Automatic Simultaneous Segmentation and Fast Registration of Histological Images (ASSAR)

Jan Kybic, Jiří Borovec
Department of Cybernetics, Faculty of Electrical Engineering, Czech Technical University in Prague, Czech Republic
http://cmp.felk.cvut.cz/~kybic

CENTER FOR MACHINE PERCEPTION

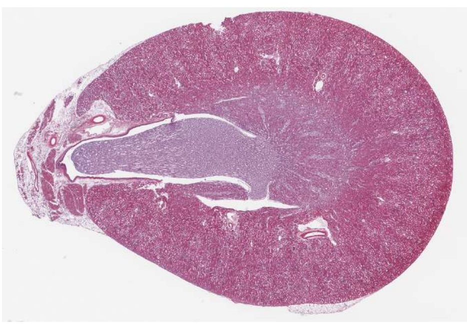
Abstract

We describe an automatic method for fast registration of images with very different appearances. The images are jointly segmented into a small number of classes, the segmented images are registered, and the process is repeated. The segmentation calculates feature vectors on superpixels and then it finds a softmax classifier maximizing mutual information between class labels in the two images. For speed, the registration considers a sparse set of rectangular neighborhoods on the interfaces between classes. A triangulation is created with spatial regularization handled by pairwise spring-like terms on the edges. The optimal transformation is found globally using loopy belief propagation. Multiresolution helps to improve speed and robustness. Our main application is registering stained histological slices, which are large and differ both in the local and global appearance. We show that our method has comparable accuracy to standard pixel-based registration, while being faster and more general.

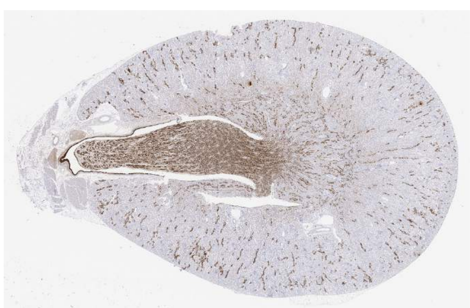
Motivation & Challenges

- Different **large scale** appearance (color and texture)

Rat kidney



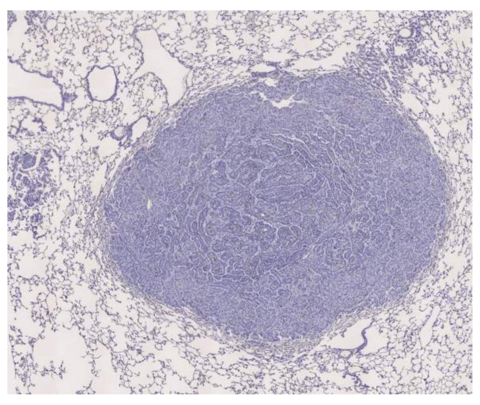
Hematoxylin & Eosin



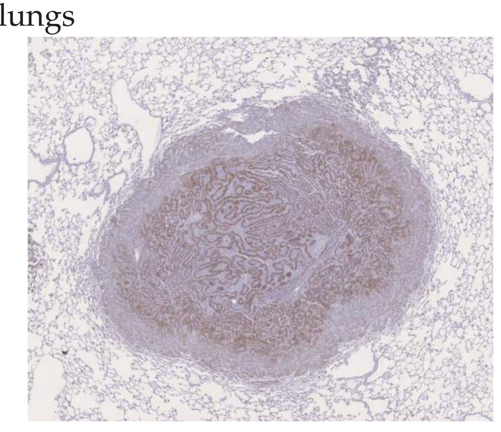
Pan Cytokeratin

- Different **small scale** appearance (details)

Rat lungs



CD31 protein staining



Pro-surfactant protein-C staining

- Few reliable landmarks, repeated structures.
- Very large images (10^8 pixels, $\sim 10\,\mathrm{GB}$)

Method Overview

Key idea

- Each point has a **class** (tissue type).
- Given a class, appearances are conditionally independent.

Iterate until convergence

1. Jointly segment both images
2. Register both segmentations

} optimizing common **MIL criterion**

Mutual Information on Labels (MIL)

- Compare **segmentation labels** f,g of two images, F and G.
- Window $\Omega(\boldsymbol{x}_i)$ of size $h \times h$ pixels around $\boldsymbol{x}_i$
- Co-occurrence probabilities $p_{m,n}$ of class m in F and n in G

$$\mathrm{MIL}(f,g) = \sum_{m,n} p_{m,n} \log \frac{p_{m,n}}{p_m p_n}$$

- **Local MIL:** $p_{m,n}$ inside window $\Omega(\boldsymbol{r}_i)$; $\bar{p}_{m,n}$ over the whole image

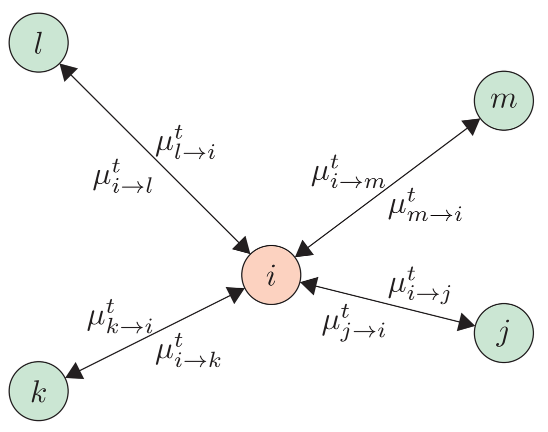
$$\mathrm{MIL}(f,g) \approx \text{const} + \sum_i D_i \qquad D_i = -\sum_{m=1}^{L_F} \sum_{n=1}^{L_G} p_{m,n} \log \frac{\bar{p}_{m,n}}{\bar{p}_m \bar{p}_n}$$

- **Advantages:** Fast and robust. Works for very **different appearances.**

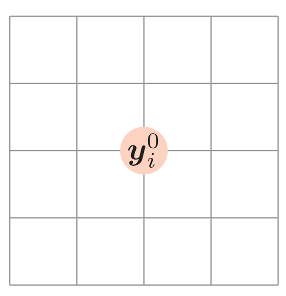
Belief Propagation

- Only small integer displacements $\boldsymbol{y}_i \in \mathbb{Z}^2$, $\|\boldsymbol{y}_i\|_\infty \le d$
- Each node i sends a **message** to all neighbors, $(i,j) \in E$.

$$\mu^t_{i\to j}(\boldsymbol{y}_j) = \min_{\boldsymbol{y}_i} \Big(\omega_{ij} \left\|\boldsymbol{y}_i - \boldsymbol{y}_j\right\|^2 + D_i(\boldsymbol{y}_i) + \sum_{s \ne j} \mu^{t-1}_{s\to i}(\boldsymbol{y}_i) \Big)$$



Interaction graph (part of)



Message $\mu^t_{i\to\cdot}$, domain Υ_i

Experimental Comparison

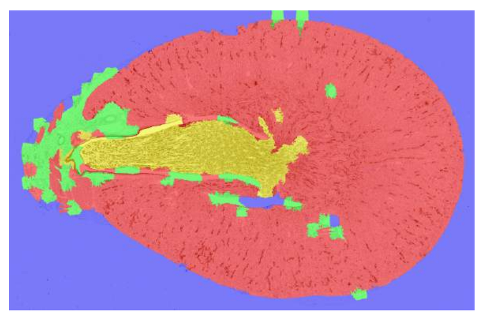
method		time	mean err.	median err.	succ. rate
bUnwarpJ	*B-splines, SSD*	401	64	50	50%
Elastix	*B-splines, MI*	515	54	45	67%
OpenCV+Elastix	*+SURF,RANSAC*	764	**23**	**12**	88%
ASSAR	*MIL,BP*	**130**	36	17	**91%**

$\approx 2000 \times 2000$ pixels. Images on the right. Ground truth — manual landmarks.
ASSAR in Python+Cython, bUnwarpJ in Java, OpenCV+Elastix in C++.
1 ASSAR iteration $\approx 30\,\mathrm{s}$ — often acceptable.

Segmentation

$$f = \Psi_{\boldsymbol{a}_F} F, \quad g = \Psi_{\boldsymbol{a}_G} G \qquad \text{maximize} \quad \mathrm{MIL}(f, T \circ g)$$

- SLIC superpixels S_i
- Superpixel descriptors $\boldsymbol{x}_i = [1 \ \tilde{\boldsymbol{x}}_i]$ (e.g. mean color, texture)
- Optimize $\boldsymbol{a}_{FG} = [\boldsymbol{a}_F \ \boldsymbol{a}_G]$ by BFGS



Softmax

Minimize

$$J_S = -\mathrm{MIL}(f, T \circ g) + \frac{\beta}{2L} \left\|\mathbf{a}_{FG}\right\|^2$$

Probability of S_i belonging to class k

$$z_{i,k} = \frac{\exp(\boldsymbol{a}_k^T \boldsymbol{x}_i)}{\sum_{l=1}^{L} \exp(\boldsymbol{a}_l^T \boldsymbol{x}_i)}$$

Initialization

- Choose L random locations U_k
- Assume $f(U_k^F) = g(U_k^G) = k$
- Find ML estimate of $\boldsymbol{a}_{FG}$ (softmax cost)
- Continue minimizing J_S.
- Repeat N_{init} times.

Registration

- Minimize

$$J(T) = -\mathrm{MIL}(f, T \circ g) + \mathcal{R}(T)$$

- Sparse **control points** $\boldsymbol{r}_i$ on class boundaries

$$\boldsymbol{y}_i = T(\boldsymbol{r}_i)$$

- **Local MIL criterion**

$$J(T) = \sum_i D_i(\boldsymbol{y}_i) + \mathcal{R}$$

- Optimize $\{\boldsymbol{y}_i\}$ by BP
- Interpolate T everywhere
- Multiresolution

Regularization

- Triangulate control points + additional points (to get "nice" triangles)
- Pairwise differences in relative displacements approximate thin membrane

$$\mathcal{R} = \frac{1}{2} \sum_{(i,j)\in E} \omega_{ij} \left\| (\boldsymbol{y}_i - \boldsymbol{r}_i) - (\boldsymbol{y}_j - \boldsymbol{r}_j) \right\|^2$$

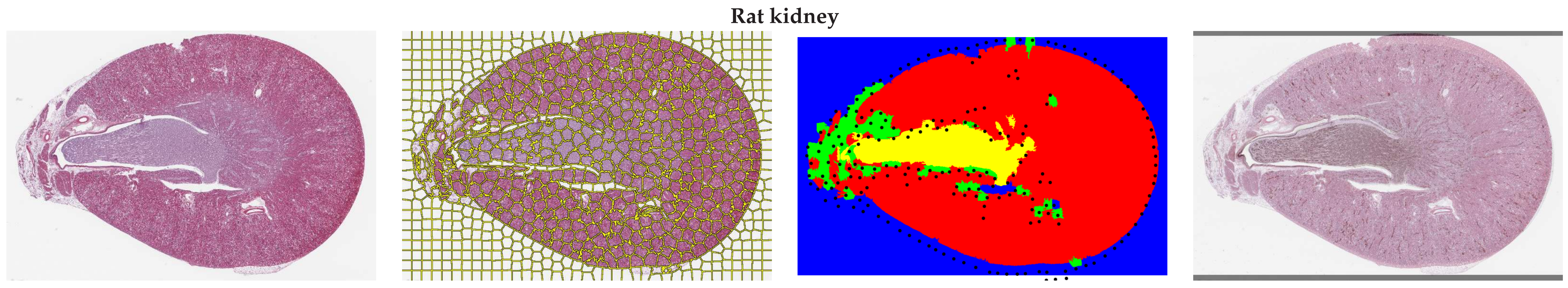
$$\text{with} \quad \omega_{ij} = \sum_{T \in \mathcal{N}(i,j)} \lambda \frac{\|\boldsymbol{r}_i - \boldsymbol{r}_j\|^2}{8A} \left(3 \cot^2 \alpha + \frac{1}{2}\right)$$

Affine initialization

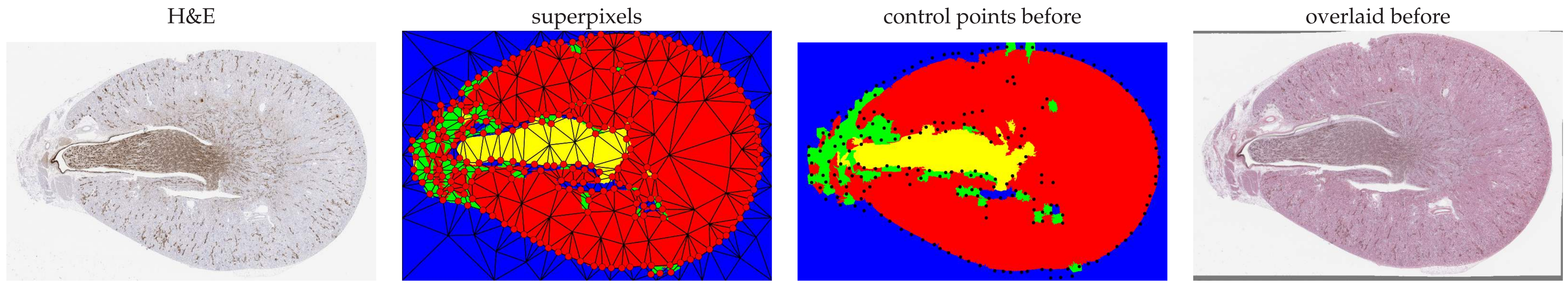
- To increase robustness and speed
- Find locally optimal displacements $\boldsymbol{y}_i^* = \arg\min_{\boldsymbol{y}} D_i(\boldsymbol{y})$
- Fit affine transformation T_A to $\boldsymbol{y}_i$

Example Images

Rat kidney

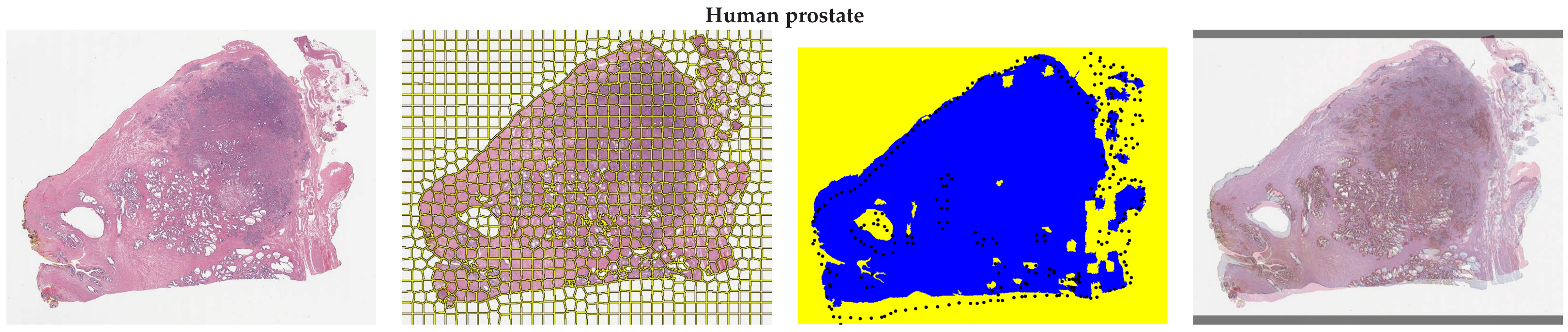


H&E | superpixels | control points before | overlaid before

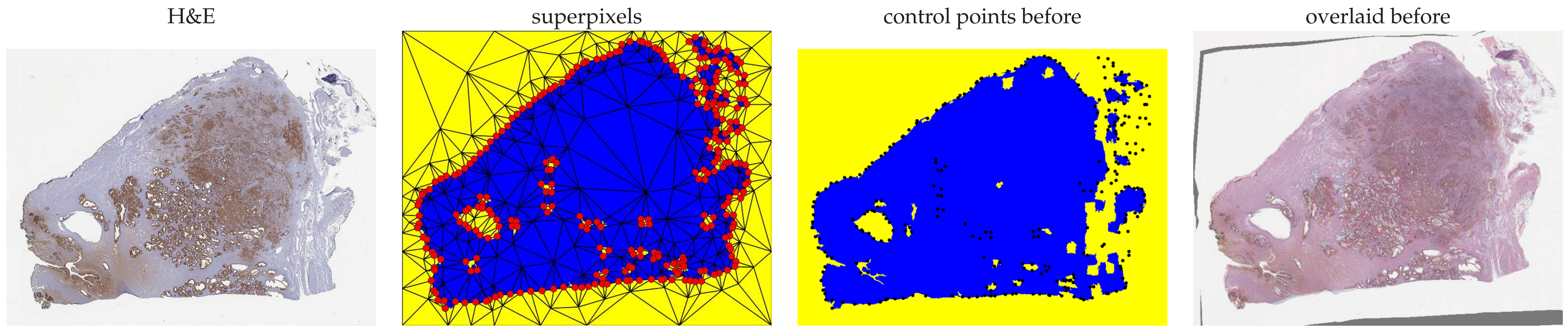


Pan Cytokeratin | segmentation & mesh | control points, 1st iteration | overlaid after

Human prostate



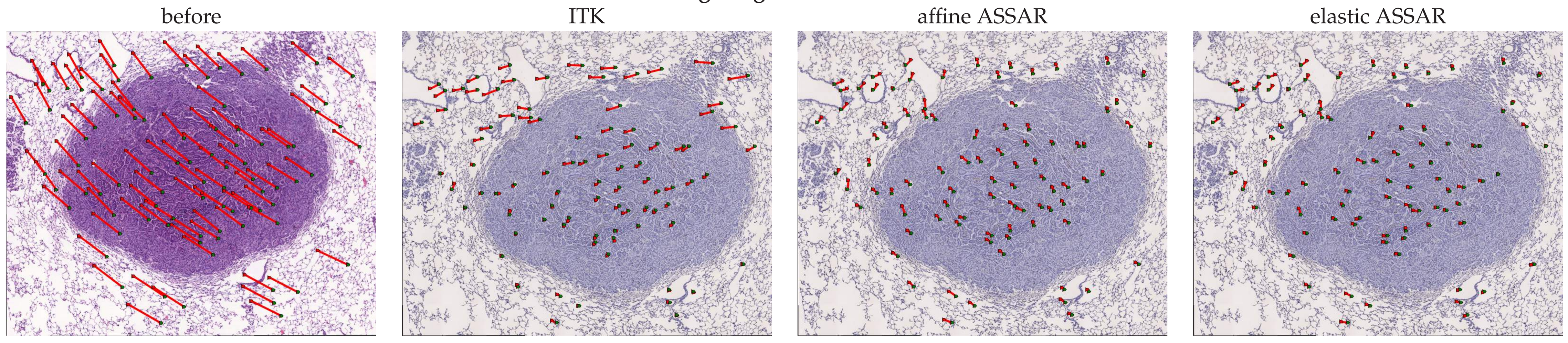
H&E | superpixels | control points before | overlaid before



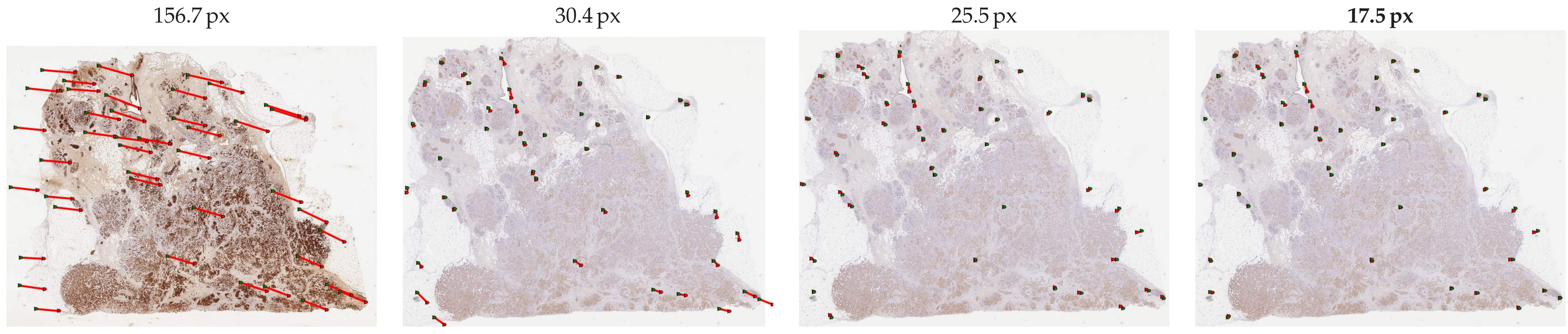
Human prostate, PSAP | segmentation & mesh | control points, 1st iteration | overlaid after

Target registration error

before | ITK | affine ASSAR | elastic ASSAR



156.7 px | 30.4 px | 25.5 px | **17.5 px**



145.3 px | 18.9 px | 10.3 px | **7.7 px**