde{z}_4 \mid x_4) \quad p(\widetilde{z}_5 \mid x_5) \quad p(\widetilde{z}_6 \mid x_6) \quad p(\widetilde{z}_7 \mid x_7) \quad p(\widetilde{z}_8 \mid x_8) \quad \cdots$$

16 tiles → 8 pairs



- Resample and merge catalogs from adjacent tiles
- Compute weights for merged catalogs, e.g.,

$$w^n_{1:2} \propto \frac{p(\widetilde{z}^n_{1:2})p(x^n_{1:2} \mid \widetilde{z}^n_{1:2})}{p(\widetilde{z}^n_1)p(x^n_1 \mid \widetilde{z}^n_1)\ p(\widetilde{z}^n_2)p(x^n_2 \mid \widetilde{z}^n_2)}$$

$p(\widetilde{z}_{1:16} \mid x_{1:16})$

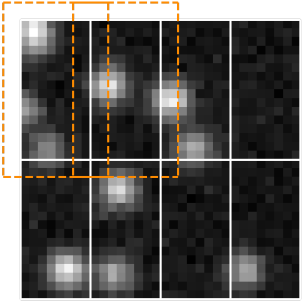
$p(\widetilde{z}_{1:8} \mid x_{1:8})$ $p(\widetilde{z}_{9:16} \mid x_{9:16})$

$p(\widetilde{z}_{1:4} \mid x_{1:4})$ $p(\widetilde{z}_{5:8} \mid x_{5:8})$ $\cdots$ $\cdots$

$p(\widetilde{z}_{1:2} \mid x_{1:2})$ $p(\widetilde{z}_{3:4} \mid x_{3:4})$ $p(\widetilde{z}_{5:6} \mid x_{5:6})$ $p(\widetilde{z}_{7:8} \mid x_{7:8})$ $\cdots$

$p(\widetilde{z}_1 \mid x_1)$ $p(\widetilde{z}_2 \mid x_2)$ $p(\widetilde{z}_3 \mid x_3)$ $p(\widetilde{z}_4 \mid x_4)$ $p(\widetilde{z}_5 \mid x_5)$ $p(\widetilde{z}_6 \mid x_6)$ $p(\widetilde{z}_7 \mid x_7)$ $p(\widetilde{z}_8 \mid x_8)$ $\cdots$

16 tiles → 8 pairs



- Resample and merge catalogs from adjacent tiles
- Compute weights for merged catalogs, e.g.,

$$w^n_{1:2} \propto \frac{p(\widetilde{z}^n_{1:2})p(x^n_{1:2} \mid \widetilde{z}^n_{1:2})}{p(\widetilde{z}^n_1)p(x^n_1 \mid \widetilde{z}^n_1)\;\; p(\widetilde{z}^n_2)p(x^n_2 \mid \widetilde{z}^n_2)}$$

$p(\widetilde{z}_{1:16} \mid x_{1:16})$

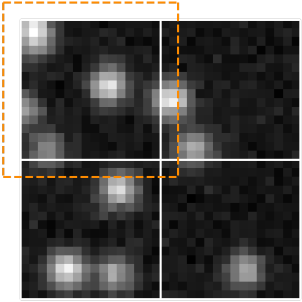
$p(\widetilde{z}_{1:8} \mid x_{1:8})$ $\quad$ $p(\widetilde{z}_{9:16} \mid x_{9:16})$

$p(\widetilde{z}_{1:4} \mid x_{1:4})$ $\quad$ $p(\widetilde{z}_{5:8} \mid x_{5:8})$ $\quad$ $\cdots$ $\quad$ $\cdots$

$p(\widetilde{z}_{1:2} \mid x_{1:2})$ $\quad$ $p(\widetilde{z}_{3:4} \mid x_{3:4})$ $\quad$ $p(\widetilde{z}_{5:6} \mid x_{5:6})$ $\quad$ $p(\widetilde{z}_{7:8} \mid x_{7:8})$ $\quad$ $\cdots$

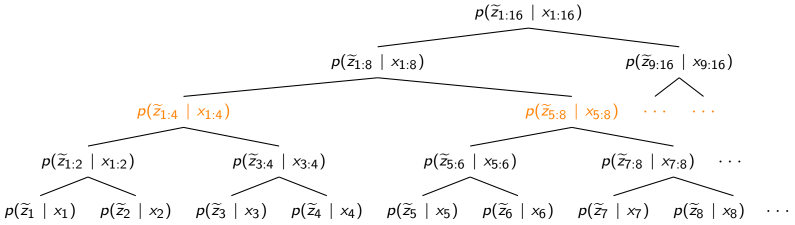
$p(\widetilde{z}_1 \mid x_1)$ $\quad$ $p(\widetilde{z}_2 \mid x_2)$ $\quad$ $p(\widetilde{z}_3 \mid x_3)$ $\quad$ $p(\widetilde{z}_4 \mid x_4)$ $\quad$ $p(\widetilde{z}_5 \mid x_5)$ $\quad$ $p(\widetilde{z}_6 \mid x_6)$ $\quad$ $p(\widetilde{z}_7 \mid x_7)$ $\quad$ $p(\widetilde{z}_8 \mid x_8)$ $\quad$ $\cdots$

8 pairs → 4 quadrants

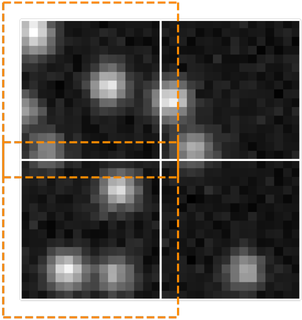


- Resample and merge catalogs from adjacent pairs
- Compute weights for merged catalogs, e.g.,

$$w^n_{1:4} \propto \frac{p(\widetilde{z}^n_{1:4})p(x^n_{1:4} \mid \widetilde{z}^n_{1:4})}{p(\widetilde{z}^n_{1:2})p(x^n_{1:2} \mid \widetilde{z}^n_{1:2}) \;\; p(\widetilde{z}^n_{3:4})p(x^n_{3:4} \mid \widetilde{z}^n_{3:4})}$$



8 pairs $\rightarrow$ 4 quadrants



- Resample and merge catalogs from adjacent pairs
- Compute weights for merged catalogs, e.g.,

$$w^n_{1:4} \propto \frac{p(\widetilde{z}^n_{1:4})p(x^n_{1:4} \mid \widetilde{z}^n_{1:4})}{p(\widetilde{z}^n_{1:2})p(x^n_{1:2} \mid \widetilde{z}^n_{1:2}) \;\; p(\widetilde{z}^n_{3:4})p(x^n_{3:4} \mid \widetilde{z}^n_{3:4})}$$

$p(\widetilde{z}_{1:16} \mid x_{1:16})$

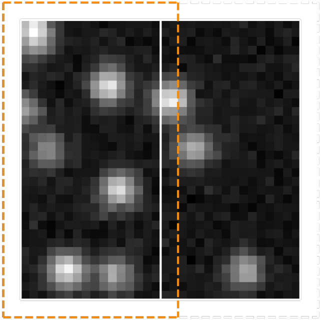
$p(\widetilde{z}_{1:8} \mid x_{1:8})$ $\quad$ $p(\widetilde{z}_{9:16} \mid x_{9:16})$

$p(\widetilde{z}_{1:4} \mid x_{1:4})$ $\quad$ $p(\widetilde{z}_{5:8} \mid x_{5:8})$ $\quad$ $\cdots$ $\quad$ $\cdots$

$p(\widetilde{z}_{1:2} \mid x_{1:2})$ $\quad$ $p(\widetilde{z}_{3:4} \mid x_{3:4})$ $\quad$ $p(\widetilde{z}_{5:6} \mid x_{5:6})$ $\quad$ $p(\widetilde{z}_{7:8} \mid x_{7:8})$ $\quad$ $\cdots$

$p(\widetilde{z}_1 \mid x_1)$ $\quad$ $p(\widetilde{z}_2 \mid x_2)$ $\quad$ $p(\widetilde{z}_3 \mid x_3)$ $\quad$ $p(\widetilde{z}_4 \mid x_4)$ $\quad$ $p(\widetilde{z}_5 \mid x_5)$ $\quad$ $p(\widetilde{z}_6 \mid x_6)$ $\quad$ $p(\widetilde{z}_7 \mid x_7)$ $\quad$ $p(\widetilde{z}_8 \mid x_8)$ $\quad$ $\cdots$

4 quadrants → 2 halves



- Resample and merge catalogs from adjacent quadrants
- Compute weights for merged catalogs, e.g.,

$$w^n_{1:8} \propto \frac{p(\tilde{z}^n_{1:8})p(x^n_{1:8} \mid \tilde{z}^n_{1:8})}{p(\tilde{z}^n_{1:4})p(x^n_{1:4} \mid \tilde{z}^n_{1:4})\;\; p(\tilde{z}^n_{5:8})p(x^n_{5:8} \mid \tilde{z}^n_{5:8})}$$

$p(\tilde{z}_{1:16} \mid x_{1:16})$

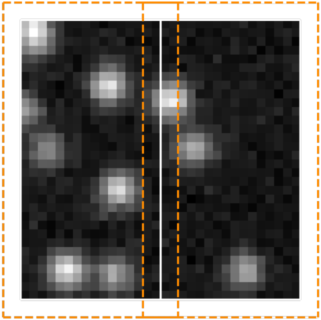
$p(\tilde{z}_{1:8} \mid x_{1:8})$ $\quad$ $p(\tilde{z}_{9:16} \mid x_{9:16})$

$p(\tilde{z}_{1:4} \mid x_{1:4})$ $\quad$ $p(\tilde{z}_{5:8} \mid x_{5:8})$ $\quad \cdots \quad \cdots$

$p(\tilde{z}_{1:2} \mid x_{1:2})$ $\quad$ $p(\tilde{z}_{3:4} \mid x_{3:4})$ $\quad$ $p(\tilde{z}_{5:6} \mid x_{5:6})$ $\quad$ $p(\tilde{z}_{7:8} \mid x_{7:8})$ $\quad \cdots$

$p(\tilde{z}_1 \mid x_1)$ $\quad$ $p(\tilde{z}_2 \mid x_2)$ $\quad$ $p(\tilde{z}_3 \mid x_3)$ $\quad$ $p(\tilde{z}_4 \mid x_4)$ $\quad$ $p(\tilde{z}_5 \mid x_5)$ $\quad$ $p(\tilde{z}_6 \mid x_6)$ $\quad$ $p(\tilde{z}_7 \mid x_7)$ $\quad$ $p(\tilde{z}_8 \mid x_8)$ $\quad \cdots$

4 quadrants → 2 halves



- Resample and merge catalogs from adjacent quadrants
- Compute weights for merged catalogs, e.g.,

$$w^n_{1:8} \propto \frac{p(\tilde{z}^n_{1:8})\, p(x^n_{1:8} \mid \tilde{z}^n_{1:8})}{p(\tilde{z}^n_{1:4})\, p(x^n_{1:4} \mid \tilde{z}^n_{1:4}) \;\; p(\tilde{z}^n_{5:8})\, p(x^n_{5:8} \mid \tilde{z}^n_{5:8})}$$

$p(\tilde{z}_{1:16} \mid x_{1:16})$

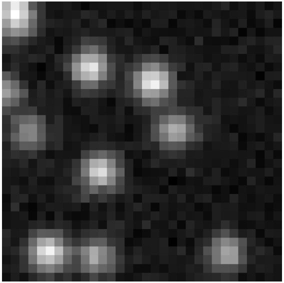
$p(\tilde{z}_{1:8} \mid x_{1:8})$ $\quad$ $p(\tilde{z}_{9:16} \mid x_{9:16})$

$p(\tilde{z}_{1:4} \mid x_{1:4})$ $\quad$ $p(\tilde{z}_{5:8} \mid x_{5:8})$ $\quad \cdots \quad \cdots$

$p(\tilde{z}_{1:2} \mid x_{1:2})$ $\quad$ $p(\tilde{z}_{3:4} \mid x_{3:4})$ $\quad$ $p(\tilde{z}_{5:6} \mid x_{5:6})$ $\quad$ $p(\tilde{z}_{7:8} \mid x_{7:8})$ $\quad \cdots$

$p(\tilde{z}_1 \mid x_1)$ $\quad$ $p(\tilde{z}_2 \mid x_2)$ $\quad$ $p(\tilde{z}_3 \mid x_3)$ $\quad$ $p(\tilde{z}_4 \mid x_4)$ $\quad$ $p(\tilde{z}_5 \mid x_5)$ $\quad$ $p(\tilde{z}_6 \mid x_6)$ $\quad$ $p(\tilde{z}_7 \mid x_7)$ $\quad$ $p(\tilde{z}_8 \mid x_8)$ $\quad \cdots$

2 halves → 1 image



- Resample and merge catalogs from adjacent halves
- Compute weights for merged catalogs, e.g.,

$$w^n_{1:16} \propto \frac{p(\widetilde{z}^n_{1:16})p(x^n_{1:16} \mid \widetilde{z}^n_{1:16})}{p(\widetilde{z}^n_{1:8})p(x^n_{1:8} \mid \widetilde{z}^n_{1:8}) \;\; p(\widetilde{z}^n_{9:16})p(x^n_{9:16} \mid \widetilde{z}^n_{9:16})}$$

$p(\widetilde{z}_{1:16} \mid x_{1:16})$

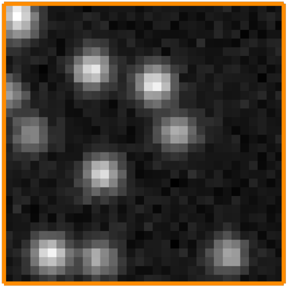
$p(\widetilde{z}_{1:8} \mid x_{1:8})$ $p(\widetilde{z}_{9:16} \mid x_{9:16})$

$p(\widetilde{z}_{1:4} \mid x_{1:4})$ $p(\widetilde{z}_{5:8} \mid x_{5:8})$ $\cdots$ $\cdots$

$p(\widetilde{z}_{1:2} \mid x_{1:2})$ $p(\widetilde{z}_{3:4} \mid x_{3:4})$ $p(\widetilde{z}_{5:6} \mid x_{5:6})$ $p(\widetilde{z}_{7:8} \mid x_{7:8})$ $\cdots$

$p(\widetilde{z}_1 \mid x_1)$ $p(\widetilde{z}_2 \mid x_2)$ $p(\widetilde{z}_3 \mid x_3)$ $p(\widetilde{z}_4 \mid x_4)$ $p(\widetilde{z}_5 \mid x_5)$ $p(\widetilde{z}_6 \mid x_6)$ $p(\widetilde{z}_7 \mid x_7)$ $p(\widetilde{z}_8 \mid x_8)$ $\cdots$

Discard detections in the padded region



$p(z_{1:16} \mid x_{1:16})$

$p(\widetilde{z}_{1:8} \mid x_{1:8})$ $p(\widetilde{z}_{9:16} \mid x_{9:16})$

$p(\widetilde{z}_{1:4} \mid x_{1:4})$ $p(\widetilde{z}_{5:8} \mid x_{5:8})$ $\cdots$ $\cdots$

$p(\widetilde{z}_{1:2} \mid x_{1:2})$ $p(\widetilde{z}_{3:4} \mid x_{3:4})$ $p(\widetilde{z}_{5:6} \mid x_{5:6})$ $p(\widetilde{z}_{7:8} \mid x_{7:8})$ $\cdots$

$p(\widetilde{z}_1 \mid x_1)$ $p(\widetilde{z}_2 \mid x_2)$ $p(\widetilde{z}_3 \mid x_3)$ $p(\widetilde{z}_4 \mid x_4)$ $p(\widetilde{z}_5 \mid x_5)$ $p(\widetilde{z}_6 \mid x_6)$ $p(\widetilde{z}_7 \mid x_7)$ $p(\widetilde{z}_8 \mid x_8)$ $\cdots$

Case study: Crowded starfields (cont.)

- 1,000 synthetic images (32 pixels $\times$ 32 pixels)
  - Up to 12 stars in each image

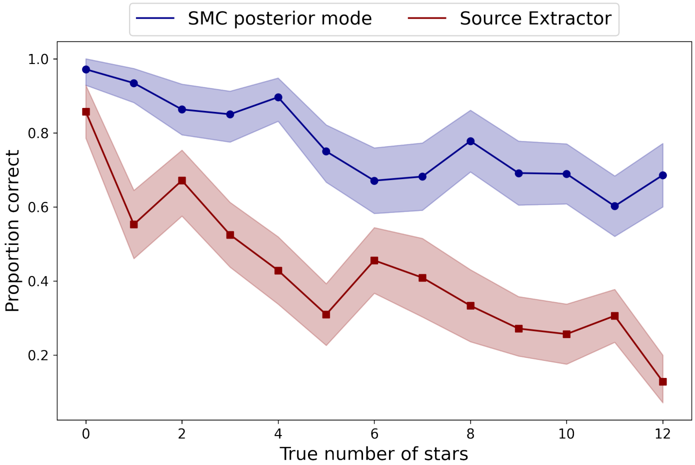
Case study: Crowded starfields (cont.)

- 1,000 synthetic images (32 pixels $\times$ 32 pixels)
  - Up to 12 stars in each image
- Compare **SMC** and **Source Extractor** in terms of estimated number of stars and estimated total flux

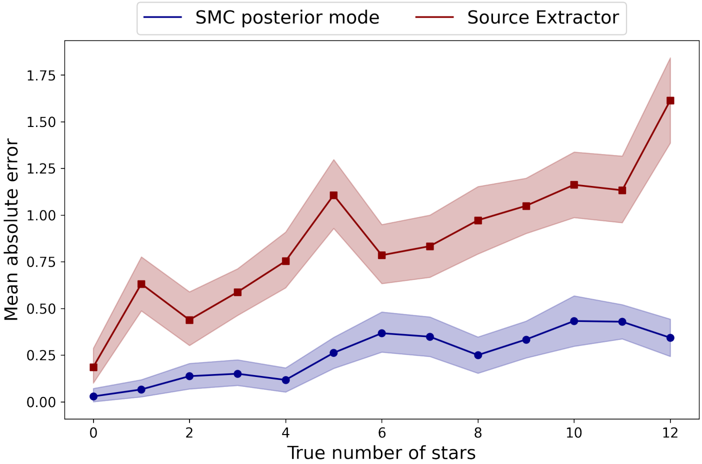
Case study: Crowded starfields (cont.)

- 1,000 synthetic images (32 pixels $\times$ 32 pixels)
  - Up to 12 stars in each image
- Compare **SMC** and **Source Extractor** in terms of estimated number of stars and estimated total flux
- SMC settings:
  - Tiles of size 8 pixels $\times$ 8 pixels, each with 2-pixel-wide padded margin
  - Make up to 5 detections per padded tile
  - 2,500 catalogs for each $s \in \{0, 1, \ldots, 5\}$

Accuracy of estimated number of stars



MAE of estimated number of stars



Accuracy of estimated total flux

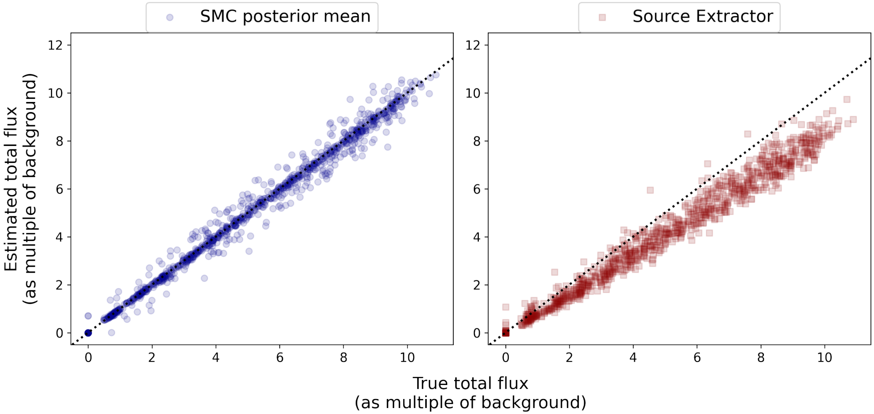


Image reconstructions

Image | One SMC catalog | Source Extractor

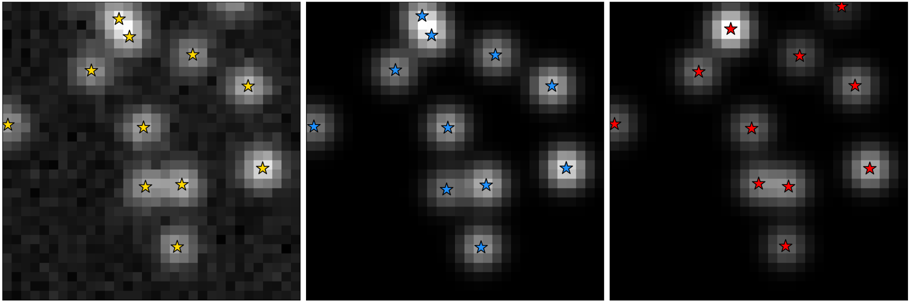


Image reconstructions

Image | One SMC catalog | Source Extractor

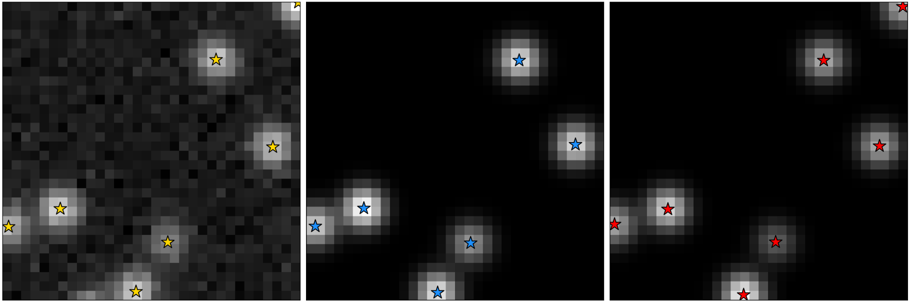
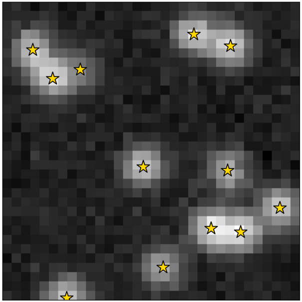
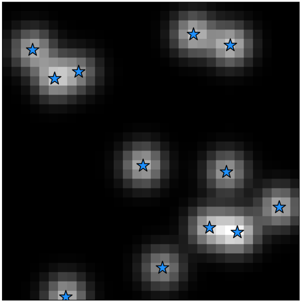


Image reconstructions

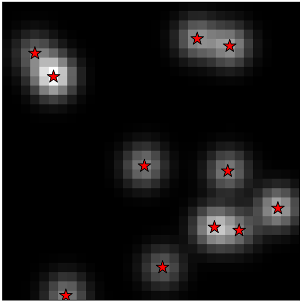
Image



One SMC catalog



Source Extractor



Limitations and future work

Limitations and future work

- **Limitation #1:** Peaked likelihood $\rightarrow$ low effective sample size

Limitations and future work

- **Limitation #1:** Peaked likelihood $\rightarrow$ low effective sample size
  - **Next step:** Incorporate tempering in the divide-and-conquer procedure

Limitations and future work

- **Limitation #1:** Peaked likelihood $\rightarrow$ low effective sample size
  - **Next step:** Incorporate tempering in the divide-and-conquer procedure

- **Limitation #2:** Mutation step of SMC absorbs most of runtime

Limitations and future work

- **Limitation #1:** Peaked likelihood $\rightarrow$ low effective sample size
  - **Next step:** Incorporate tempering in the divide-and-conquer procedure
- **Limitation #2:** Mutation step of SMC absorbs most of runtime
  - **Next step:** Use a gradient-based MCMC kernel or a normalizing flow

Limitations and future work

- **Limitation #1:** Peaked likelihood $\rightarrow$ low effective sample size
  - **Next step:** Incorporate tempering in the divide-and-conquer procedure
- **Limitation #2:** Mutation step of SMC absorbs most of runtime
  - **Next step:** Use a gradient-based MCMC kernel or a normalizing flow
- **Limitation #3:** Requires an accurate parametric object model

Limitations and future work

- **Limitation #1:** Peaked likelihood $\rightarrow$ low effective sample size
  - **Next step:** Incorporate tempering in the divide-and-conquer procedure
- **Limitation #2:** Mutation step of SMC absorbs most of runtime
  - **Next step:** Use a gradient-based MCMC kernel or a normalizing flow
- **Limitation #3:** Requires an accurate parametric object model
  - **Next step:** Investigate sensitivity to model misspecification

Thank you!

https://linktr.ee/timwhite0