Superpixel Segmentation: An Evaluation

David Stutz
david.stutz@rwth-aachen.de

Introduction

Superpixel algorithms have become a standard tool in computer vision. However, varying experimental setups and metrics used for evaluation make direct comparison difficult. We address this shortcoming with a thorough comparison of thirteen state-of-the-art superpixel algorithms including algorithms utilizing depth information.

Superpixel Algorithms

Superpixel algorithms			
NC	Ren *et al*.	2003	C/MatLab
FH	Felzenswalb *et al*.	2004	C++
QS	Vedaldi *et al*.	2008	C/MatLab
TP	Levinshtein *et al*.	2009	C/MatLab
SLIC	Achanta *et al*.	2010	C++
CIS	Veksler *et al*.	2010	C++
ERS	Liu *et al*.	2011	C++
PB	Zhang *et al*.	2011	C++
CRS	Mester *et al*.	2011	C++
SEEDS	Van den Bergh *et al*.	2012	C++
reSEEDS	**SEEDS** reimplementation	–	C++
TPS	Tang *et al*.	2012	C/MatLab
Superpixel algorithms using depth information			
reSEEDS3D	**reSEEDS** using depth	–	C++
DASP	Weikersdorfer *et al*.	2012	C++
VCCS	Papon *et al*.	2013	C++

Datasets & Benchmark

BSDS500 (Arbeláez *et al*., 2011)**.** 100 training and 200 test images of size 481 $\times$ 321.

NYUV2 (Silberman *et al*., 2012)**.** 200 training and 400 test images of size 608 $\times$ 448 including pre-processed depth.

Benchmark. N set of pixels; $G = \{G_i \subseteq N\}$ ground truth segmentation; $S = \{S_j \subseteq N\}$ superpixel segmentation; $TP(G,S)$, $FN(G,S)$ true positive and false negative boundary pixels.

Boundary Recall:

$$Rec(G,S) = \frac{|TP(G,S)|}{|FN(G,S)| + |TP(G,S)|}.$$

Undersegmentation Error:

$$UE(G,S) = \frac{1}{|N|} \sum_{G_i \in G} \sum_{S_j \cap G_i \neq \emptyset} \min\{|S_j \cap G_i|, |S_j - G_i|\}.$$

Qualitative Results

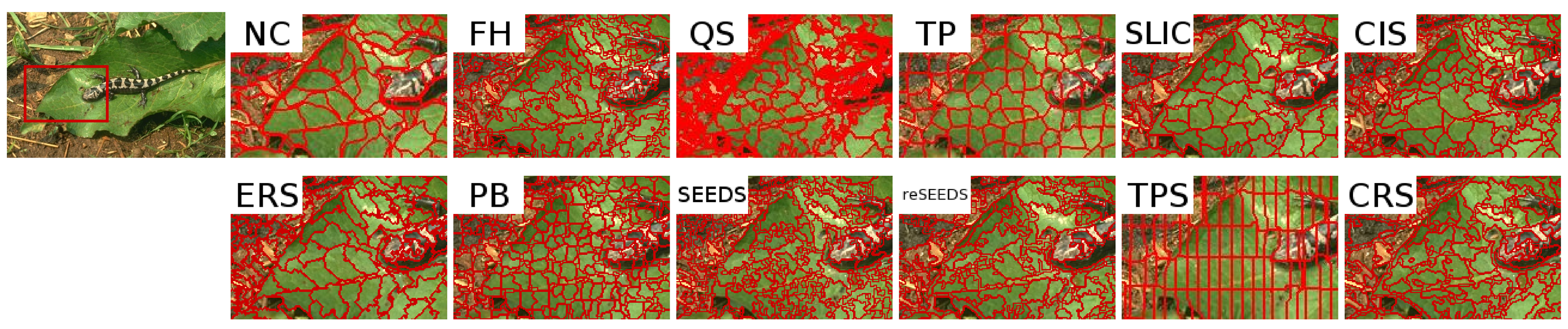

Figure 1: Superpixel segmentations obtained on an image from the BSDS500.

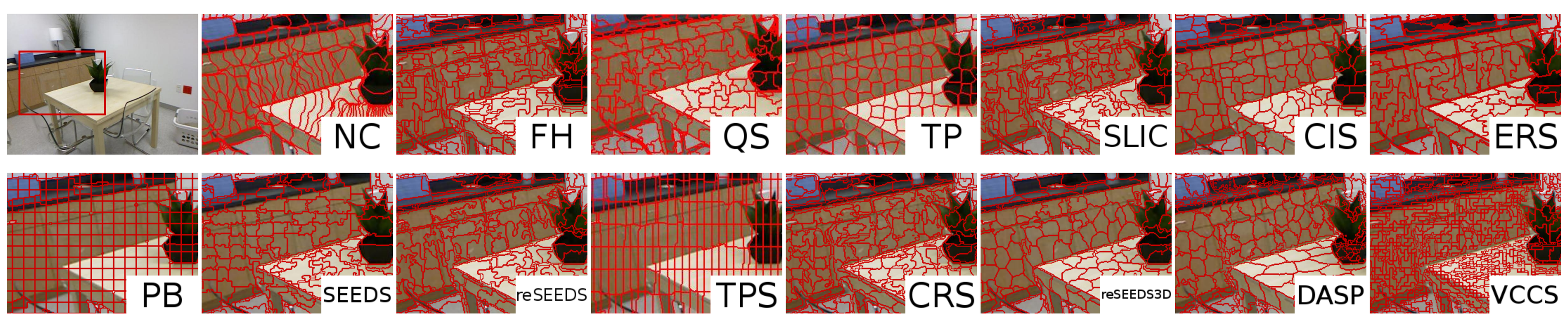

Figure 2: Superpixel segmentations obtained on an image from the NYUV2.

Qualitative Results (cont'd)

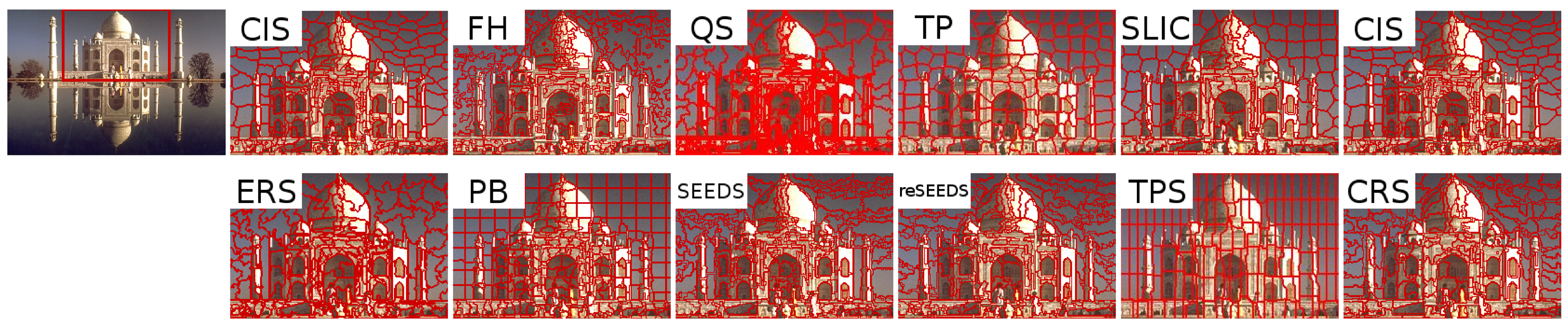

Figure 3: Superpixel segmentations obtained on an image from the BSDS500.

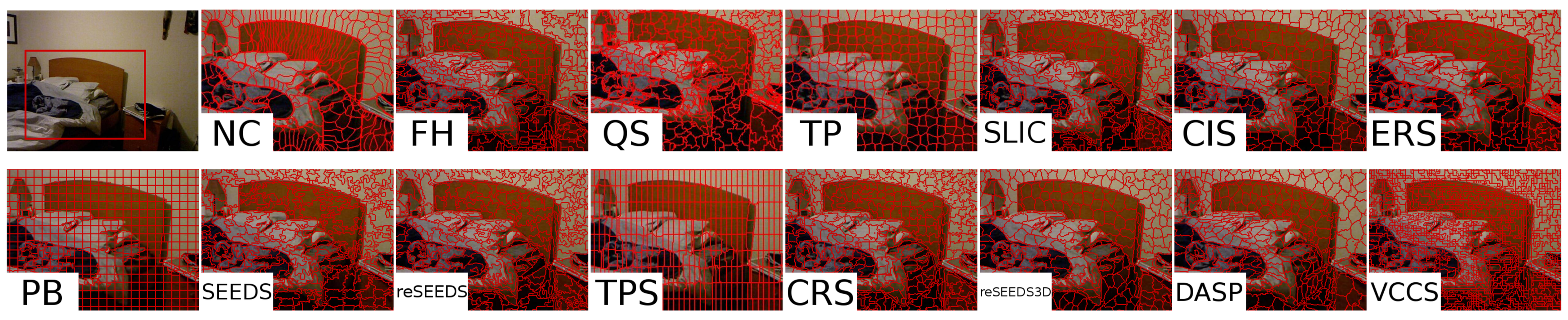

Figure 4: Superpixel segmentations obtained on an image from the NYUV2.

Quantitative Results

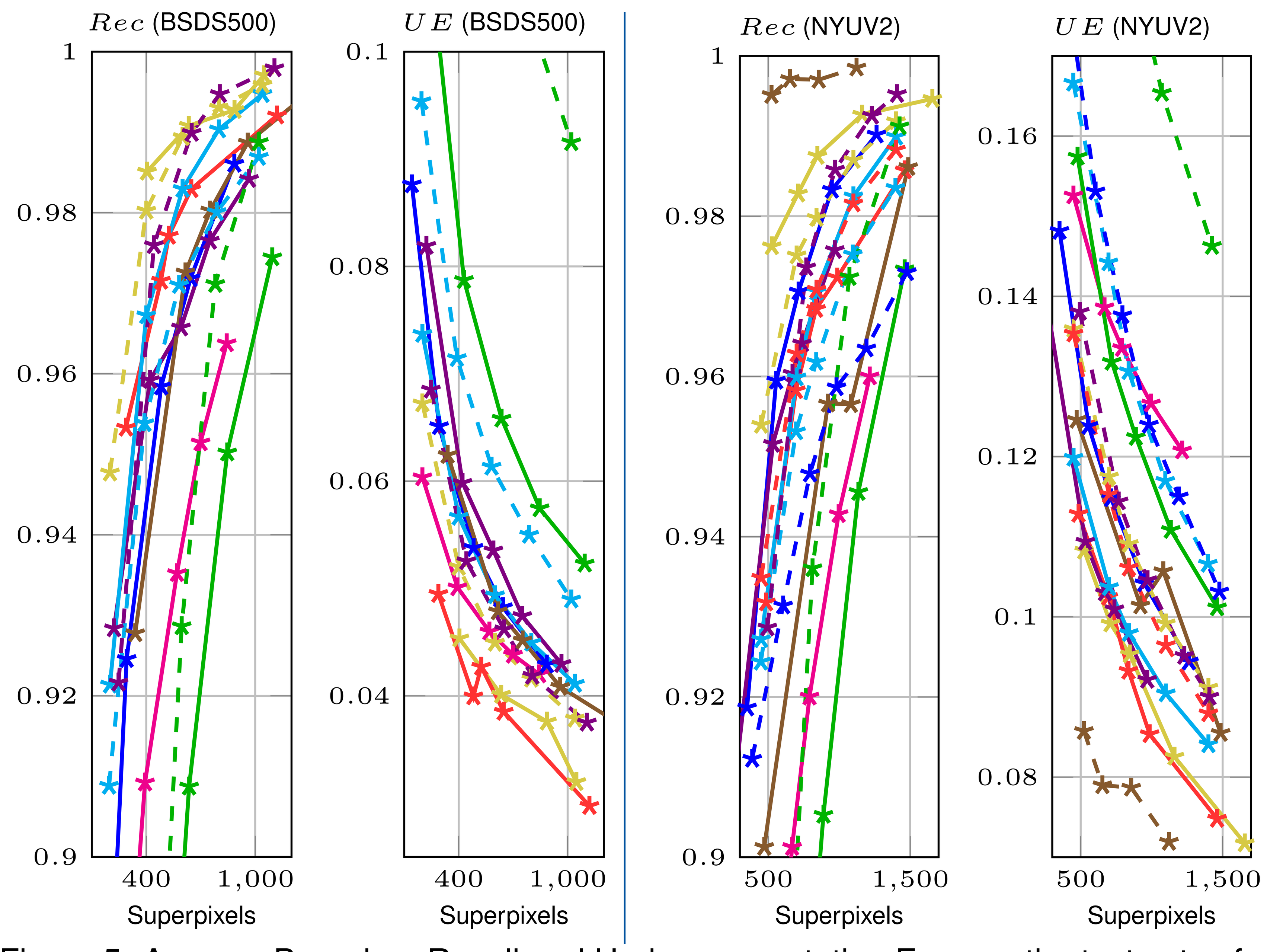

Figure 5: Average Boundary Recall and Undersegmentation Error on the test sets of the BSDS500 (left) and the NYUV2 (right); parameters have been optimized on the corresponding training sets using discrete grid search.

Runtime

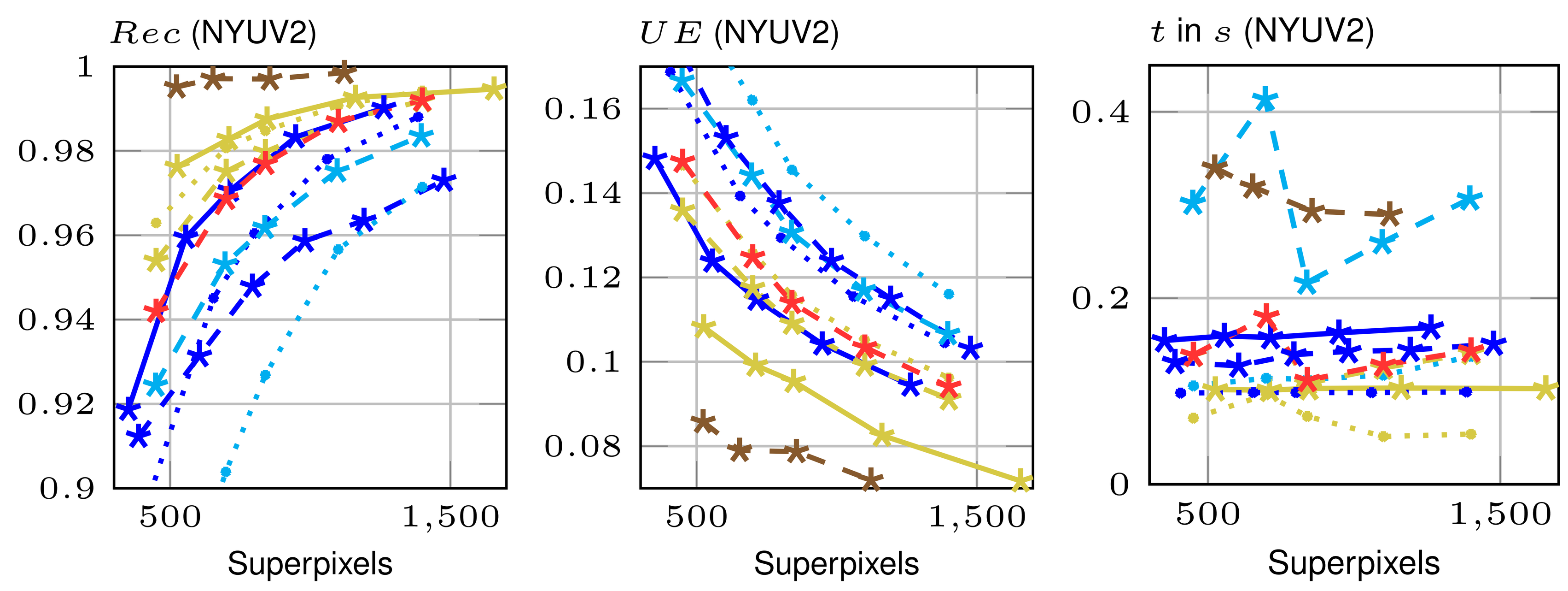

Figure 6: Average Boundary Recall, Undersegmentation Error and runtime on the test set of the NYUV2; * marks algorithms optimized for low runtime: **SLIC** *; **SEEDS** *; **reSEEDS** *.

Conclusion

In conclusion, several algorithms demonstrate excellent performance and low runtime. Furthermore, using depth information does not improve performance in all cases (e.g. **DASP**, **reSEEDS3D**). However, directly segmenting point clouds turns out to be advantageous (e.g. **VCCS**). While additional aspects may be necessary to asses superpixel algorithms with regard to specific applications, the presented results offer a general overview and allow to select suitable superpixel algorithms for a wide range of applications.

Further results and source code: `davidstutz.de`

Acknowledgments:
Alexander Hermans, Prof. Dr. Bastian Leibe