# KTC2023 Proposal 1 - Post-processing smoothness prior reconstruction using CNNs

R. G. Beraldo, L. A. Ferreira, F. S. Moura, A. K Takahata, R. Suyama

Center for Engineering, Modeling and Applied Social Sciences
Federal University of ABC (Brazil)

October 31, 2023

# Github repository

These slides are related to the code available at
https://github.com/robert-abc/KTC2023-ABC/
in order to join the Kuopio Tomography Challenge 2023
https://www.fips.fi/KTC2023.php

Context: The KTC example code algorithm is based in the smoothness prior.

Available at:
https://www.fips.fi/KTCdata.php
https://zenodo.org/records/8252370

# KTC 2023 - Example code (Smoothness prior)

**Forward model**

- z $= 1e - 6$; %contact impedances (for all electrodes)
- sigma0 $= 1$; %linearization point (for all mesh nodes)

**Smoothness prior**

- corrlength $= 0.115$;
- var_sigma $= 0.05^2$;
- mean_sigma $= \sigma_0$;

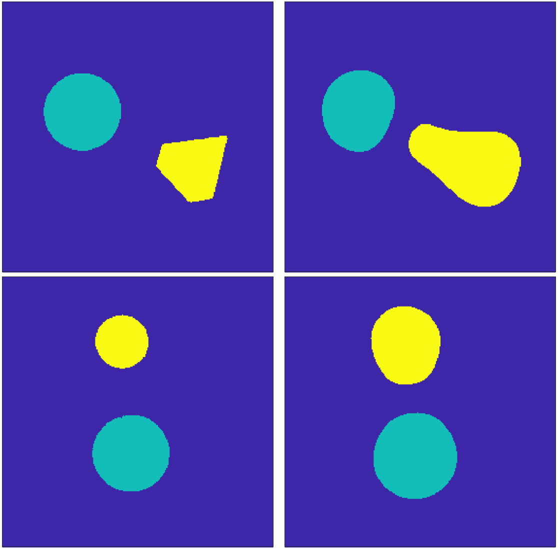
**Noise model**

- noise_std1 $= 0.05$; %standard deviation for first noise component (relative to each voltage measurement)
- noise_std2 $= 0.01$; %standard deviation for second noise component (relative to the largest voltage measurement)

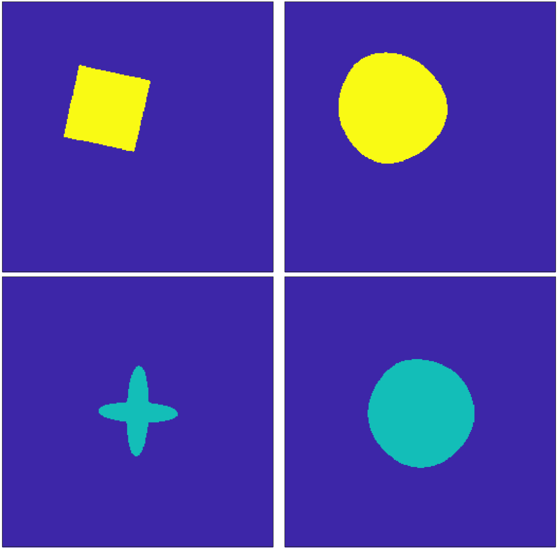
**Tikhonov-like one-step reconstruction** ($\lambda = 1$)

$$\Delta\sigma = (\mathbf{J}^T \Gamma_n \mathbf{J} + \mathbf{L}^T \mathbf{L})^{-1} \mathbf{J}^T \Gamma_n \Delta\mathbf{v} \tag{1}$$

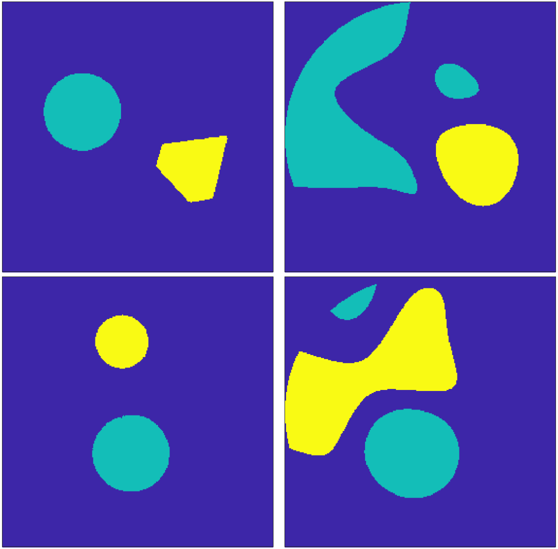
# KTC 2023 - Example code: Level 1 (Score = 0.69)



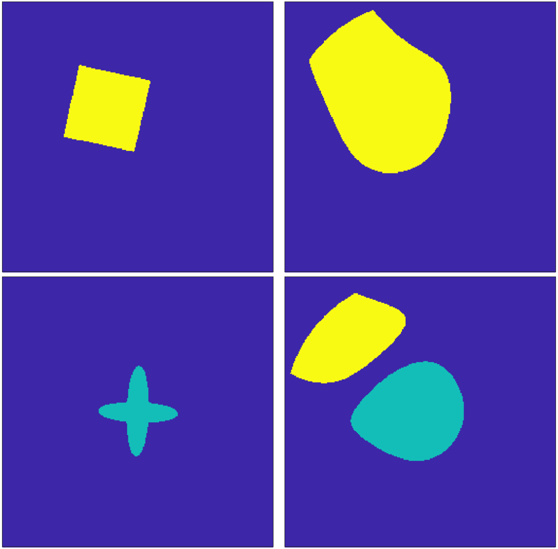
# KTC 2023 - Example code: Level 1 (Score = 0.69)



# KTC 2023 - Example code: Level 7 (Score = 0.39)

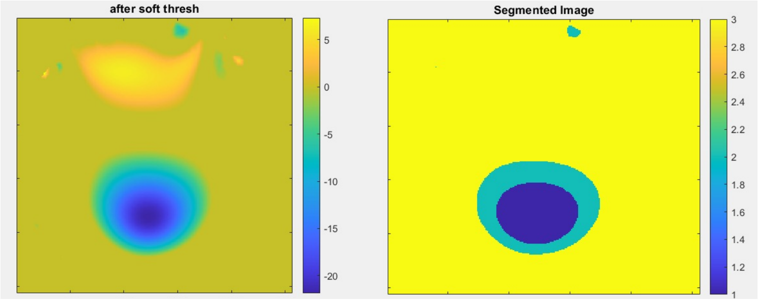


# KTC 2023 - Example code: Level 7 (Score = 0.39)



# Problems:

- Artifacts in the region of discarded electrodes
  - Artifacts amplitude may be similar to those of the targets
- Smooth solutions may result in poor Otsu's method segmentation




Here, it would be necessary a fourth label to include the yellow target

- We have to obtain reconstructions with sharp/abrupt transitions to make segmentation easier using Otsu's method

First proposal: Step-by-step
Post-processing the results to extract meaningful information from the original reconstruction and to perform segmentation

# KTC 2023: First proposal - Step 1/5

**Step 1: Smoothness Prior** - Obtain an initial reconstruction using the same original parameters as in Slide 4

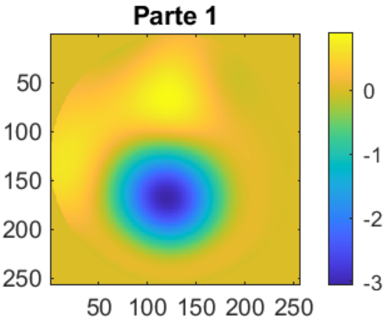
- z $= 1e-6$;
- sigma0* $= 1$;
- corrlength $= 0.115$;
- var_sigma $= 0.05^2$;
- mean_sigma $= \sigma_0$;
- noise_std1 $= 0.05$;
- noise_std2 $= 0.01$;

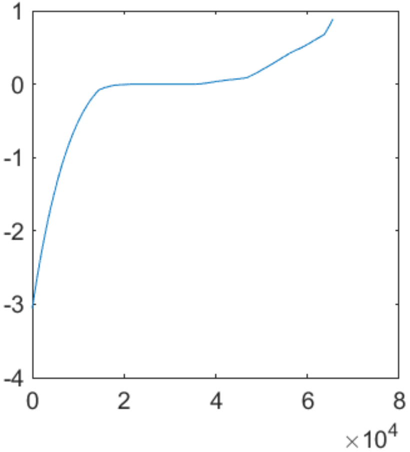
*It is possible to compare the forward operator with the water-only tank reference data. Using this estimated value did not bring clear benefits to the proposal.

The difference: including a regularization parameter in the Tikhonov-like one-step reconstruction ($\lambda = 20$)

$$\Delta\sigma = (\mathbf{J}^T \Gamma_n \mathbf{J} + \lambda \mathbf{L}^T \mathbf{L})^{-1} \mathbf{J}^T \Gamma_n \Delta\mathbf{v} \tag{2}$$

**Step 1: Smoothness Prior** with $\lambda = 20$ (Case 2, level 7)




The figure from the right is the sorted visualization, in ascending order, of the pixel values from the figure of the left.

# KTC 2023: First proposal - Step 1/5

**Legend**: Hypothesis $\mathcal{H}$, Limitations $\mathcal{L}$, and alternatives $\mathcal{A}$

**Step 1: Smoothness Prior**

- $\mathcal{H}$: From the last slide, we see that both targets are present in the reconstruction, but smoothness and artifacts degrade the image. The desired information is available in the smooth reconstructed image, but hidden.
- $\mathcal{L}$: The reconstruction is not sharp, it presents ringing and artifacts by deactivating the electrodes. The shape of the object is rounded and the targets are larger than they should be.
- $\mathcal{A}$: use alternative $\mathbf{L}$, objective criteria to choose $\lambda$, change the original functional (using TV regularization, for example), obtain the initial reconstruction using CNNs

Observations:

- This was the only step in the mesh domain. The interpolation of the conductivities at the nodes for the *pixels* was performed with the function available in the dataset

# KTC 2023: First proposal - Step 2/5

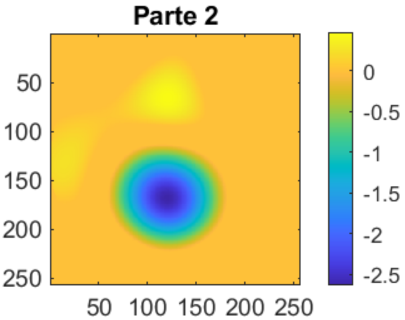
**Step 2: Soft thresholding**

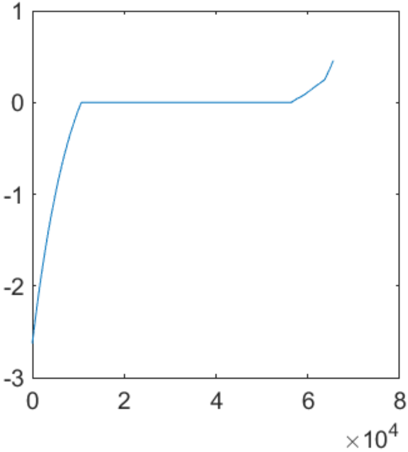
- General equations: Let $\mathbf{x}$ be a vector, $x_i$ the ith element of that vector, and T a threshold.

$$S_T(\mathbf{x}) = \begin{cases} x_i - T, & x_i > T \\ 0, & |x_i| \leq T \\ x_i + T, & x_i < -T, \end{cases} \tag{3}$$

  - en.wikipedia.org/wiki/Proximal_gradient_methods_for_learning
- The chosen threshold varies from case to case. It is a percentage of the highest value of $\Delta\sigma$ obtained from Step 1:
  - T = max(max(abs($\Delta\sigma$)))*0.14;
- Implementation is based on
  - epfl-lts2.github.io/unlocbox-html/doc/utils/soft_threshold_code.html

**Step 2: Soft thresholding** (Case 2, level 7)




The figure from the right is the sorted visualization, in ascending order, of the pixel values from the figure of the left.

**Step 2: Soft thresholding**

- $\mathcal{H}$: We want to remove small artifacts, isolate the targets, and to keep the edges of the targets smooth. It is a pre-processing to facilitate the work of the CNN in the next step.
- $\mathcal{L}$: It is difficult to choose a fixed threshold for all cases. We used a threshold depending on the values of $\Delta\sigma$, parameterized by a constant, which was chosen "by hand" and with good performance on the training data. It is not possible to guarantee a good result in the test set.
- $\mathcal{A}$: Choose another strategy to define the threshold. Using another thresholding operator.

Observation:

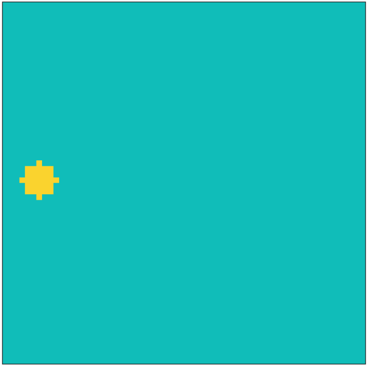
- From here on, all operations were performed in the image domain (regular grid)

**Step 3: Post-processsing using a** *CNN*

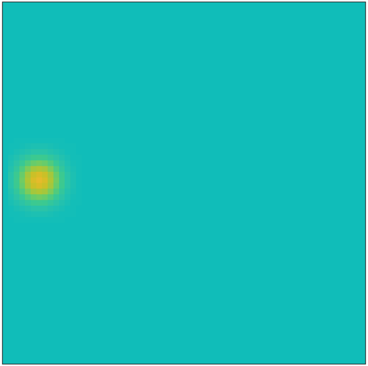
- Normalizing values between 0 and 1, where $\Delta\sigma = 0$ corresponds to 0.5 in this new scale;
- To do so, we consider the following equation:
  where C$=$ 2.5*max(max(abs($\Delta\sigma$)));
  - Using larger C values eliminates more information from the image
  - This choice of C accumulates values closer to 0.5;
- Resize the figure to $64 \times 64$ (CNN Input)
- Perform the CNN inference on the image
- Resize the figure back to $256 \times 256$

# *CNN*: Training pair examples

Deblurring-only



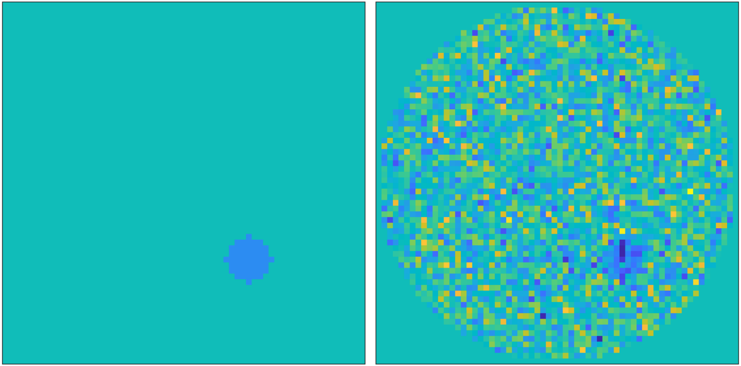
(a) $\Delta\boldsymbol{\rho}_1$



(b) $\mathbf{h}_g * \Delta\boldsymbol{\rho}_1$

Where $\mathbf{h}_g$ is a gaussian kernel.

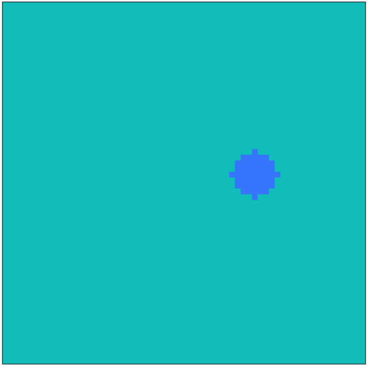
Denoising-only



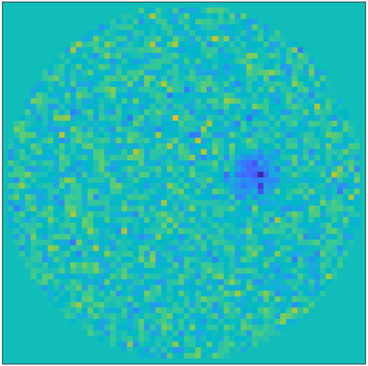
(a) $\Delta\boldsymbol{\rho}_2$ (b) $\Delta\boldsymbol{\rho}_2 + \boldsymbol{\delta}$

Where $\boldsymbol{\delta}$ is a gaussian noise.

Denoising and deblurring

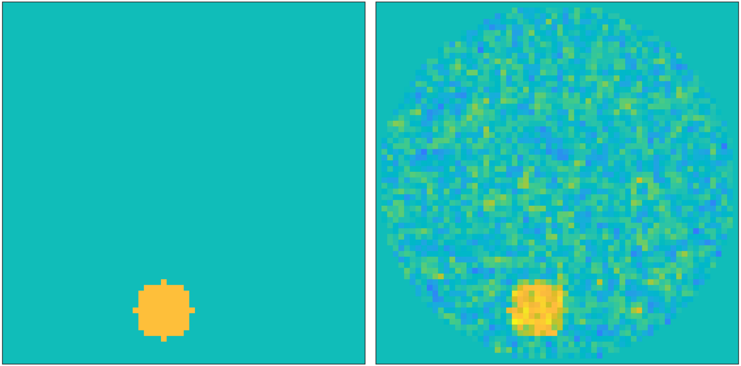


(a) $\Delta\boldsymbol{\rho}_3$



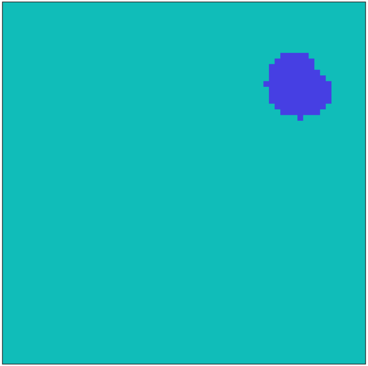
(b) $(\mathbf{h}_g * \Delta\boldsymbol{\rho}_3) + \boldsymbol{\delta}$

Denoising and deblurring

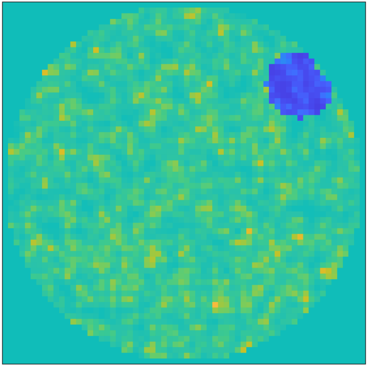


(a) $\Delta\boldsymbol{\rho}_4$ (b) $\mathbf{h}_g * (\Delta\boldsymbol{\rho}_4 + \boldsymbol{\delta})$

Denoising and deblurring



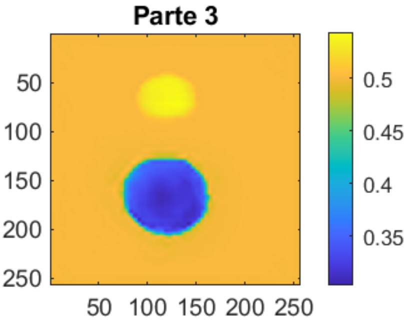
(a) $\Delta\boldsymbol{\rho}_5$

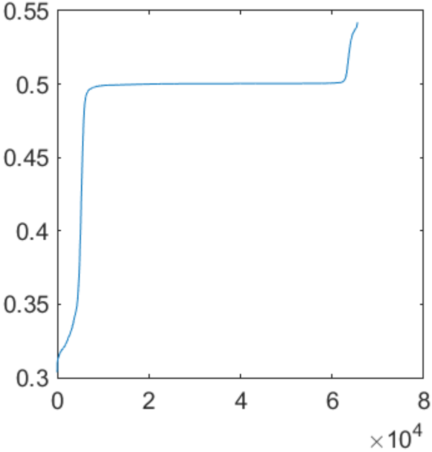


(b) $\Delta\boldsymbol{\rho}_5 + (\mathbf{h}_g * \boldsymbol{\delta})$

# KTC 2023: First proposal - Step 3/5

**Step 3: Post-processsing using a** *CNN* (Case 2, level 7)




The figure from the right is the sorted visualization, in ascending order, of the pixel values from the figure of the left.

# KTC 2023: First proposal - Step 3/5

**Step 3: Post-processsing using a *CNN***

- $\mathcal{H}$: The CNN is generic and does not provide information about EIT reconstruction. It was trained with only 1 circular target per image. Its task is to perform *denoising* and *deblurring*. In practice, as the training output was a binary image, the CNN makes the targets sharper and performs a thresholding/segmentation as well.
- $\mathcal{L}$: The CNN training was unique and we use the same weights of the CNN in all the challenge levels. When targets are not circular, such as images with artifacts, the CNN output in that region may be unpredictable. It is possible for small targets to be eliminated. It is possible for artifacts to be mistaken for targets and highlighted by mistake.
- $\mathcal{A}$: Training with images of size $256 \times 256$. Training with simultaneous targets and different shapes. Hyperparameter tuning. Training the CNN for each level to include information specific to electrode disconnection.

**Step 3: Post-processsing using a *CNN***

Other limitations:

- The CNN was trained with $64 \times 64$ images because of memory and time requirements. This requires two imresize, which interferes with the final result
- Constant $C$ was chosen by hand, with no guarantee of being the ideal value. The higher $C$, the more the CNN eliminates noise/targets.
- To train the CNN according to the difficulty level of the challenge, it would be necessary to simulate the generation of artifacts due to the disconnection of electrodes. When the input data are partial reconstructions, this is possible. In the case of proposed *CNN* we just use blurry geometric figures with additive noise.

# KTC 2023: First proposal - Step 3/5

**Step 3: Post-processsing using a** *CNN*

Observations:

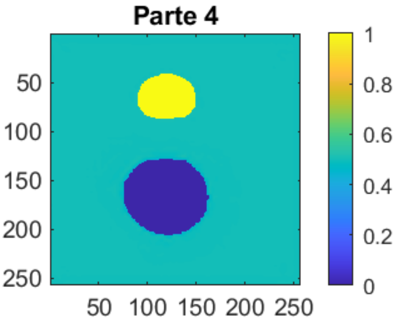
- The expected result is the segmented result.
- At each stage, it is not necessary to worry about the amplitude of the original signals, in conductivity units.
- Thus, in this proposal, it is not necessary to renormalize the CNN output to conductivity units.
- If it were an application of CNN per iteration, greater attention would be needed to the signal amplitudes

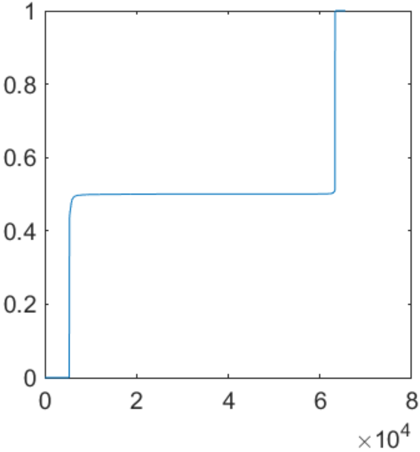
# KTC 2023: First proposal - Step 4/5

**Step 4: Pre-Segmentation**

- After normalization and inference by CNN, the $\Delta\rho$ values represent materials
  - $> 0.5$ (conductive)
  - $0.5$ (Without variation)
  - $< 0.5$ (resistive)
- However, going directly from the CNN output to the Otsu segmentation did not lead to good results, as there are variation of values within the targets.
- It is possible to do a manual pre-segmentation:
  - c1 $=$ (max(max($\Delta\sigma$)) - 0.50)./4;
  - $\Delta\sigma$($\Delta\sigma$>0.5 + c1) = 1;
  - c2 $=$ (min(min($\Delta\sigma$)) - 0.50)./4;
  - $\Delta\sigma$($\Delta\sigma$ < 0.48 + c2) = 0;

**Step 4: Pre-Segmentation** (Case 2, level 7)




The figure from the right is the sorted visualization, in ascending order, of the pixel values from the figure of the left.

**Step 4: Pre-Segmentation**

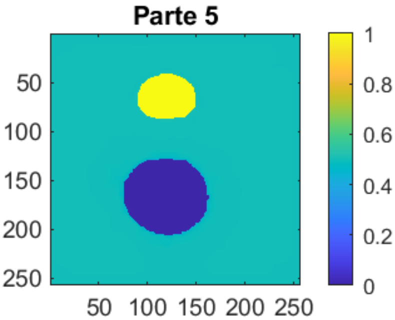
- $\mathcal{H}$: Considering that the CNN output is already a reasonably clear/sharp image, separating conductive ($>$0.5) and resistive ($<$0.5) materials is straightforward. In the end, it is nothing more than *multiple thresholding* (3 cases).
- $\mathcal{L}$: This threshold was chosen by hand and may not be ideal for all cases.
- $\mathcal{A}$: Think of another way to help Otsu's method or to use another segmentation technique instead of it.

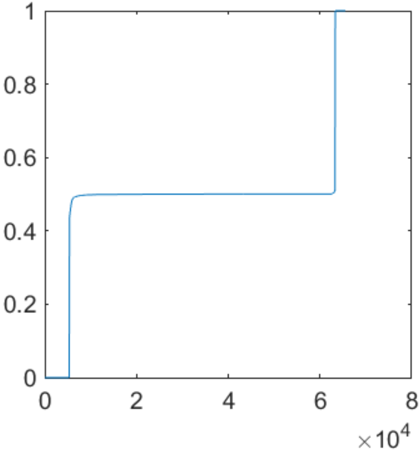
Observations: If you consider all values $> 0.5$ and $< 0.5$, many artifacts appear in the images. Therefore, it was necessary to choose a new threshold for segmentation.
Furthermore, the resulting image from Step 4 is not fully segmented, as there are values between the thresholds and 0.5, as can be seen in the sorted values graph.

**Step 5: Opening morphological filter**

- Using structuring element se $=$ strel('disk',12)
- https://www.mathworks.com/help/images/ref/strel.html
- $\mathcal{H}$: Remove small artifacts from the pre-segmented image
- $\mathcal{L}$: If the structuring element is large, small targets can be deleted.
- $\mathcal{A}$: Use other morphological filters and structuring elements

**Step 5**: Opening morphological filter (Case 2, level 7)

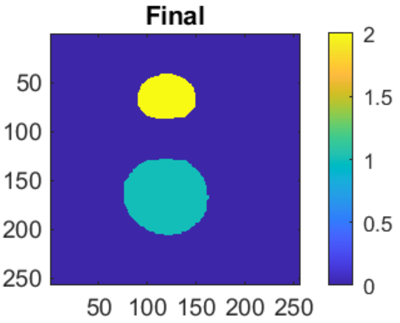


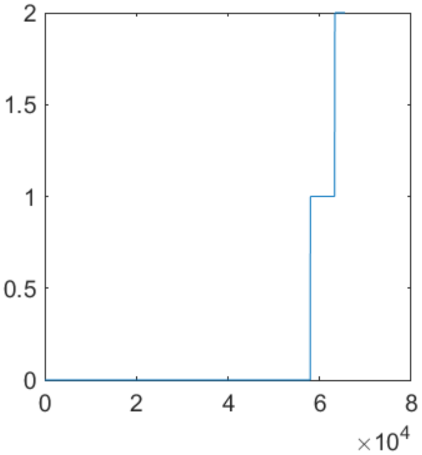




The figure from the right is the sorted visualization, in ascending order, of the pixel values from the figure of the left.

# KTC 2023: First proposal - Final reconstruction

Otsu's method segmentate and assigns the expected values to the targets
Final reconstruction (Case 2, level 7)

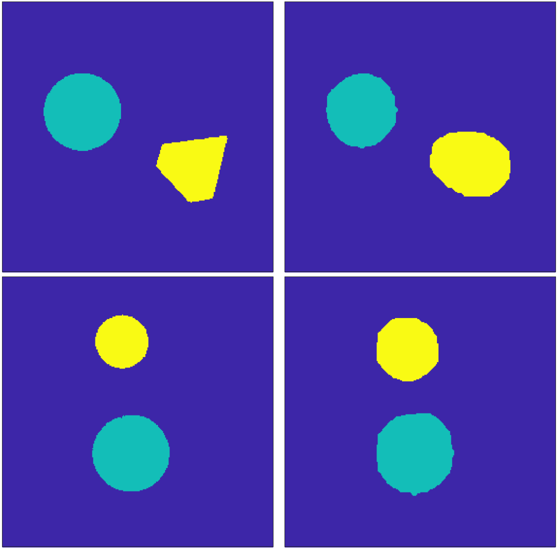


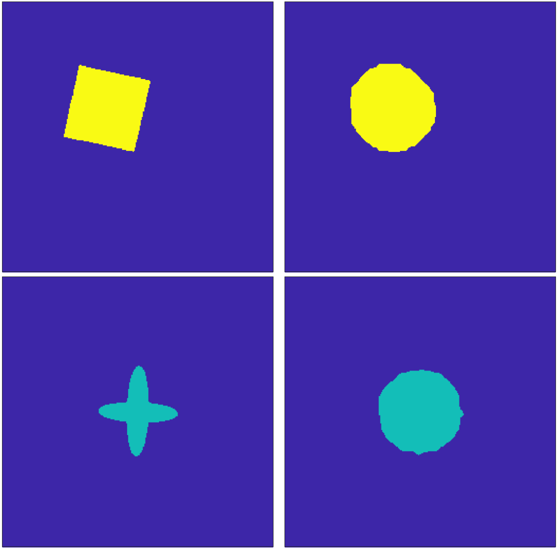

The figure from the right is the sorted visualization, in ascending order, of the pixel values from the figure of the left.

# First proposal: Level 1 results

# KTC 2023 - First proposal: Level 1 (Score = 0.81)

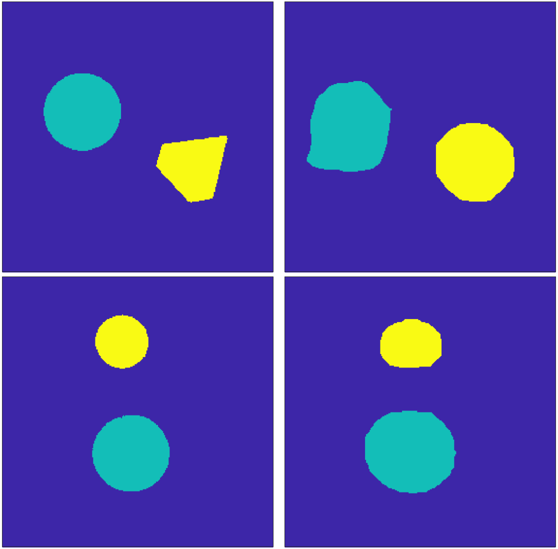


# KTC 2023 - First proposal: Level 1 (Score = 0.81)

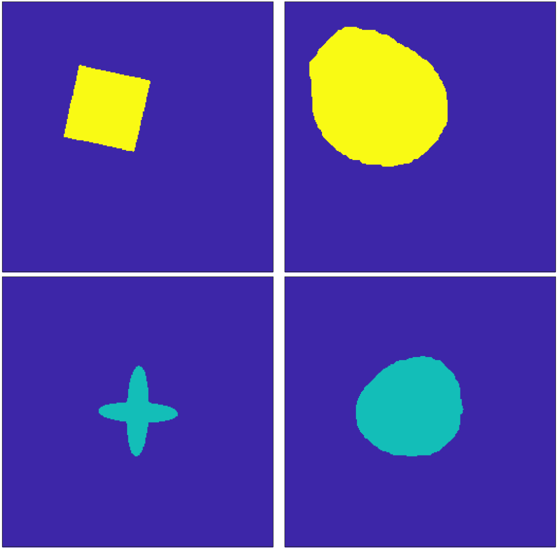


# First proposal: Level 7 results

# KTC 2023 - First proposal: Level 7 (Score = 0.69)



# KTC 2023 - First proposal: Level 7 (Score = 0.69)



# Final comments: First proposal overview

- The proposal is simple: It uses the KTC code base and uses additional post-processing steps to help segmentation using Otsu's method
- The idea of using CNNs to post-process reconstructions is not new, but several options are available depending on what exactly the task of the CNN in relation to the training pairs
- For each image, it has a maximum score less than one when using SSIM: It does not correct shape deformation/non-convexity, the targets become circular.
- On the other hand, the score did not drop much at the highest levels.
- In terms of proposal, the different thing about it is the CNN. Using the same CNN at all levels makes it less specific.

# Final comments: First proposal drawbacks

- The final segmented image is very dependent on this initial reconstruction. Some cases that could cause problems are exactly what we imagine for the higher levels:
  - Many simultaneous targets
  - Small and close targets
  - Targets with small conductivity variation in relation to water
  - Simultaneous conductive targets, but each one with different conductivities
- With so many parameters to adjust, it is not possible to garantee that they will be suitable for the test set. In other words, it is not an overfitting of the CNN itself, but of the method as a whole.
  - In other words, it is a kind of calibration that can fail if the results obtained in Step 1 of the method are very different (outside the operating range of the system)

Well, we'll give it a shot :)