Linear Object Detection in Document Images using **Multiple Object Tracking**

P. Bernet, J. Chazalon, E. Carlinet, A. Bourquelot and E. Puybareau — *EPITA Research Lab (LRE), France*

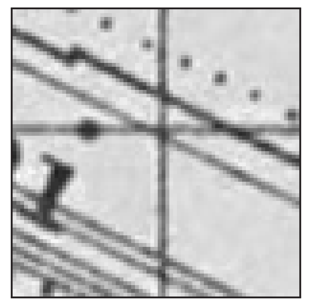
https://github.com/EPITAResearchLab/bernet.23.icdar

DOI 10.5281/zenodo.7871318 DOI 10.5281/zenodo.7927611

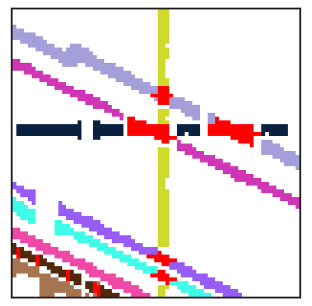
ANR-18-CE38-0013

Open code and data

Goal

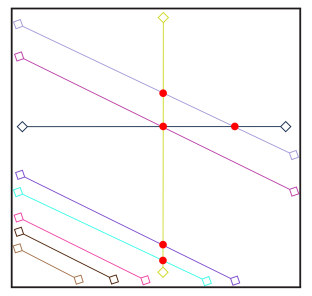


Input:
Document image



Output:
Pixel-accurate instance segmentation

with intersections and missing parts
→ *for object removal...*



Optional output:
Simplified vectorization

start, end, intersection coordinates
→ *for deslanting...*

Limitations of **Existing Linear Object Detectors** Applied to Document Images

Family	Output accuracy	Segm. type	Quality	Training -free	Fast	Open implem.	handles ... objects: *curved*	*rotated*	*erased*	*dashed*	*intersect.*	Comment
Pixel-wise edge classifiers ▷ *U-Net, HED, BDCN, EDTER...*	**pixel-level** classification	semantic		✘	✘	✔	✔	✔	✘	✘	✘	Good preprocessing for us
Hough-transform-based detectors ▷ *HT, RHT, PHT...*	coordinates start-end	**instance**		✔	✔	✔	✘	✔	≈	≈	✘	Avoid on doc. img.
Region growing tracers ▷ *Canny, LSD, EDLine, AG3Line, ELSED...*	coordinates start-end, seq.	**instance**		✔	✔	✔	✔	✔	≈	✘	≈	Tricky to use on doc. img.
Deep linear object detectors ▷ *Faster-RCNN-like: L-CNN, HAWP, F-Clip...*	coordinates start-end	**instance**		✘	✘	partial	✘	✔	≈	≈	✘	Lack of train. data/models
Vertex sequence generators/decoders ▷ *Polygon-RNN, LETR...*	coordinates sequence	**instance**		✘	✘	partial	✔	✔	✔	✔	≈	Lack of train. data/models
(Linear) object trackers ▷ *Kalman filters: one forgotten approach*	**pixel-level** connectivity	**instance**		✔	✔	✘	✔	✔	✔	✔	✔	**We re-implemented it.**

Contributions and **Takeaways**

An **open source** tool which **accurately segments instances of linear objects** in **document images**.

A **Multiple Object Tracking (MOT) framework** which generalizes the original (forgotten) approach of [Poulain d'Andecy et al., ICPR'94].

Revised datasets and metrics to run benchmarks on two sub-tasks:

- ▷ **coarse vectorization** and
- ▷ **pixel-level instance segmentation**

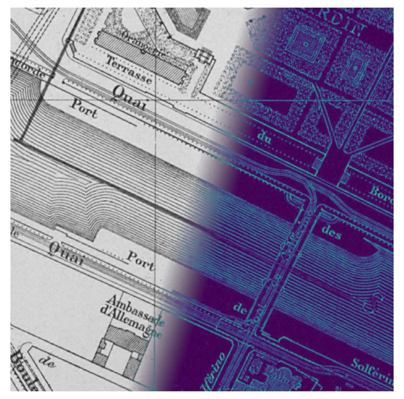
Key findings:

⇒ **Detection using MOT**: **best scores** and almost the **fastest** among **training-free** approaches

⇒ **Viable alternatives to Kalman filters** for pixel-level instance segmentation: EMA, 1€...

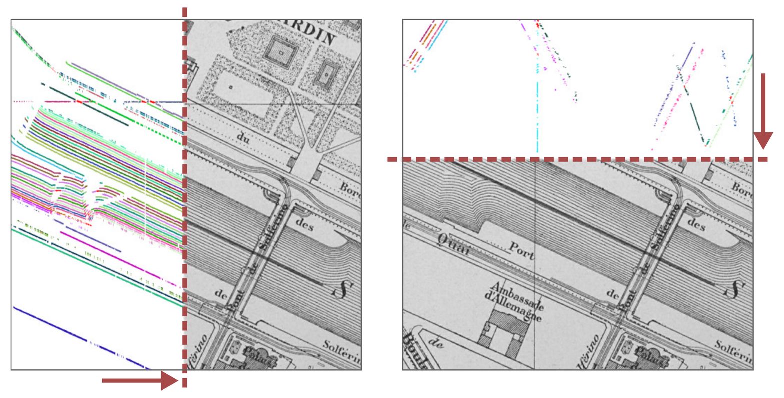
Multiple Object Tracking Framework for pixel-accurate instance segmentation of complex linear objects in document images

Overview



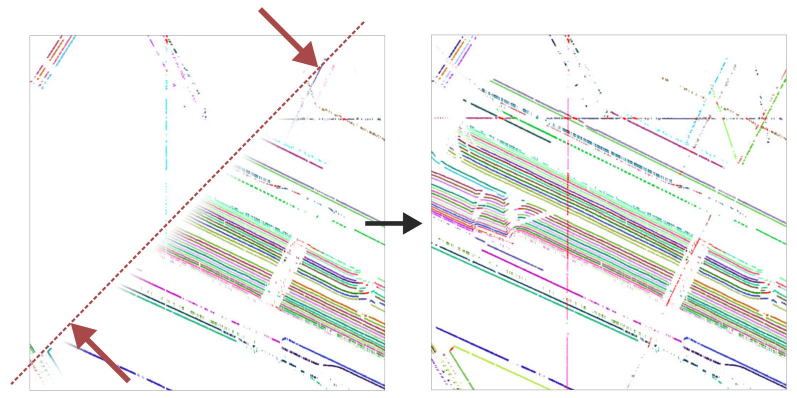
Preprocessing

- ▷ optional stage (gray image is sufficient)
- ▷ any edge semantic segmentation system



Processing

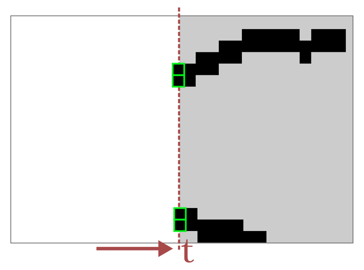
- ▷ 1 horizontal scan of the image, column-wise
- ▷ 1 vertical scan of the image, row-wise



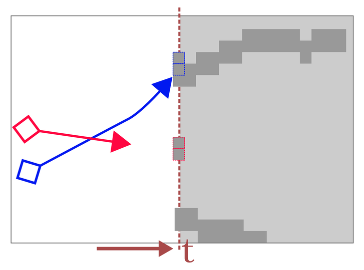
Postprocessing

- ▷ merge object duplicates (close to 45°)
- ▷ filter objects by attribute (length in partic.)

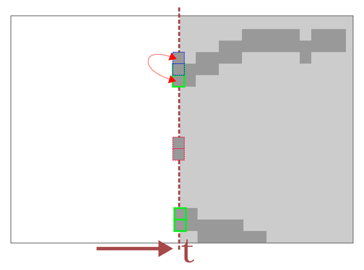
Processing Steps — *repeated at each scene t of each scan*



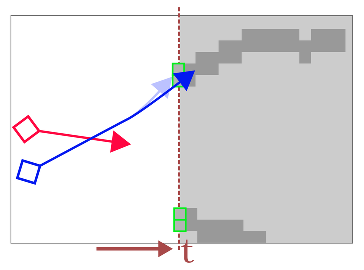
1 **Extract** the actual observations from the current scene at t.



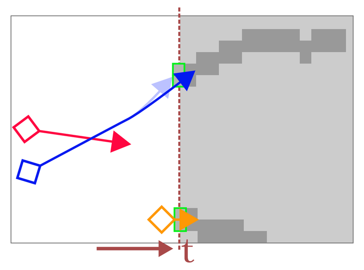
2 **Predict** the position of the next observation (at t) for each tracker using its internal state.



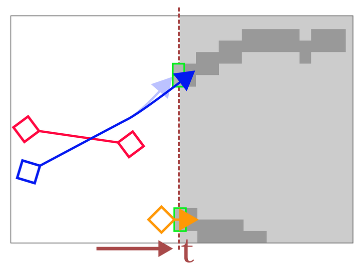
3 **Match** predictions with actual observations based on minimal distance.



4 **Integrate** matched observations into each tracker state or extrapolate if no match.

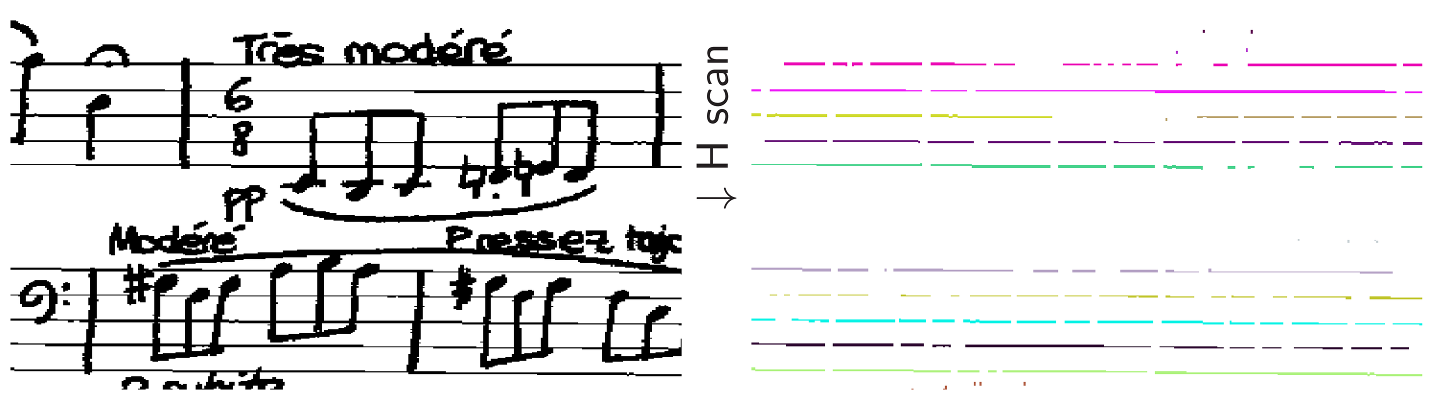


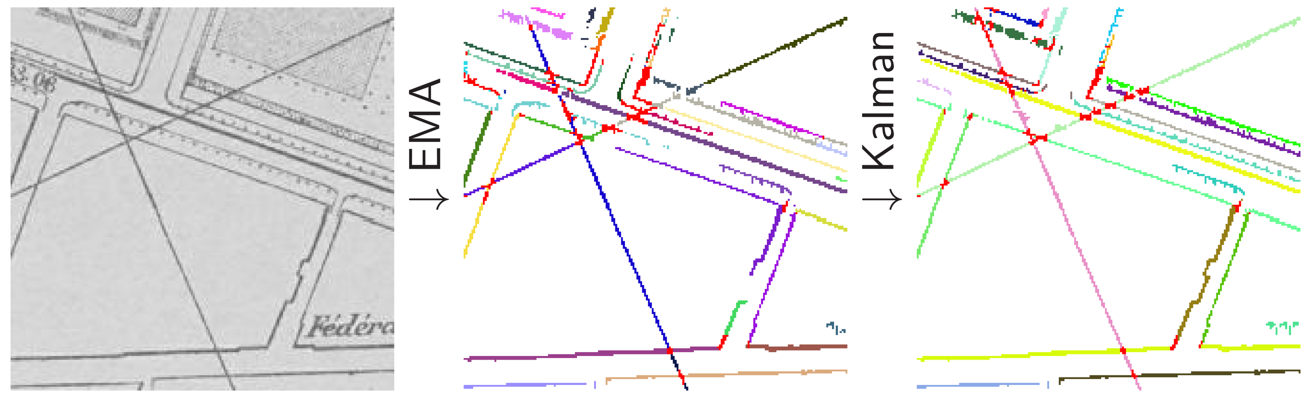
5 **Start** new trackers for unmatched observations.



6 **Stop** trackers lost for too long.

Some **Qualitative** Results





Evaluations and **Results**

Vectorization Task — *comparison against other training-free approaches*

Expected output: start-end coordinates for each target linear object

Dataset new

- ▷ public dataset of historical trade directories
- ▷ train: 5 images, test: 190 images (≈ 4.3 object/image)

Metrics new

- ▷ **F-score** from [Cho *et al.*, TPAMI'18] adapted to ensure precision $\in [0, 1]$
- ▷ **F-score$_2$**: same with fragmentation penalty

avg (± stddev)	Time (ms)	F-score Train	F-score Test	F-score$_2$ Train	F-score$_2$ Test
MOT (Kalman)	633	**95.2 (±7.5)**	**90.1 (±24.0)**	**92.7 (±13.0)**	**87.6 (±24.6)**
AG3Line	434	66.2 (±23.8)	72.5 (±35.4)	25.9 (±9.9)	24.2 (±13.7)
CannyLines	551	81.2 (±22.7)	84.4 (±24.2)	39.0 (±13.5)	34.2 (±14.0)
EDLines	314	83.2 (±23.6)	87.4 (±24.0)	35.5 (±8.3)	30.5 (±12.3)
ELSED	**264**	91.1 (±11.3)	87.0 (±26.6)	45.3 (±9.6)	35.2 (±13.7)
Hough	419	80.5 (±14.5)	64.8 (±30.0)	23.5 (±9.2)	18.2 (±10.1)
LSD	2338	18.7 (±10.3)	12.5 (±8.5)	1.6 (±1.5)	0.5 (±0.6)
LSD II	2206	76.7 (±28.6)	53.3 (±43.7)	47.6 (±24.7)	20.7 (±17.9)

Instance Segmentation Task — *comparison among tracking algorithms*

Expected output: list of pixels contained by each target linear object

Dataset new

- ▷ ICDAR'13 competition on music staff removal with extra instance labeling
- ▷ train: 5 images, test: 1995 images

Metrics

- ▷ **ICDAR'13** competition metric
- ▷ **COCO Panoptic**: modern instance segmentation performance indicator

avg (± stddev)	Time (ms)	Panoptic Quality Train	Panoptic Quality Test	F-Score (ICDAR'13) Train	F-Score (ICDAR'13) Test
Last observation	323	86.3 (± 5.2)	83.7 (± 11.1)	95.9 (± 2.1)	95.4 (± 2.7)
SMA	323	67.6 (± 17.0)	66.0 (± 17.6)	90.7 (± 6.5)	89.9 (± 7.4)
EMA	322	74.0 (± 14.9)	65.5 (± 18.4)	92.5 (± 4.7)	89.6 (± 7.7)
Double exp.	**320**	55.4 (± 16.2)	51.7 (± 15.8)	87.3 (± 5.0)	83.8 (± 8.6)
One euro	327	**87.2 (± 5.9)**	**85.1 (± 9.6)**	**95.9 (± 2.1)**	**95.7 (± 2.2)**
Kalman	328	85.0 (± 7.1)	80.7 (± 15.6)	95.3 (± 2.5)	94.1 (± 5.6)