Fast Approximate Modelling of the Next Combination Result for Stopping the Text Recognition in a Video

Konstantin Bulatov, Nadezhda Fedotova, Vladimir V. Arlazarov

Smart Engines, FRC CSC RAS, MIPT

Mobile text recognition

Mobile DAR systems:

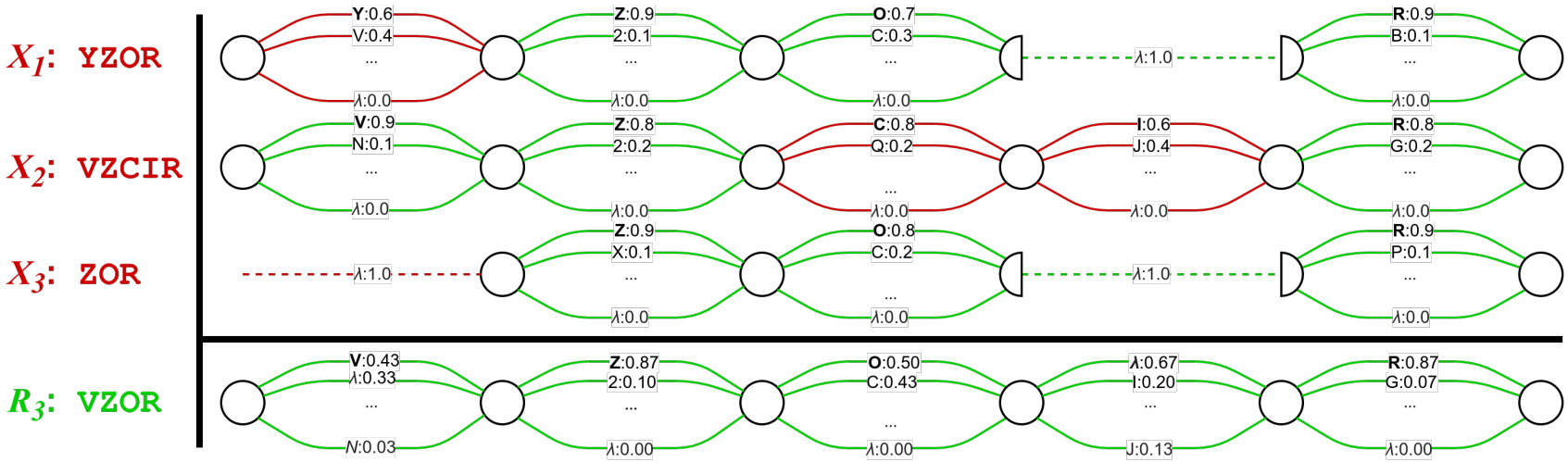
- Mobile document data extraction in real time
- Ability to use video stream to increase recognition quality
- Per-frame text recognition results combination → stopping problem

Goal of the study: construct and evaluate computationally efficient methods for stopping the process of text recognition in a video

Existing methods

Text recognition result: $X = (x_{jk}) \in [0.0, 1.0]^{M \times K}, \quad \forall j \sum_{k=1}^{K} x_{jk} = 1.0$

ROVER per-frame combination procedure:



Stopping decision based on modelling of the next combination result: threshold the estimated distance (generalized Levenshtein) to the next combination result.

Existing methods

Modelling is performed in a straightforward way:

$$\hat{\Delta}_n = \frac{1}{n+1}\left(\delta + \sum_{i=1}^{n} \rho(R_n, R(X_1, \ldots, X_n, X_i))\right)$$

The modelling method was shown to achieve a lower mean error level given the same mean number of processed frames and vice versa – lower mean number of processed frames for reaching the same mean error level.

Downside: high complexity. If M is the maximal length of individual results X_i and S_n is the length of the combined result R_n, the complexity of making a decision at stage n is $O(nS_nK(S_n + M))$.

Proposed approximations

Naive alignment:

During the test combination $R(X_1, \ldots, X_n, X_i)$ we will assume that the rows of X_i will be aligned with the same rows of R_n as on stage i – skipping costly alignment.

The length of $R(X_1, \ldots, X_n, X_i)$ stays the same as the length of R_n.

Naive Levenshtein:

The alignment between R_n and $R(X_1, \ldots, X_n, X_i)$ is direct – each j-th row of R_n is aligned with the j-th row of $R(X_1, \ldots, X_n, X_i)$, thus the generalized Levenshtein distance between them is a sum of distances in terms of the row metric – skipping costly distance computation.

Proposed methods

Accumulators: $Y_n = (y_{ijk}) \in [0.0, 1.0]^{n \times S_n \times (K+1)}$; $A_{jk} = \sum_{i=1}^{n} y_{ijk}$.

y_{ijk} is a membership value of class k from a row of input X_i which was aligned and merged into the j-th row of the current combined result R_n.

Method A:

$$\hat{\Delta}_n \approx \frac{1}{n+1}\left(\delta + \sum_{i=1}^{n}\sum_{j=1}^{S_n}\sum_{k=0}^{K}\Delta_{ijk}\right),$$

$$\Delta_{ijk} = \frac{|A_{jk} - n \cdot y_{ijk}|}{2n(n+1)},$$

Complexity: $O(nS_nK)$.

Method B:

$$L_{jk} \subset \{1, \ldots, n\} : \quad \forall i \in L_{jk} : n \cdot y_{ijk} < A_{jk},$$

$$B_{jk} = \sum_{i \in L_{jk}} y_{ijk},$$

$$\sum_{i=1}^{n} \Delta_{ijk} = \frac{A_{jk}|L_{jk}| - nB_{jk}}{n(n+1)},$$

Complexity: $O(S_nK \log n)$ using treaps.

Experimental evaluation

Datasets

ID documents

Datasets: MIDV-500 and MIDV-2019

Field groups: document numbers, numerical dates, names in Latin alphabet, and MRZ

All clips normalized to the length of 30 frames

2239 evaluated clips from MIDV-500 and 992 clips from MIDV-2019

Recognition using Smart IDReader.

Arbitrary text

Datasets: ICDAR 2015 Train and YouTube Video Text

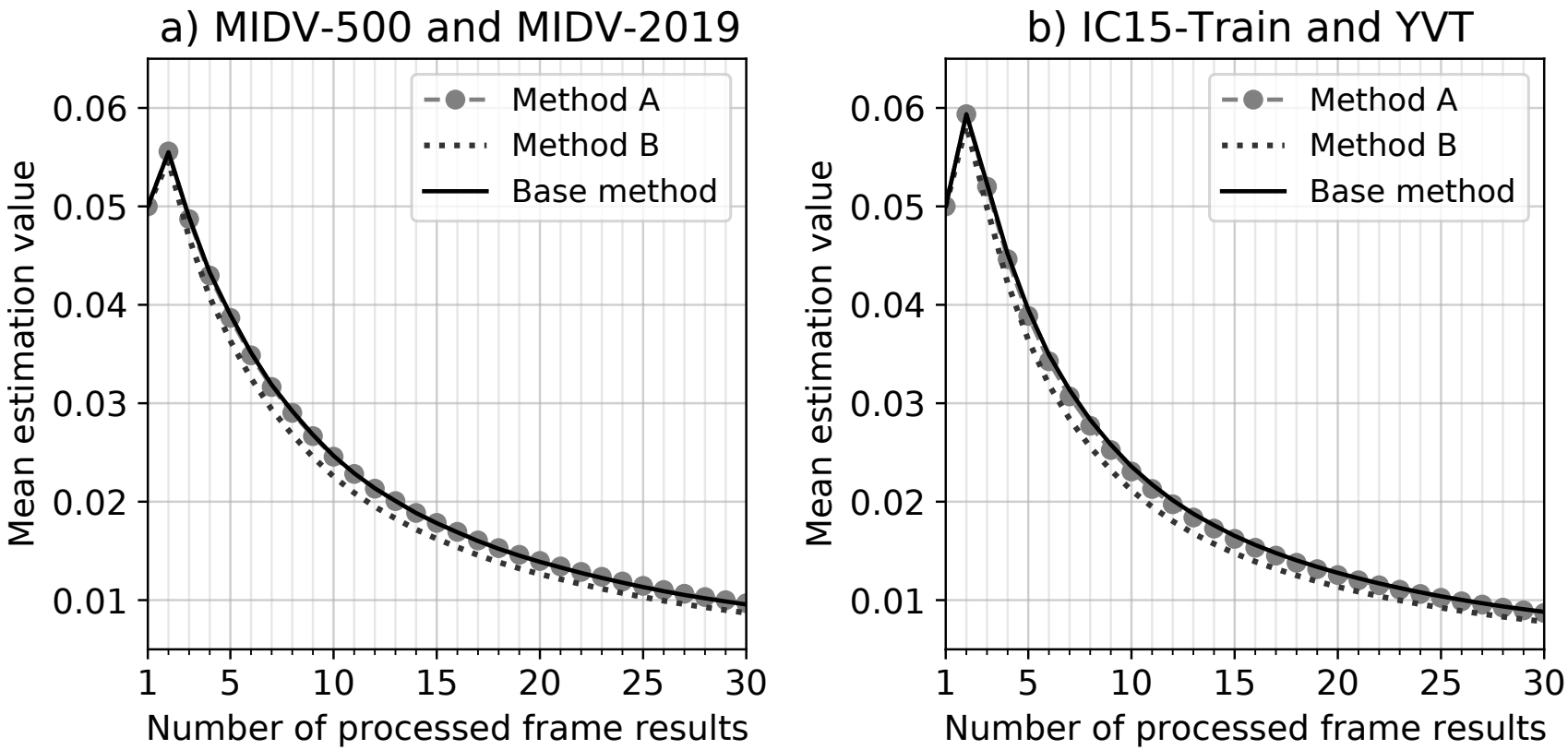
Text objects with alphanumeric characters which are present on at least 30 frames

851 evaluated clips from IC15-Train and 409 clips from YVT

Recognition: published model using thin-plate spline transform, ResNet features, BiLSTM seq. modelling and attention-based prediction (ICCV 2019, J. Baek et al).

Experimental evaluation

Validating the approximations

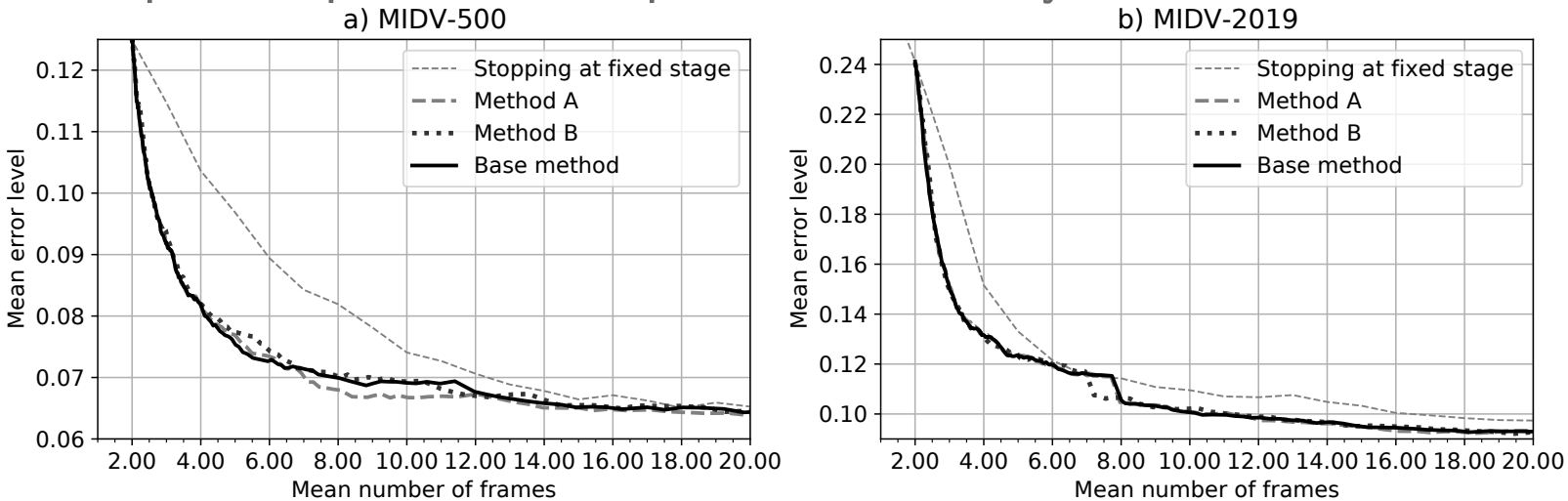


Mean estimation value $\hat{\Delta}_n$ computed using the three evaluated stopping methods on each stage.

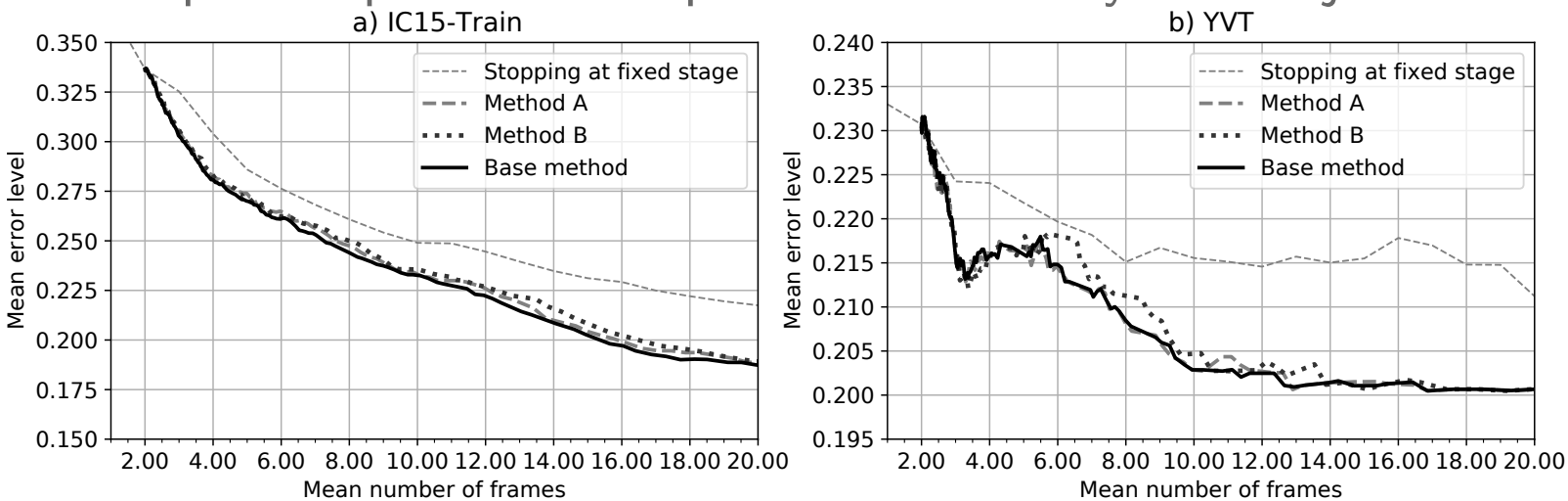
Experimental evaluation

Expected performance profiles

Expected performance profiles – identity document fields.



Expected performance profiles – arbitrary text objects.



Experimental evaluation

Timing (Python 3.7.4 prototype implementation, AMD Ryzen 9 3950X)

Time to combine and make stopping decision – identity document fields.

Method	Time on stage n (in seconds)				
	$n = 5$	$n = 10$	$n = 15$	$n = 20$	$n = 25$
Base	0.228	0.448	0.678	0.906	1.143
Method A	**0.022**	**0.022**	**0.023**	**0.024**	**0.024**
Method B	0.027	0.030	0.032	0.033	0.034

Time to combine and make stopping decision – arbitrary text objects.

Method	Time on stage n (in seconds)				
	$n = 5$	$n = 10$	$n = 15$	$n = 20$	$n = 25$
Base	0.024	0.050	0.078	0.107	0.136
Method A	**0.002**	**0.003**	**0.003**	**0.003**	**0.003**
Method B	0.004	0.004	0.005	0.005	0.005

Conclusion

- We proposed two approximate modelling schemes for text recognition in a video stream which allow to compute the estimated distance to the next combination result and make a stopping decision;
- Both proposed methods were evaluated on open datasets;
- It was shown that the assumptions and approximations had almost no effect on the performance of the stopping method in terms of the achieved mean error level and mean number of consumed observations;
- At the same time, the proposed computational schemes have significantly higher computational performance than the baseline method.

Fast Approximate Modelling of the Next Combination Result for Stopping the Text Recognition in a Video

Konstantin Bulatov, Nadezhda Fedotova, Vladimir V. Arlazarov

Smart Engines, FRC CSC RAS, MIPT

E-mail: kbulatov@smartengines.com