Super-Resolution Appearance Transfer for 4D Human Performances - Supplementary Material

Marco Pesavento Marco Volino Adrian Hilton
Centre for Vision, Speech and Signal Processing
University of Surrey, UK
{m.pesavento,m.volino,a.hilton}@surrey.ac.uk

This document presents additional details and qualitative results for the SRAT pipeline introduced in the main paper. Section 1 further explains Equation 2 of the paper. Section 2 explains the training procedure for RCAN [10]. Section 3 defines the metrics used for evaluating our pipeline stages. Section 4 presents the limitations of SRAT and Section 5 offers further evaluations of the SRAT stages. Visual results of the colour mapping stage are shown in Section 5.1. Other results for the super-resolution (SR) stage are presented in Section 5.2. In Section 5.3, there are additional considerations on the performances of different configurations of SRAT.

1. Colour transfer algorithm

To learn the colour transfer function $\phi_\theta(x)$ that aims to transfer the appearance from a high-resolution (HR) static capture of a human subject to a dynamic video performance capture of the same subject, we introduce a parametric energy function (Equation 2 in the main paper):

$$\theta = \frac{1}{N}\sum_{l=1}^{N} \underset{\theta_l}{\operatorname{argmin}}\{||p_f||^2 - 2 < p_f|p_I >\} \qquad (1)$$

This function is minimized with gradient descent algorithm where:

$$||p_f||^2 = \sum_{i=1}^{K}\sum_{k=1}^{K} \mathcal{N}(0; \phi_\theta(\mu_f^{(k)}) - \phi_\theta(\mu_f^{(i)}), 2h^2 I)\pi_f^{(k)}\pi_f^{(i)} \qquad (2)$$

$$< p_f|p_I > = \sum_{i=1}^{K}\sum_{k=1}^{K} \mathcal{N}(0; \phi_\theta(\mu_f^{(k)}) - \mu_I^{(i)}, 2h^2 I)\pi_f^{(k)}\pi_I^{(i)} \qquad (3)$$

p_f is the distribution of the input frame image I_{LR} while p_I is the distribution of the correspondent HR image I_{HR}. These two images are paired during the first step of the second stage of SRAT (couples identification).
p_f is a Gaussian mixture distribution of parameter K defined by the Normal distribution $\mathcal{N}(0; \phi_\theta(\mu_f), h^2 I)$ computed at 0 with parameterised mean $\phi_\theta(\mu_f)$ and covariance $h^2 I$ (h is a control bandwidth and I is the identity matrix). p_I is a Gaussian Mixture distribution of parameter K as well, but its mean μ_I is not parameterised.
In Equation 2, the norm is computed as the subtraction of the parameterised mean of two cluster i and k of p_f while in Equation 3 the mean of p_I is subtracted to the parameterised one of p_f. The weights $\pi_{f\ i=1,\dots,K}^{(i)}$ as well as $\pi_{I\ k=1,\dots,K}^{(k)}$ are chosen equiprobable with $\pi_f^{(i)} = 1/K$ and $\pi_I^{(k)} = 1/K$ respectively.
K is a fixed number defining the K variable of the K-means algorithm. K-means selects K clusters in the colours of the input images to define their Gaussian mixture distributions p_f and p_I before the minimization step [2].
In other words, the algorithm first selects K clusters in the colours of the target (I_{LR}) and of the reference (I_{HR}) image. Then, it models the colour distribution as a Gaussian mixture distribution whose mean is parameterised only for I_{LR}. Finally, the energy function of Equation 1 is minimized and $\phi_\theta(x)$ is defined from a set of input couples of I_{LR} and I_{HR}.

2. Implementation of RCAN

In this work we use RCAN-style network [10] to super resolve texture map. In order to achieve this, we adapt the training protocol and utilise datasets of human texture maps to train and fine-tune RCAN.

Training datasets: RCAN presents a very deep architecture and the larger the architecture, the more data is needed to produce viable results. A common problem derives from the lack of human texture maps available for training.
We first train RCAN with a dataset made of texture maps of 7 human models downloaded from [1]. To augment the training data and to ease the training, we crop the texture maps into patches with the size of 256x256. The final dataset presents 1872 patches.
To enhance RCAN performances, we fine-tune it with original texture maps of the input model retrieved in a preprocessing stage. In particular, we use 410 texture maps for *SingleF* model and 441 for *SingleM*. For every differ-

ent input model of SRAT, RCAN can be fine-tuned with the original texture maps of the specific model.

Training process: RCAN was trained for 800 epochs with the patches of texture maps and then fine-tuned for 200 epochs with the original texture maps of the input model. In each training batch, 16 low-resolution (LR) color patches (48x48 in size) are extracted as inputs. The learning rate is initially set to 10^{-4} and then halved every 10^5 iterations of back-propagation. ADAM optimizer [4] is used with β_1=0.9, β_2=0.999 and $\epsilon = 10^8$.

3. Evaluation metrics

Colour transfer stage: to evaluate the effect of the colour transfer from the HR images to the input frames, we consider their normalized colour histograms. We compute the Jensen-Shannon (JS) divergence and the Chi-Squared (χ^2) distance. Specifically, we measure the dissimilarity of the colour histogram of a frame with the colour histogram of the HR image which was paired with the frame in the colour correction step of the colour mapping stage of SRAT. The JS divergence measures the similarity of two statistical distributions by subtracting the average of the two distribution entropies with the entropy of their average [9]:

$$JS(H, H') = \sum_{l=1}^{N}(H_i \log(\frac{2H_i}{H_i+H_i'} + H_i' \log(\frac{2H_i'}{H_i+H_i'}))) \quad (4)$$

The χ^2 distance weights with higher importance the dissimilarity between two small bins of the histograms [9]:

$$\chi^2(H, H') = \sum_{l=1}^{N} \frac{2(H_i - H_i')^2}{H_i + H_i'} \quad (5)$$

where, for both the equations, H is the normalized colour histogram of the frame, H' is the normalized colour histogram of the HR image, N is the the number of bins of each histogram and H_i is the value of the i^{th} bin of H [9]. We select $N = 64$. The presented results are computed considering all the frames of *SingleF* (7040) and *SingleM* (7520) models.

Texture-map SR stage: the SR texure maps are evaluated with PSNR and SSIM [13] on Y channel of transformed YCbCr space. These metrics are computed by testing the network with 104 texture maps of *SingleF* and with 113 texture maps of *SingleM*.

4. Limitations

SRAT only enhances the appearance of the model. Therefore, if the initial reconstruction produces low quality 3D shapes, our pipeline is not able to modify the geometry of the model and the final output will still have a low quality shape. Another limitation is during the couple identification step of the colour mapping stage when Densepose is not able to detect the human model. The partial texture map is in this case a black image and the couple cannot be created. If the selected couples for learning the colour transfer function do not cover the surface entirely, the corrected frames present unnatural colours. Figure 1 illustrates some of these limitations.

(a) (b) (c)

Figure 1: Failure cases: (a) 4D reconstruction, (b) *SingleM* colour transfer, (c) *SingleF* colour transfer.

5. Further results evaluation

In this section, additional visual results are shown for the evaluation studies of the main paper. A detailed analysis of the performances of different configurations is undertaken as well.

5.1. Colour mapping stage evaluation

Input couple selection: visual results of failure cases caused by not giving as input to the colour transfer algorithm enough couples to cover the whole surface are shown in Figure 2 for 1 input couple and Figure 3 for 2 input couples for *SingleF*. For *SingleM* model, the results with 1 couple as input are illustrated in Figure 4 and with 2 couples in Figure 5. In the learning stage not all the colours of the model are present in the input couples and in the reference palette some colours of the model are therefore missing. The outputs show how the colours of the specific parts of the subject that were not seen during the learning stage are unnatural and influenced by the colours of other parts of the subject.

Comparison with TPS [2]: our colour transfer algorithm is based on TPS [2]. This method, as explained in the main paper, learns the colour transfer function from only a pair of target and reference image. Therefore, the algorithm learns a different function for every input frame because it models the colour distributions applying K-means algorithm. The learned colour transfer function can vary every time K-means is applied. The colours of the corrected outputs may differ for consecutive frames as well as for the same frame of different cameras. Examples of the former case are shown in Figure 6 for *SingleF* and Figure 8 for *SingleM*. The second failure case is illustrated in Figure 7 for *SingleF* and Figure 9 for *SingleM*. If TPS is applied, the colours of the clothes and face of the models change in consecutive frames and when they are acquired with different cameras. On the contrary, the colours do not change if our method is applied in the aforementioned cases .

Config.	Colour transfer (s)	Super resolution (s)	Average time per frame (s)	Total time SingleF (hours)	Total time SingleM (hours)
SRAT (ours)	14.13	128.49	354.57	43.33	46.29
1	8.45	128.49	136.94	16.73	17.87
2	35.16	128.49	63.65	20	21.36
3	14.13	197.19	3381.12	413.24	441.42
4	56.71	197.19	4062.4	496.51	530.36
5	35.16	197.19	759.75	389.91	416.49

Table 1: Performance of SRAT pipeline configurations.

5.2. Texture-map super resolution stage evaluation:

Different training models of RCAN: Figure 10 shows the outputs of all the considered training models of RCAN for *SingleF* while Figure 11 for *SingleM*. A portion of the texture-map and its correspondent heatmap are illustrated. The heatmap highlights the dissimilarity between the ground-truth and the SR texture map: in a scale from blue to red colours, the blue one indicates that two correspondent pixels of the ground-truth and the SR texture maps are the most similar while the red colour represents the pixels which are most dissimilar. The heatmap of the best case (1b) adopted in our pipeline presents more blue and less green/red pixels compared to the others.

Comparison with SR networks: Figure 12 shows a portion of the texture-map and its correspondent heatmap for *SingleF* and Figure 13 for *SingleM* for the considered SR networks. Also in this case, the SRAT heatmap shows more blue pixels for both the models.

Final model outputs: Figure 14 shows details of the *SingleF* 3D model when the output SR texture maps of SRAT are rendered to the meshes. Figure 15 illustrates *SingleM*. As it is possible to see in the hairs and in the band of *SingleF* and in the button of *SingleM*, the details of the super resolved models appear less blurry than their LR version.

Figure 16 shows the output 3D models of SRAT, whose appearance is significantly enhanced compared to their original version, with brighter colours.

5.3. Different configurations of SRAT

Table 1 presents the performances of the studied configurations of SRAT (Section 4.4 of the main paper). We measure the time in seconds (s) to complete the colour transfer step and the SR texture map stage for both *SingleF* and *SingleM* models. The average time to process a single frame on the two stages is also presented. In this case, there are 16 input images (one for each camera) and 1 texture map per frame. In addition, we compute the total time (in hours) to process all the frames for each performance. *SingleF* has 440 frames per camera and 7040 total frame images and 440 texture maps in total. *SingleM* presents 470 frames per camera (7520 frame images and 470 texture maps). The size of the input frame images is 3840x2160 pixels (7680x4320 pixels when super resolved by a factor of 2) and the texture map size is 2048x2048 pixels (4096x4096 pixels when super-resolved by a factor of 2). For the super-resolution stage an NVIDIA GeForce RTX 2070 was used. If the colour mapping stage was processed with multiple CPUs and the SR stage with multiple GPUs, the processing time would be lower than the one shown in Table 1. We do not measure the time of the other stages because is constant in all the configurations. The fastest configurations are the 1st and the 2nd but their outputs present visible artefacts as shown in the main paper. The proposed SRAT pipeline is the third fastest and produces the best visual results.

References

[1] Renderpeople. https://renderpeople.com/. Accessed: 2020-07-26.

[2] M. Grogan and R. Dahyot. L2 divergence for robust colour transfer. *Computer Vision and Image Understanding*, 181:39–49, 2019.

[3] M. Haris, G. Shakhnarovich, and N. Ukita. Deep back-projection networks for super-resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1664–1673, 2018.

[4] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[5] Y. Li, V. Tsiminaki, R. Timofte, M. Pollefeys, and L. V. Gool. 3d appearance super-resolution with deep learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9671–9680, 2019.

[6] Z. Li, J. Yang, Z. Liu, X. Yang, G. Jeon, and W. Wu. Feedback network for image super-resolution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3867–3876, 2019.

[7] Y. Mei, Y. Fan, Y. Zhou, L. Huang, T. S. Huang, and H. Shi. Image super-resolution with cross-scale non-local attention and exhaustive self-exemplars mining. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[8] F. Yang, H. Yang, J. Fu, H. Lu, and B. Guo. Learning texture transformer network for image super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5791–5800, 2020.

[9] Q. Zhang and R. L. Canosa. A comparison of histogram distance metrics for content-based image retrieval. In *Imaging and Multimedia Analytics in a Web and Mobile World 2014*, volume 9027, page 90270O. International Society for Optics and Photonics, 2014.

[10] Y. Zhang, K. Li, K. Li, L. Wang, B. Zhong, and Y. Fu. Image super-resolution using very deep residual channel attention networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 286–301, 2018.

[11] Z. Zhang, Z. Wang, Z. Lin, and H. Qi. Image super-resolution by neural texture transfer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7982–7991, 2019.

[12] H. Zheng, M. Ji, H. Wang, Y. Liu, and L. Fang. Crossnet: An end-to-end reference-based super resolution network using

cross-scale warping. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 88–104, 2018.

[13] Zhou Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004.

Figure 2: On the right, 1 input couple to the colour correction step of SRAT. On the left, the corrected output for *SingleF*.

Figure 3: On the right, 2 input couples to the colour correction step of SRAT. On the left, the corrected output for *SingleF*.

Figure 4: On the right, 1 input couple to the colour correction step of SRAT. On the left, the corrected output for *SingleM*.

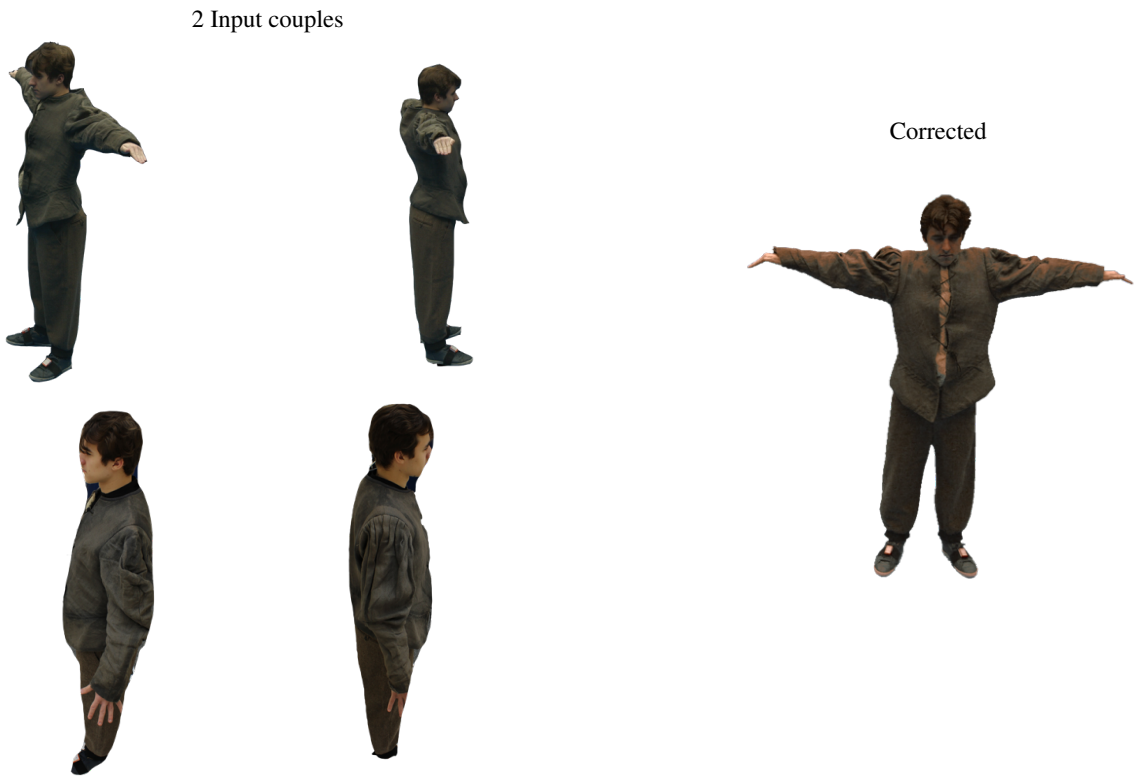

Figure 5: On the right, 2 input couples to the colour correction step of SRAT. On the left, the corrected output for *SingleM*.

Figure 6: Consecutive corrected frames for TPS [2] (top row) and SRAT (bottom row) of *SingleF model*.

Figure 7: Same corrected frame of different cameras for TPS [2] (top row) and SRAT (bottom row) of *SingleF model*.

Figure 8: Consecutive corrected frames for TPS [2] (top row) and SRAT (bottom row) of *SingleM model*.

Figure 9: Same corrected frame of different cameras for TPS [2] (top row) and SRAT (bottom row) of *SingleM model*.

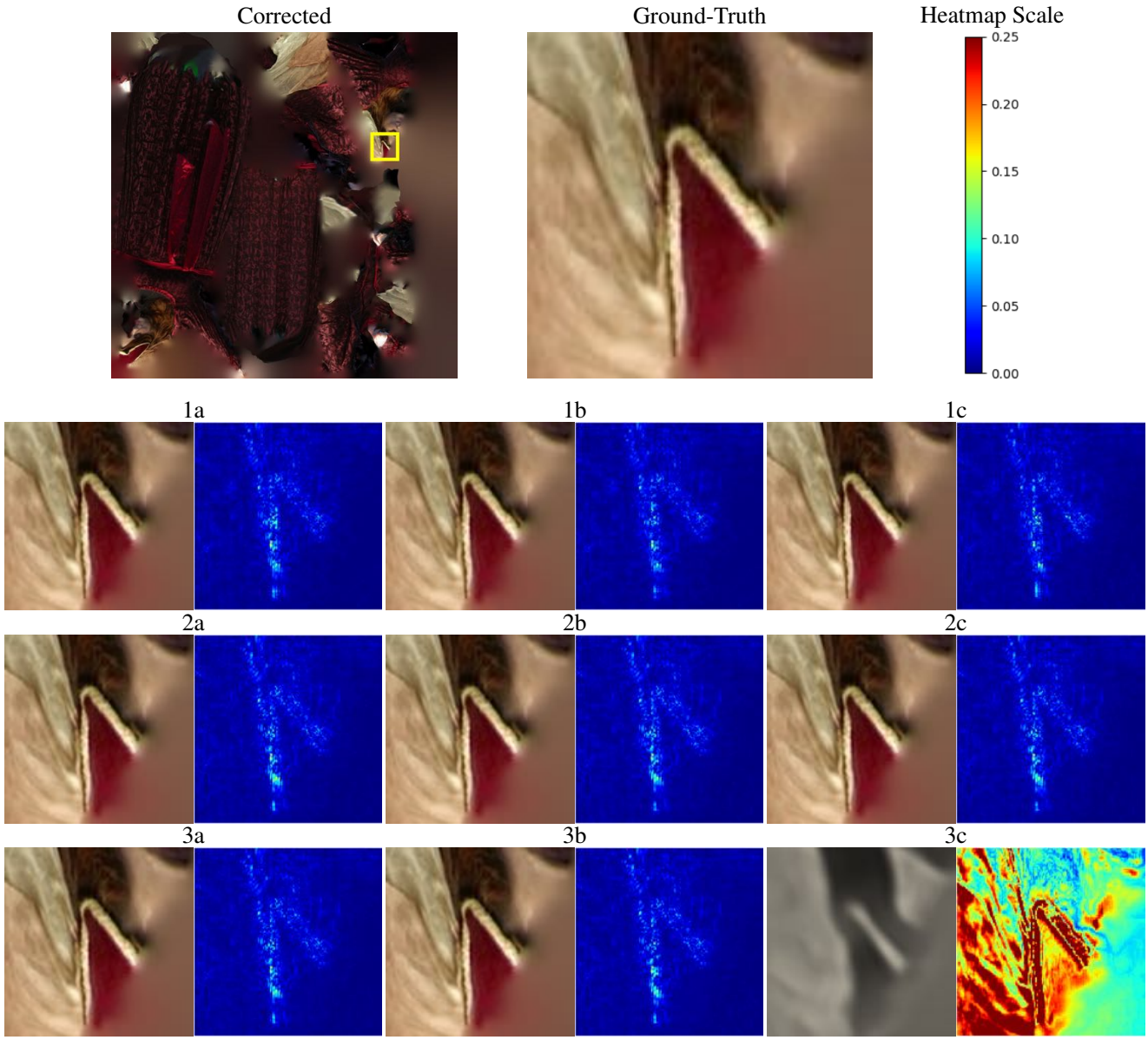

Figure 10: Portion of *SingleF* texture maps and correspondent heatmaps for different trained models of RCAN.

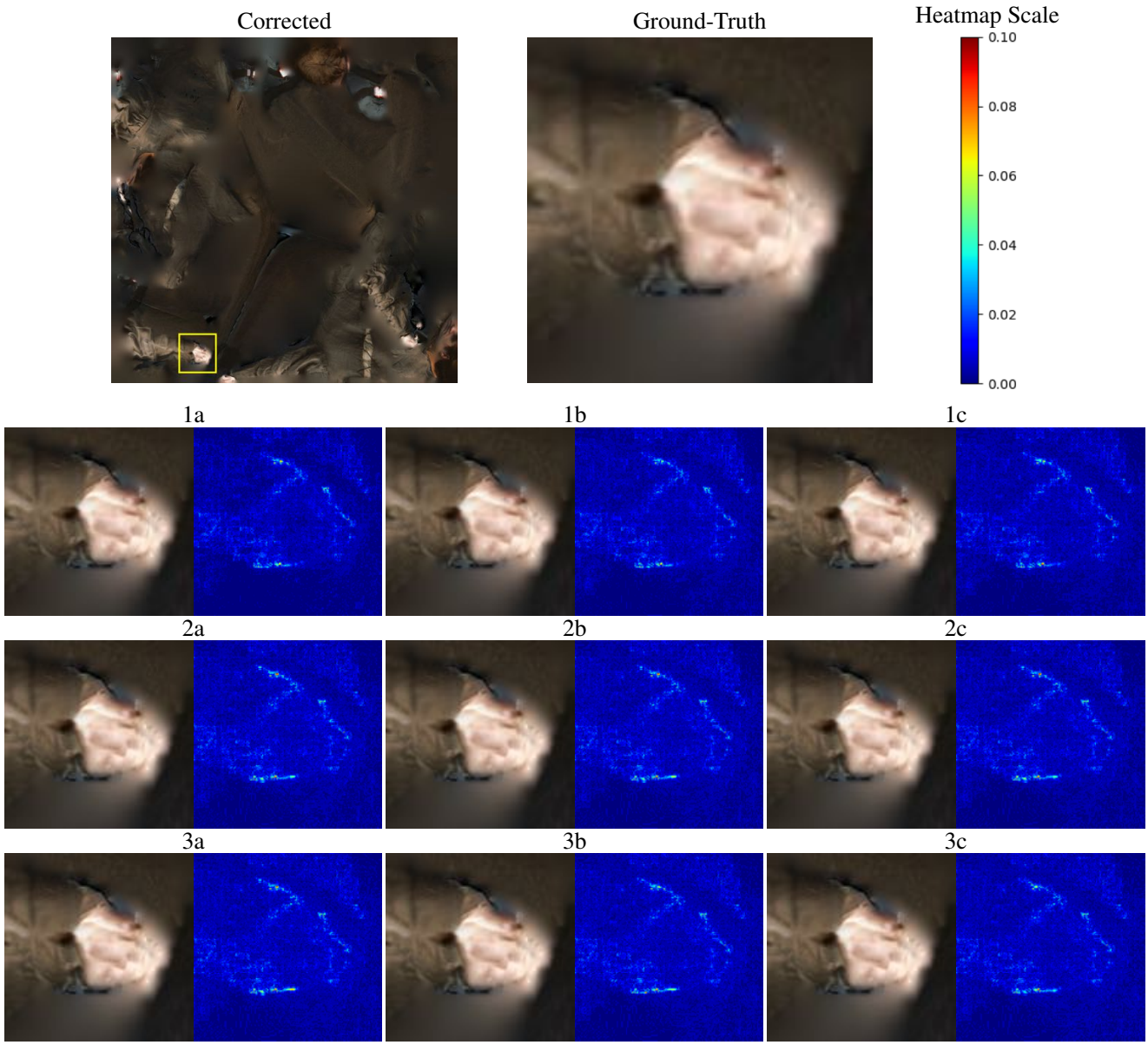

Figure 11: Portion of *SingleM* texture maps and correspondent heatmaps for different trained models of RCAN.

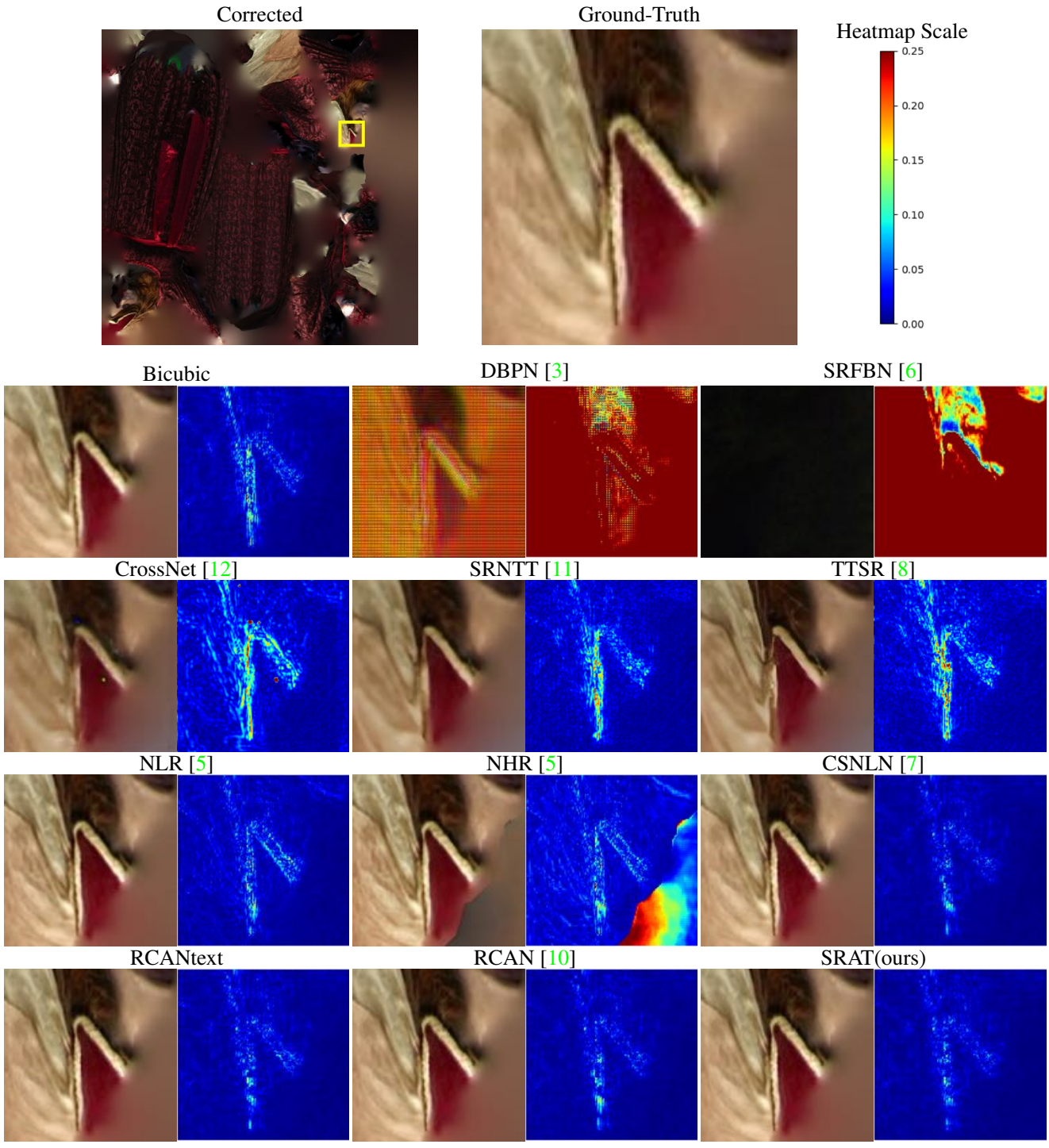

Figure 12: Portion of *SingleF* texture maps and correspondent heatmaps for different SR networks.

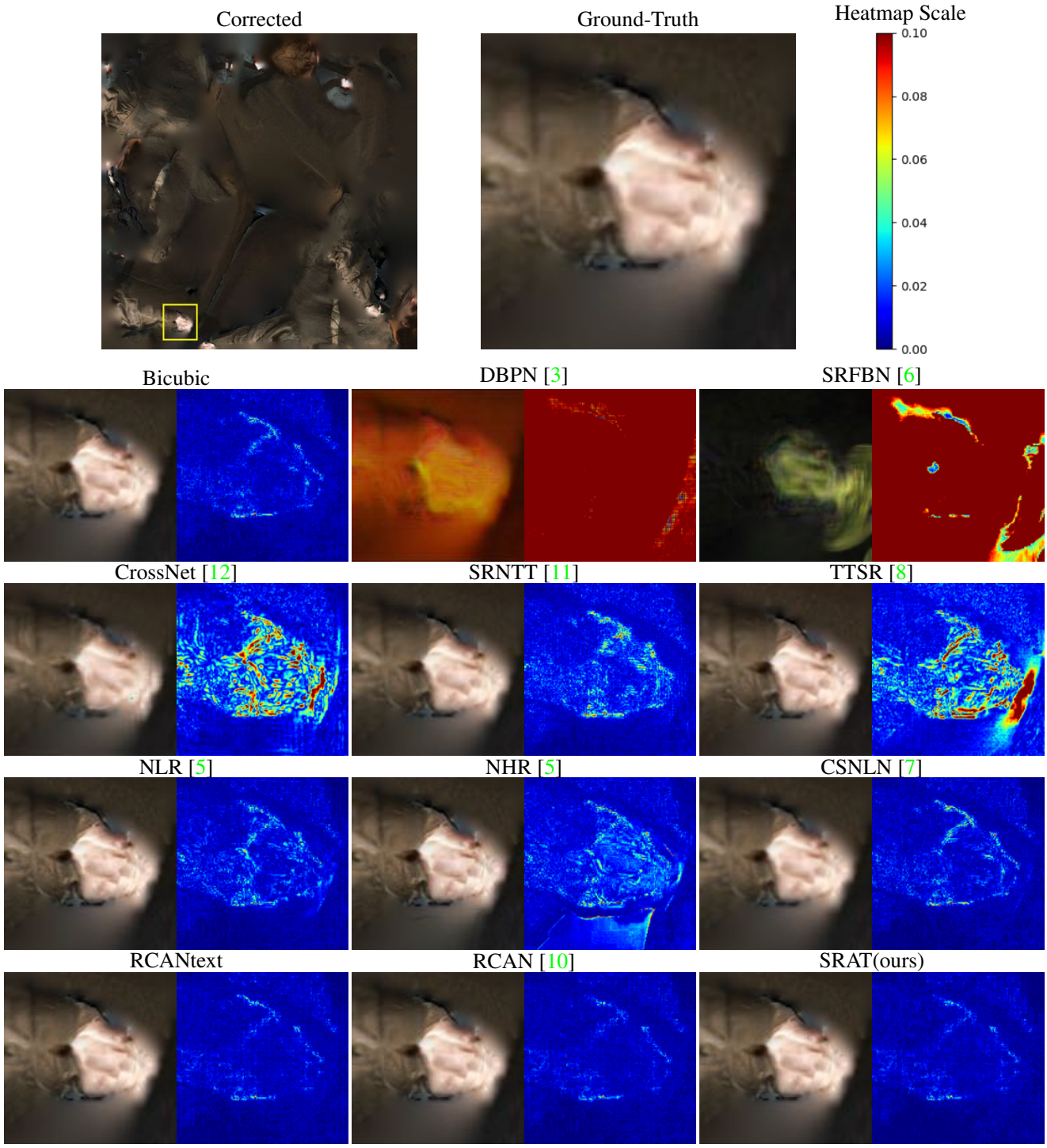

Figure 13: Portion of *SingleM* texture maps and correspondent heatmaps for different SR networks.

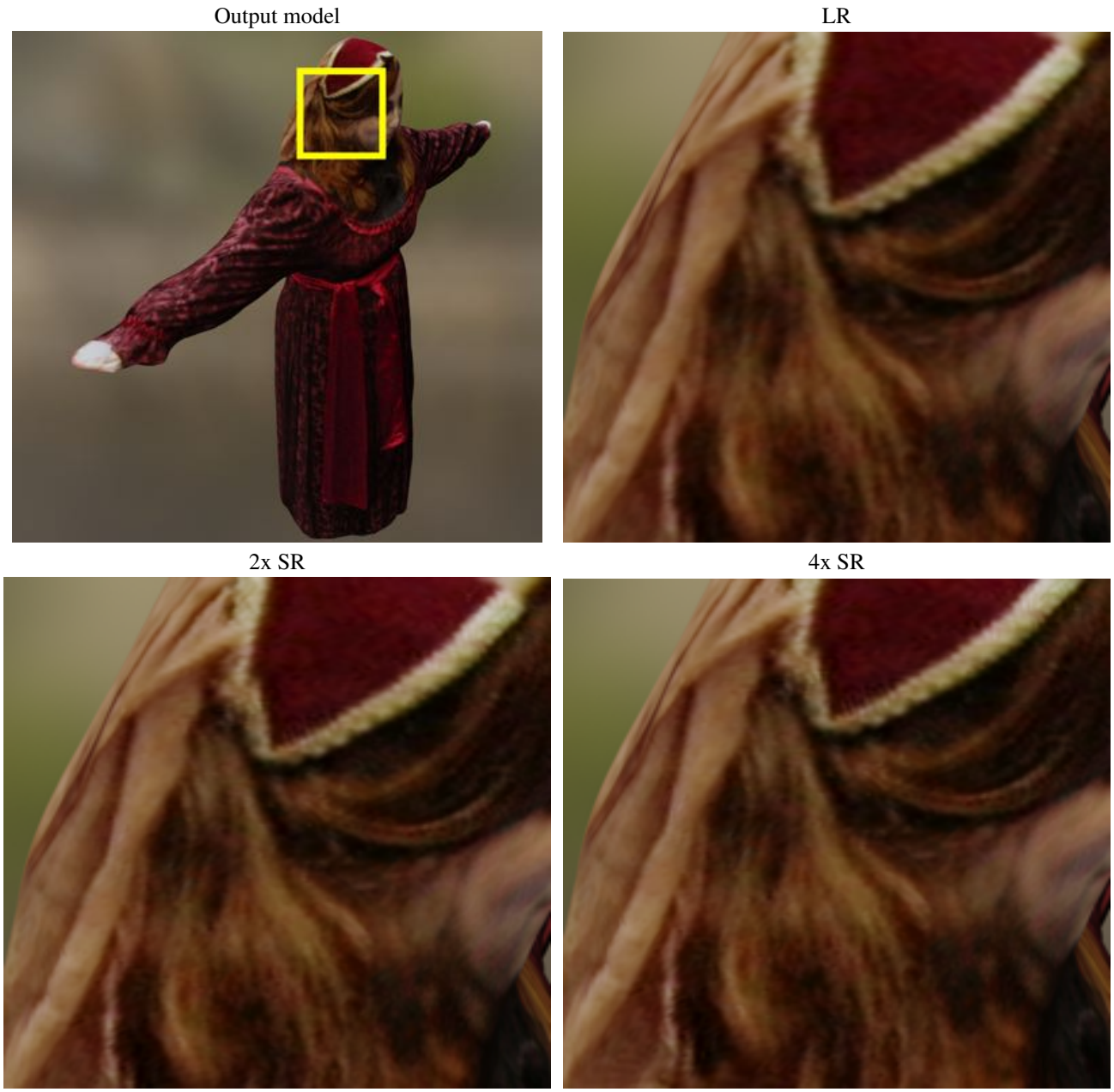

Figure 14: Details of the SR output texture map applied to *SingleF* model considering different upscaling factors.

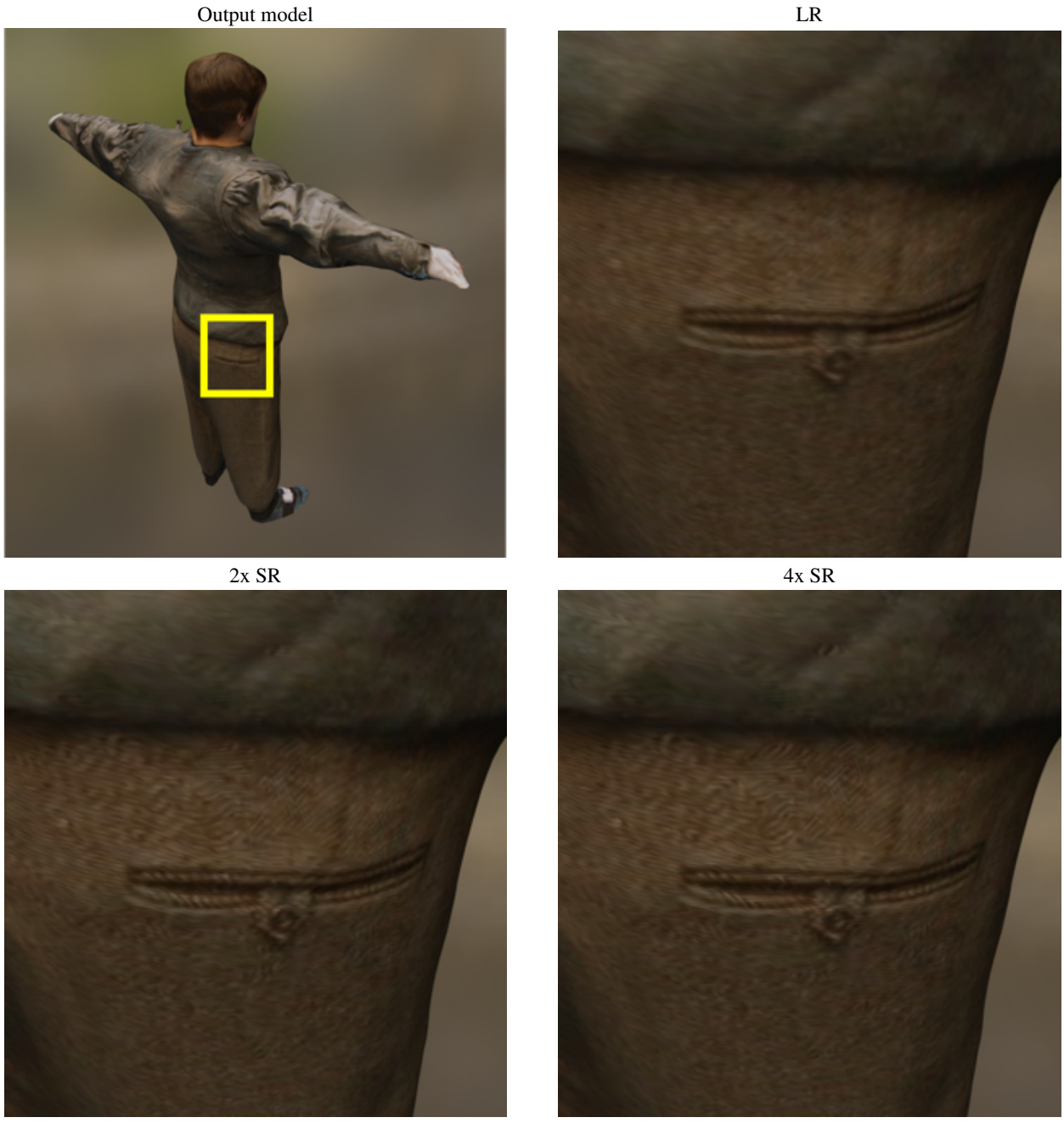

Figure 15: Details of the SR output texture map applied to *SingleM* model considering different upscaling factors.

Original *SingleF*

Output *SingleF*

Original *SingleM*

Output *SingleM*

Figure 16: Comparison between input and output models.