SSH Super-Resolution using high resolution SST with a Subpixel Convolutional Residual Network

T. Archambault[1], A. Charantonis[2,3], D. Béréziat[1], C. Meija[2], and S. Thiria[2]

[1] Sorbonne Université, CNRS, LIP6, 75005 Paris, France
[2] Laboratoire d'Océanographie et du Climat (LOCEAN), Paris, France
[3] École nationale supérieure d'informatique pour l'industrie et l'entreprise (ENSIEE), Evry, France

The oceans have a very important role in climate regulation due to its massive heat storage capacity. Thus, for the past decades oceans have been observed by satellites in order to better understand its dynamics. Satellites retrieve several data with various spatial resolution. For instance Sea Surface Height (SSH) is a low-resolution data field where Sea Surface Temperature (SST) can be retrieved in a much higher one. These two physical parameters are linked by a physical relation that can be learned by a Super-Resolution machine learning algorithm. In this work we present a Subpixel Convolutional Deep learning model that takes advantage of the higher resolution SST field to guide the downscaling of the SSH one. The data fields that we use are simulated by a physic based ocean model at a higher sampling rate than the satellites provide. We compared our approach with a convolutional neural network (CNN) model. Our architecture generalized well with validation performances of 3.94 cm RMSE and training performances of 2.65 cm RMSE.

Super-Resolution inverse problem

We consider $\mathbf{X}_{lr}$ a low resolution image derived from a high resolution $\mathbf{X}_{hr}$ through a decimation operator denoted d as in Eq. (1). We aim to recover a high-resolution image called the Super-Resolved image $\mathbf{X}_{sr}$. In our case the Super-Resolution is performed by a deep neural network that we denote g_θ and that produce $\mathbf{X}_{sr}$ through Eq. (2). This can be seen as an optimization problem minimizing a distance $\mathcal{L}$ between $\mathbf{X}_{sr}$ and $\mathbf{X}_{hr}$, Eq. (3). In our case, we took a simple decimation operator d, as it is simply the average of every pixel in a square of size 27, and $\mathcal{L}$ is the mean squared error function.

Decimation: $$\mathbf{X}_{lr} = d(\mathbf{X}_{hr}) \quad (1)$$

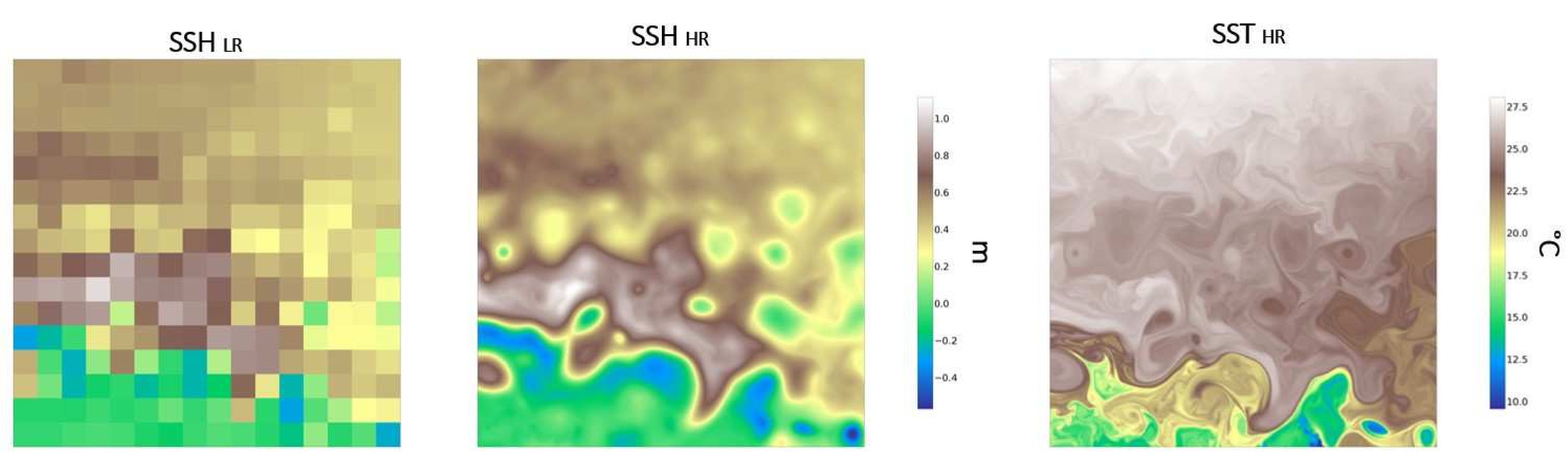
Super-Resolution: $$\mathbf{X}_{sr} = g_\theta(\mathbf{X}_{lr}, ...) \quad (2)$$

Optimization: $$\theta^* = argmin_\theta(\mathcal{L}(g_\theta(\mathbf{X}_{lr}, ...), \mathbf{X}_{hr}) \quad (3)$$

The decimation operator d is not injective, this is why we call this problem ill-posed. To find a well fitted pseudo inverse, we must constrain our problem. Here it is done by using a database and contextual information to determine which solution is physically admissible.

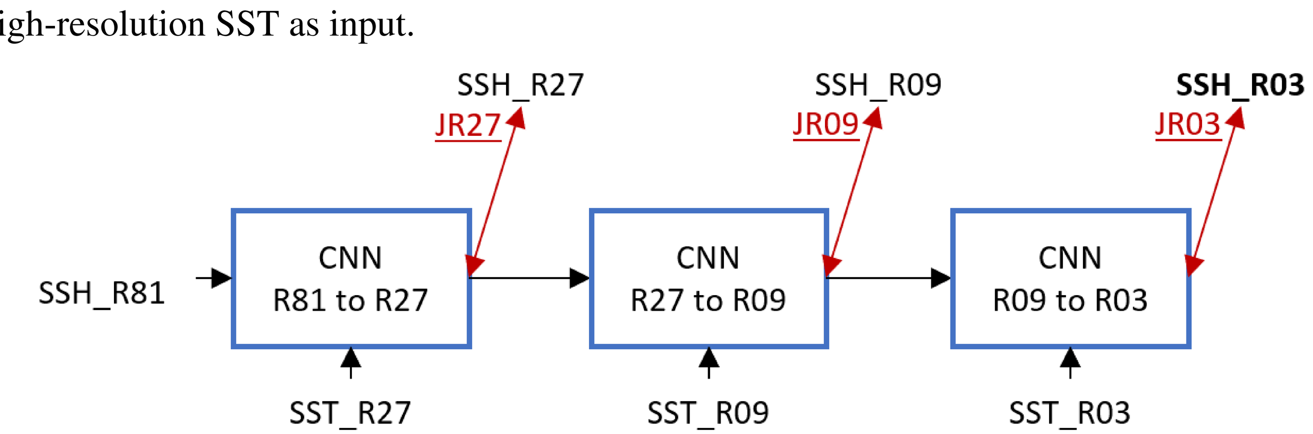
NATL60 Data

As there is no real-world SSH at the resolution that we are interested in, we use the SSH of the NATL60 model. Is is a high resolution model based on the NEMO Code and initialized with Mercator data. It has a resolution of 1/60°, that we denote R01 for one minute of arc. In this study we focus on the golf stream area from latitude 25° to 45° and from longitude 40 to 65, at a resolution of 1/30°. We used a training of 365 days of simulation starting on October 2012 and a test set 4 months in 2008.

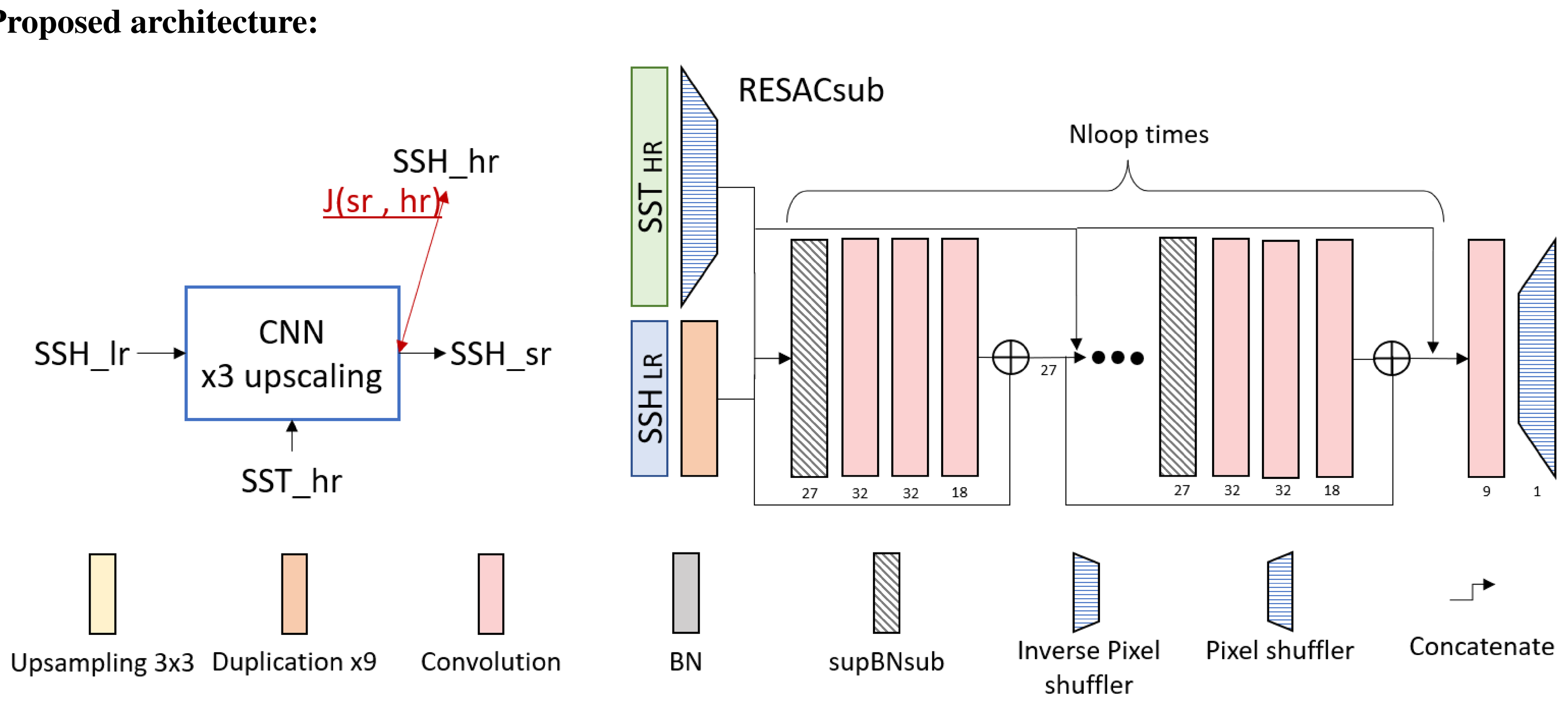


RESAC Methodology

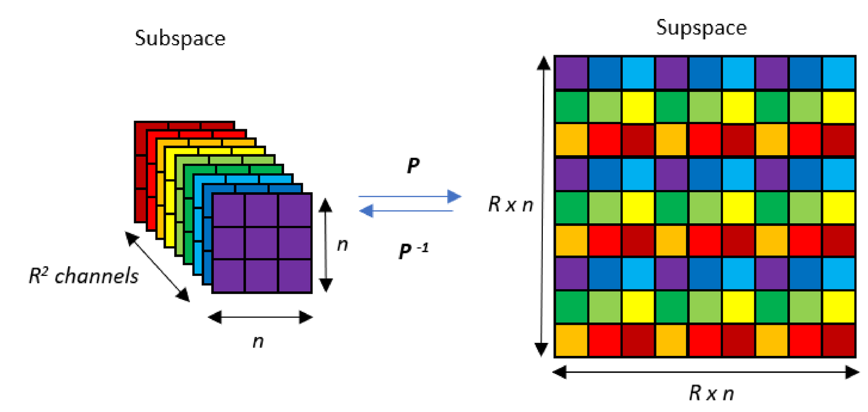
- **Control at different resolutions:** In a previous work, we introduced a methodology to deal with these problems entitled RESAC. This methodology was used to include the information on the high-resolution SST in a very large upscaling factor super-resolution. To achieve such an upscaling the neural network is cut into 3 CNN blocks, each one performing an upscaling of factor 3. Each of these blocks is controlled at its resolution and takes a low-resolution SSH with a high-resolution SST as input.



- **Proposed architecture:**

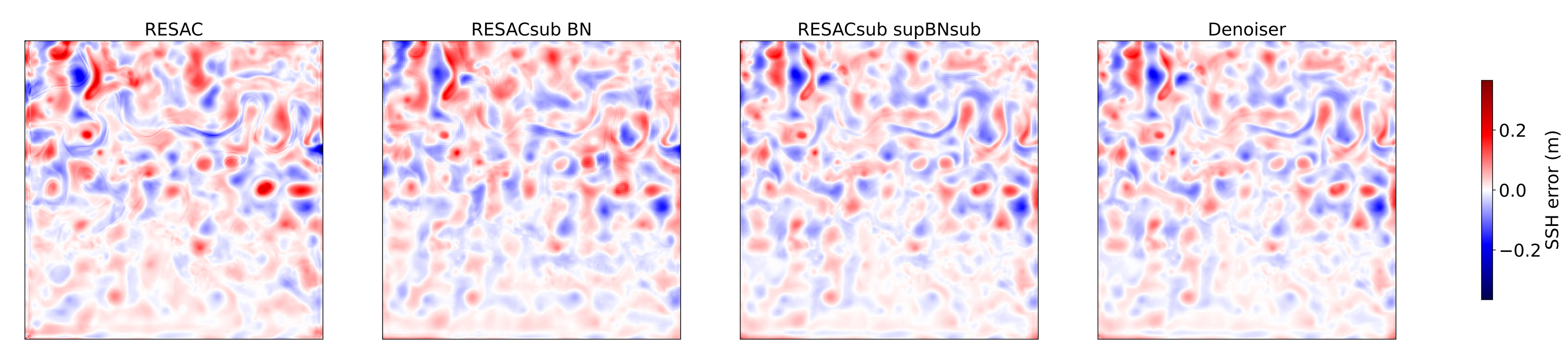


- **Subpixel convolution:** the principle of a sub-pixel convolutional layer is to perform the convolution, not in the original image space, but in a deeper and smaller space where spatial neighbors pixels are channel neighbors in the subspace. This is another manner to represent an image, which implies a pixel shuffling operator P that is a bijection between the 2 spaces.



Results

- **Difference map**

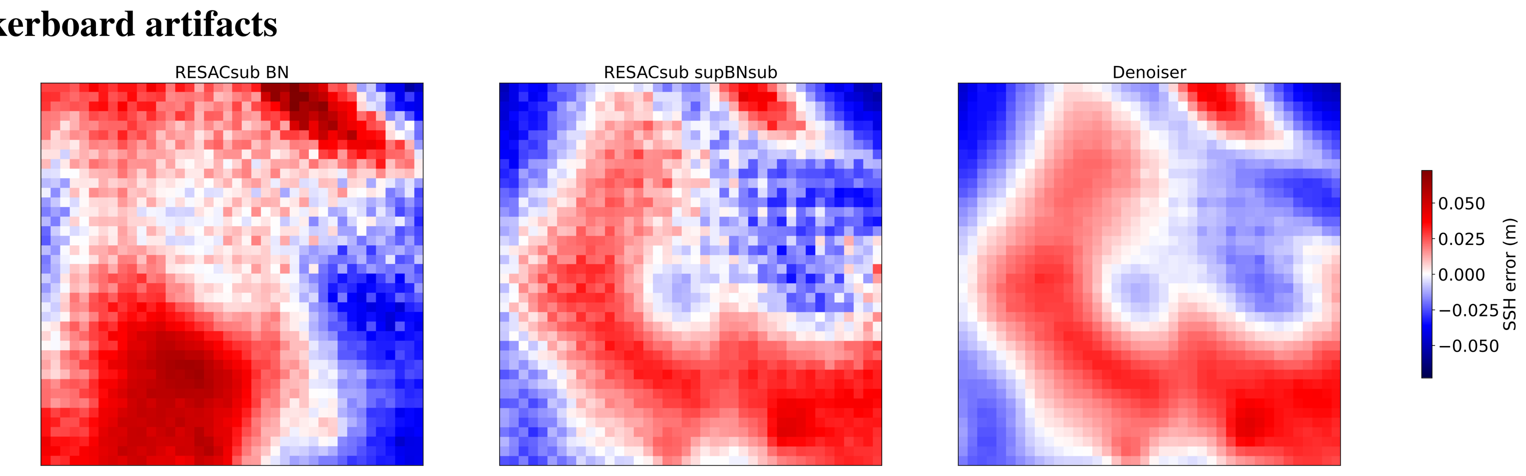


- **RMSE comparison**

Model	RESAC	sub BN	supBNsub	Denoiser	Bicubic
weights	344,976	335,442	334,722	51,841	unsupervised
RMSE (cm)	5.50 ± 0.47	4.43 ± 0.14	4.00 ± 0.26	3.94 ± 0.28	6.94
RMSE 1st dec (cm)	8.03 ± 0.91	5.33 ± 0.76	5.09 ± 0.93	5.03 ± 0.99	6.28
RMSE 10th dec (cm)	4.65 ± 0.14	4.78 ± 0.22	4.42 ± 0.12	4.36 ± 0.12	5.11
RMSE cropped (cm)	5.24 ± 0.48	4.22 ± 0.14	3.82 ± 0.25	3.80 ± 0.27	6.89

Mean and standard deviation scores on 10 trainings of each architecture with different weight initializations. The scores are given on the validation data set. We compare the models in RMSE (root mean squared error): the global RMSE is given, along with the RMSE on the first and the last decile of the target image. We also give a cropped RMSE (the RMSE of a smaller interior image to avoid border effects)

- **Checkerboard artifacts**



Conclusion

- SST information helps to constrain the Super-Resolution inverse problem
- Our method achieved a very large downscaling with a factor of 27: the subpixel network achieved a 3.94cm error where the fully convolutional network achieved a 5.50 cm error and the bicubic 6.94 cm error.
- We used a subpixel convolution and an adapted form of Batch Normalization. These methods created checkerboard artifacts so we used a denoising network with large filters to smooth the result.
- How to transfer this learning to real-world data with no ground truth: in a real-world scenario, we have access to along tracks measurements that are then interpolated into gridded data. The problem can then be seen as an interpolation problem.