EUROGRAPHICS 2011 / M. Chen and O. Deussen
(Guest Editors)

*Volume 30 (2011), Number 2*

# Estimating Color and Texture Parameters for Vector Graphics (Additional Material)

S. Jeschke[1] and D. Cline[2] and P. Wonka[3]

[1] Vienna University of Technoloy
[2] Oklahoma State University
[3] Arizona State University

**Abstract**

*This document presents results that we generated using the techniques described in the paper.*

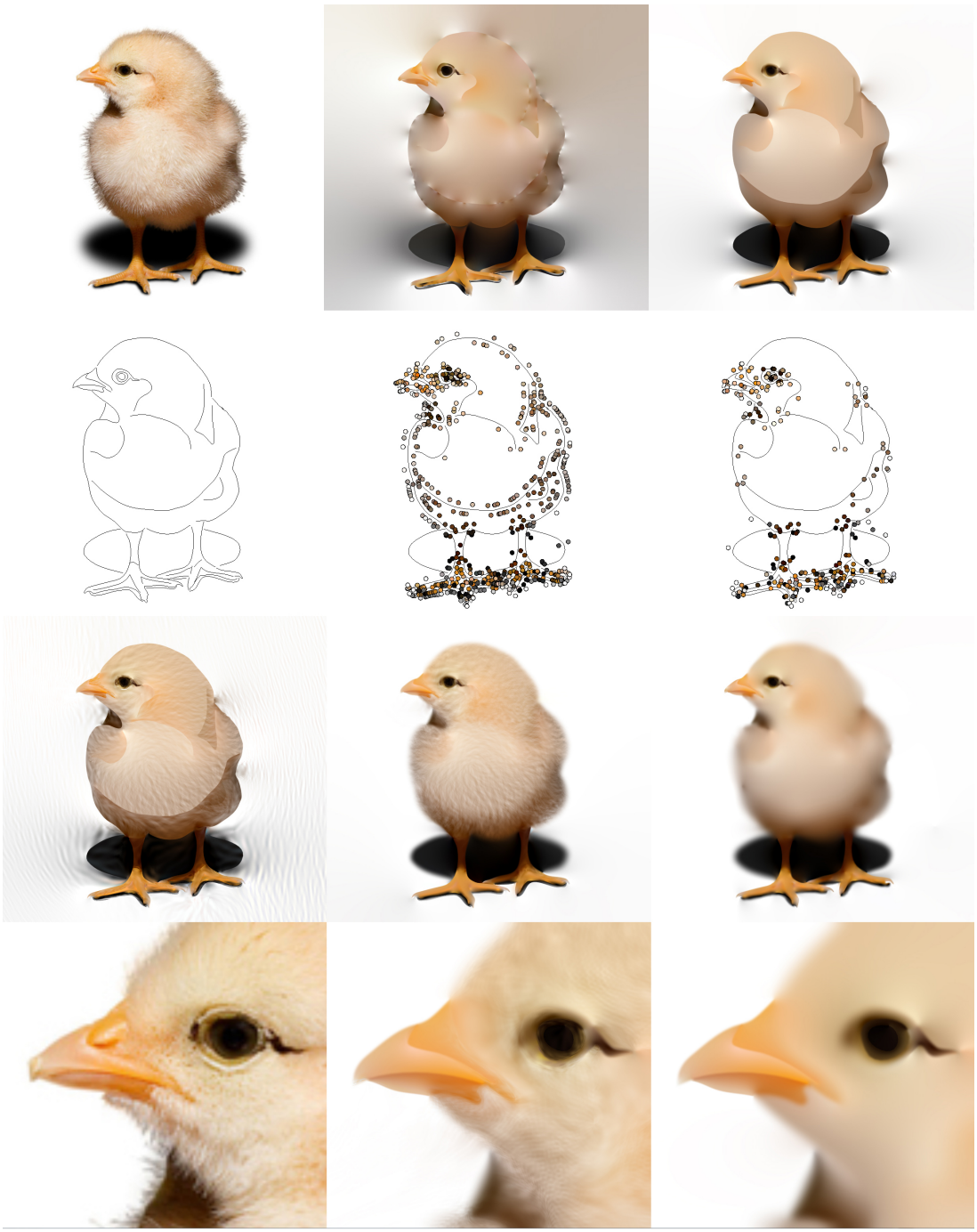

**Figure 1:** *The chicken example. Top row (from left to right): original image, result by [Orzan et al. 08], our result. Second row (from left to right): input curves, result by [Orzan et al. 08] (665 color points), our result (281 color points). Third row (from left to right): result of the automatic noise fitting, final result after manual editing, final result without textures (but with colors and blur edited). Bottom row (from left to right): original image closeup, final textured closeup, closeup without textures.*
*Note how the feather directions are mostly correctly estimated by the automatic algorithm. Even the inflection point on the chicken's stomach was correctly found and reproduced by the diffused curve directions. Near the chicken silhouette the directions mostly follow this silhouette, however. This is because the Gabor filters latch onto the gradient between chicken and background.*
*Also note how the approach of Orzan et al. wrongly picks many colors from the chicken at silhouettes due to the curve placement in the fuzzy region, which causes the dark background.*

**Figure 2:** *The Lena example. Top row (from left to right): original image, result by [Orzan et al. 08], our result. Second row: input curves, result by [Orzan et al. 08] (1178 color points), our result (682 color points). Third row: result of the automatic noise fitting, final result after manual editing, final result without textures (but with colors and blur edited). Bottom row: original image closeup, final textured closeup, closeup without textures.*
*While the colored DCI can faithfully represent most of Lena, the shawl, hat and hair textures would need very many curves for this. The gabor noise does not correctly reproduce all these regions as in the original image, but the stylization gives a similar impression. Again, the directions and scale of the textures were satisfactorily found by our automatic algorithm, which reduced the time needed for manual editing.*
*Also note how the approach of Orzan et al. produces many color points around textured regions in contrast to our approach.*

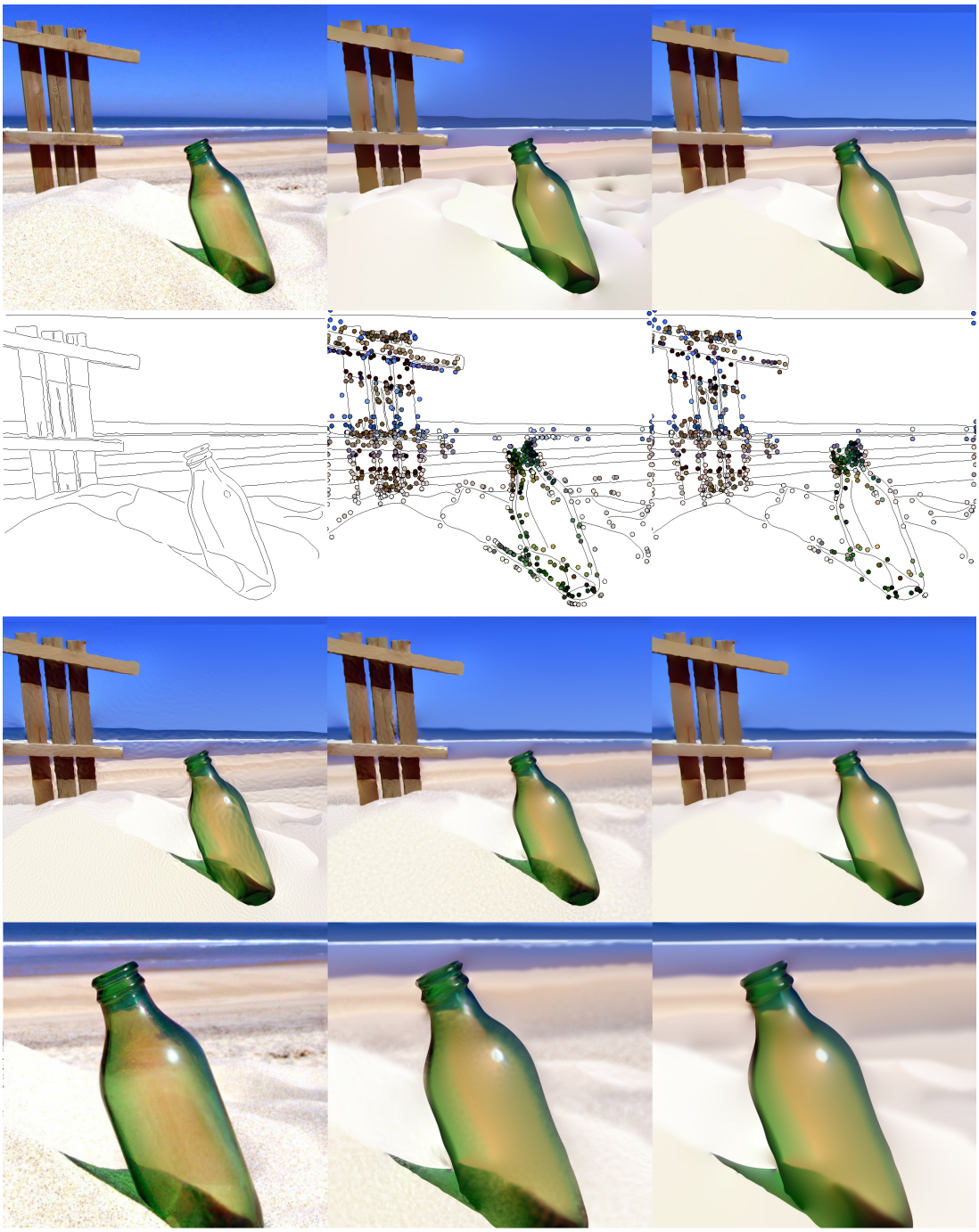

**Figure 3:** *The beach example. Top row (from left to right): original image, result by [Orzan et al. 08], our result. Second row (from left to right): input curves, result by [Orzan et al. 08] (850 color points), our result (586 color points). Third row (from left to right): result of the automatic noise fitting, final result after manual editing, final result without textures (but with colors and blur edited). Bottom row (from left to right): original image closeup, final textured closeup, closeup without textures.*
*In this example most curves were automatically found by an algorithm similar to [Orzan et al 08]. This causes the wrinkled look of the curves surrounding the wood, for example.*
*While Orzans method works well for most of the image, the right part of the sky is too dark as there are no curves placed in it. In contrast, our algorithm includes these regions as well.*
*Also note how the Gabor noise represents the subtle textures in the bottle (especially visible in the closeup), which were manually modeled using our interactive tool.*

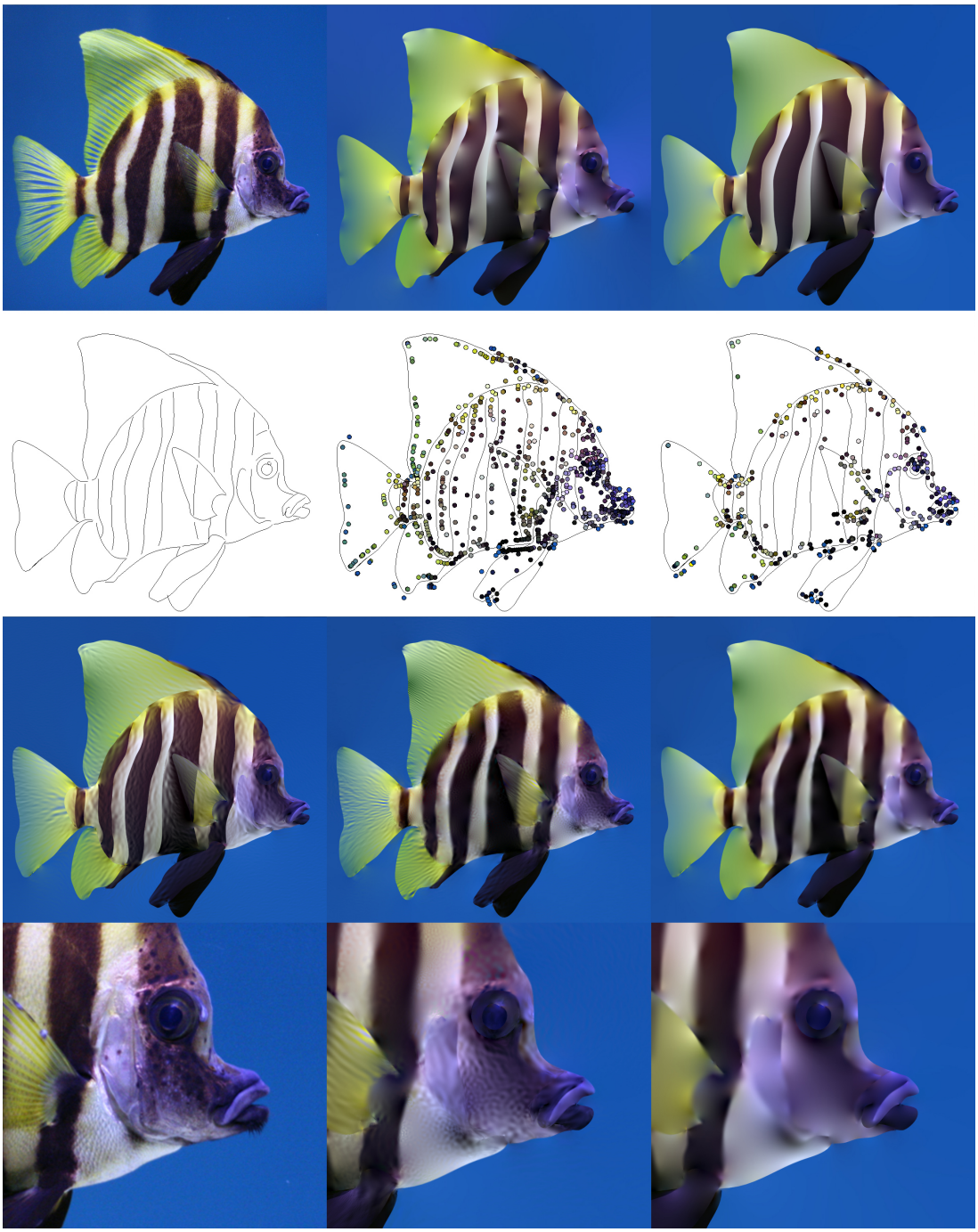

**Figure 4:** *The fish example. Top row (from left to right): original image, result by [Orzan et al. 08], our result. Second row (from left to right): input curves, result by [Orzan et al. 08] (754 color points), our result (361 color points). Third row (from left to right): result of the automatic noise fitting, final result after manual editing, final result without textures (but with colors and blur edited). Bottom row (from left to right): original image closeup, final textured closeup, closeup without textures.*
*The directions of the fin textures are correctly found by the automatic algorithm. In this example the textured fins significantly increase the expressiveness of the fish.*
*Also note the realistic scale texture in the bottom row, although this was not found by the automatic algorithm.*

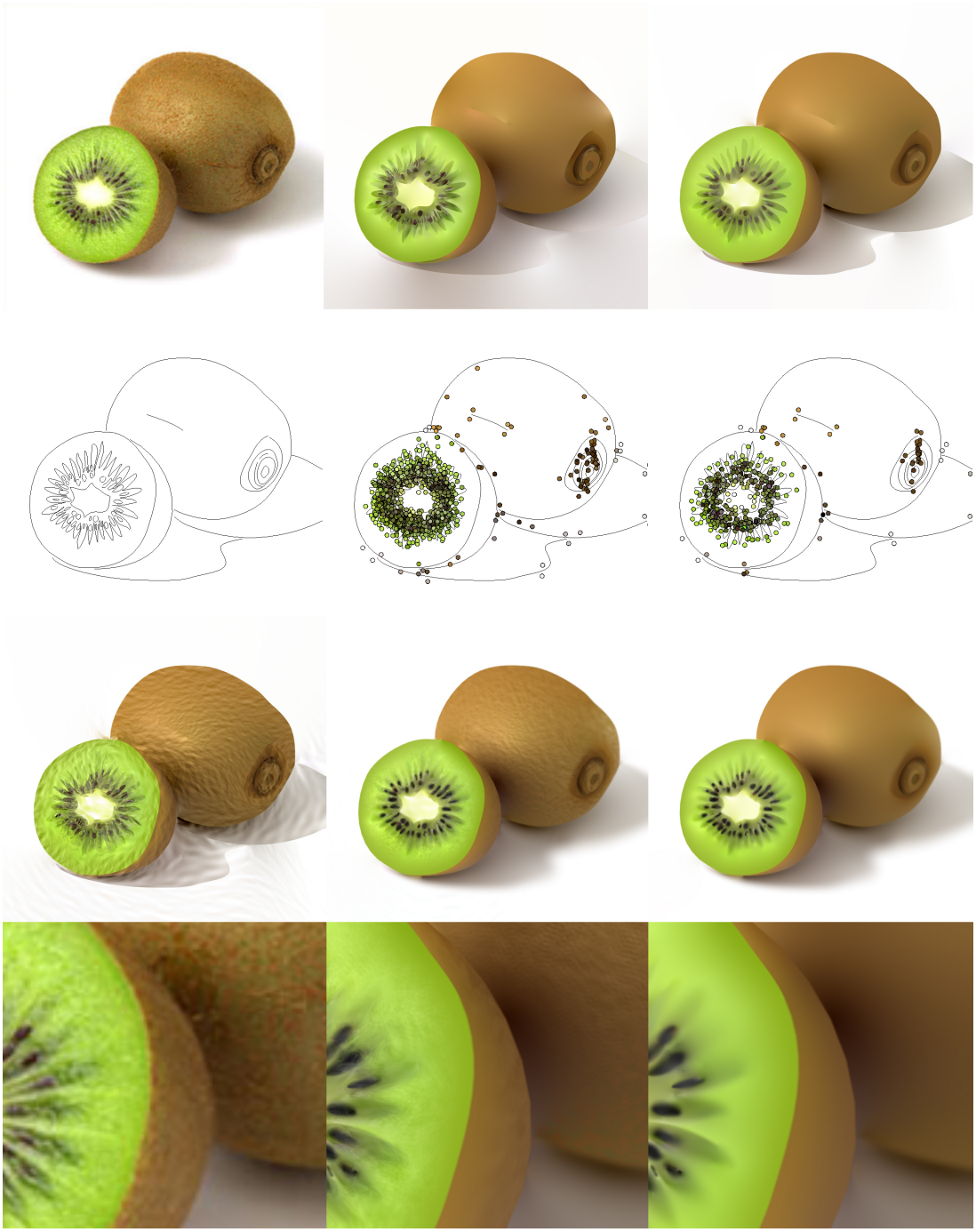

**Figure 5:** *The kiwi example. Top row (from left to right): original image, result by [Orzan et al. 08], our result. Second row (from left to right): input curves, result by [Orzan et al. 08] (1211 color points), our result (426 color points). Third row (from left to right): result of the automatic noise fitting, final result after manual editing, final result without textures (but with colors and blur edited). Bottom row (from left to right): original image closeup, final textured closeup, closeup without textures.*
*The curve coloring method of Orzan et al. works well for this model due to the smoothly or only lightly textured regions. However, the light textures still take the vector graphics its overly "clean" look and lets the image appear closer to the original.*

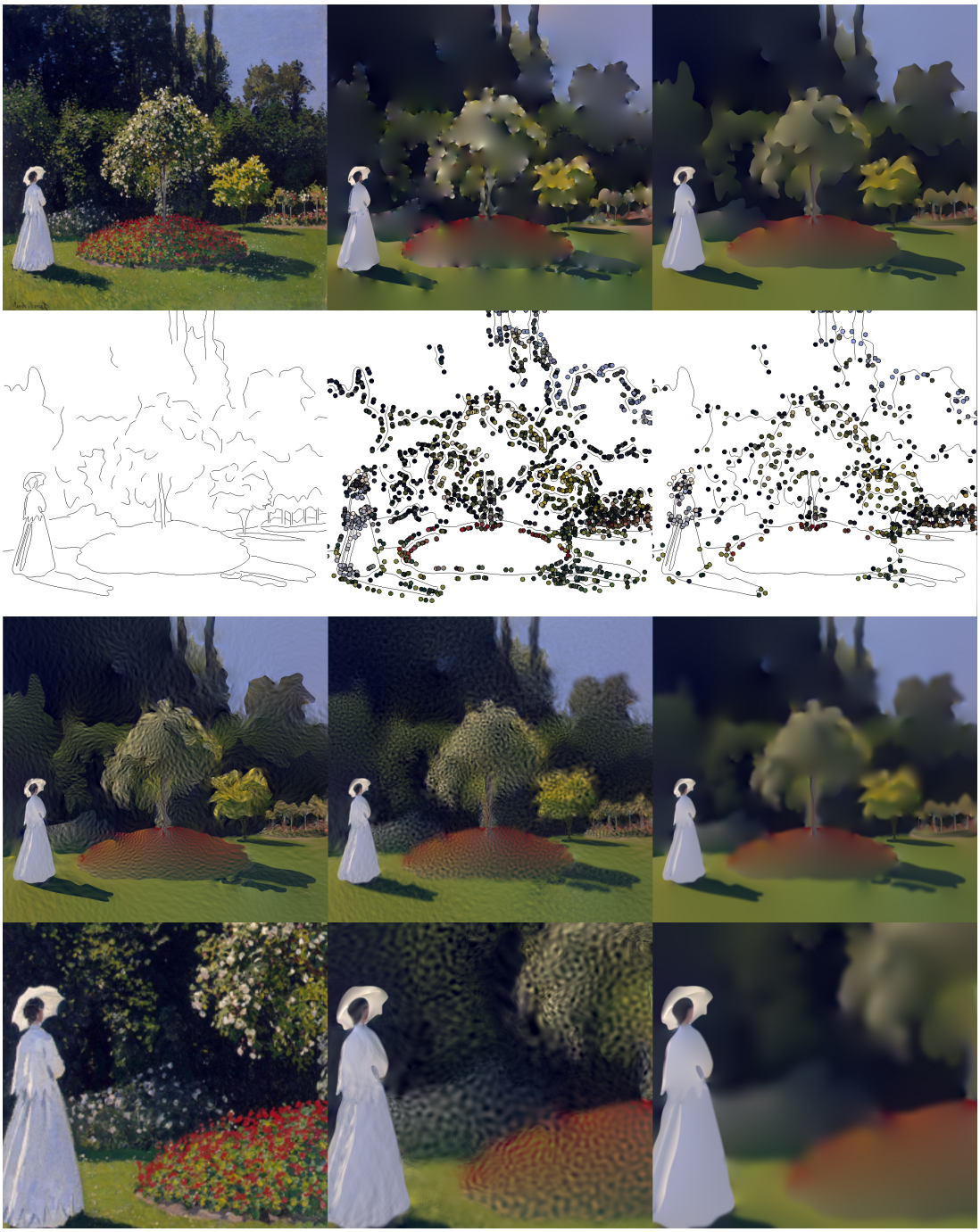

**Figure 6:** *The "Woman in the Garden" example (original image by Claude Monet). Top row (from left to right): original image, result by [Orzan et al. 08], our result. Second row (from left to right): input curves, result by [Orzan et al. 08] (1910 color points), our result (654 color points). Third row (from left to right): result of the automatic noise fitting, final result after manual editing, final result without textures (but with colors and blur edited). Bottom row (from left to right): original image closeup, final textured closeup, closeup without textures.*
*In this image the curves were quickly scribbled in the vegetation regions, similar to some of the results in [Orzan et al. 08]. One can see that Orzan et al's algorithm produces many colors due to the pronounced texture, in contrast to our algorithm. Furthermore, while the noise scale was well detected by our automatic estimation, the texture appears overly directional. Correcting this manually lets the image appear closer to the original, but the limitations of Gabor noise are still apparent.*

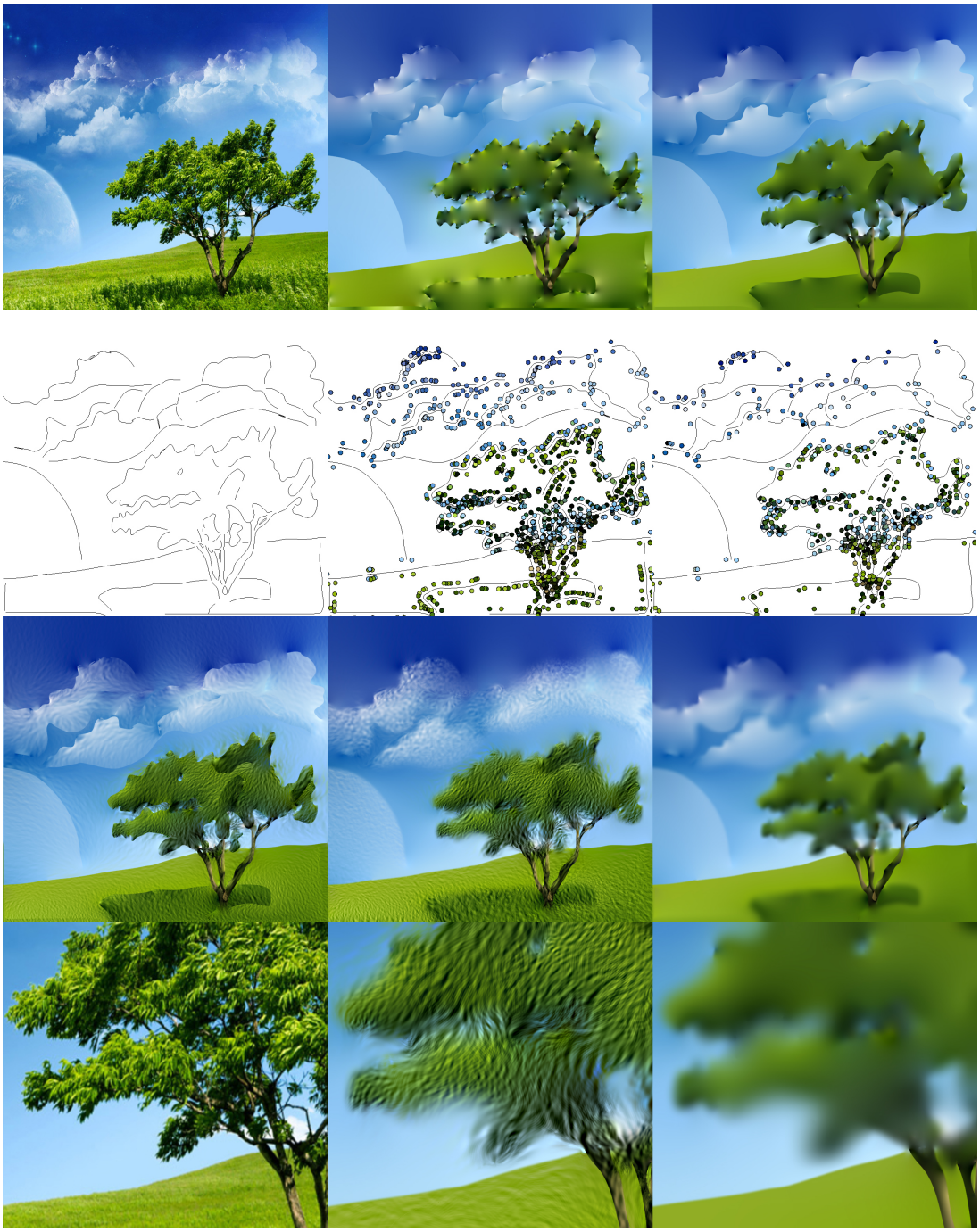

**Figure 7:** *The landscape example. Top row (from left to right): original image, result by [Orzan et al. 08], our result. Second row (from left to right): input curves, result by [Orzan et al. 08] (1401 color points), our result (664 color points). Third row (from left to right): result of the automatic noise fitting, final result after manual editing, final result without textures (but with colors and blur edited). Bottom row (from left to right): original image closeup, final textured closeup, closeup without textures. Here again the algorithm of Orzan produces many color points due to the pronounced image textures.*
*Our automatic texture fitting correctly identifies the directions of the leaves in the tree but the texture in the clouds appears too directional. In general in this example the Gabor noise produces a stylization of the input image rather than a useful reconstruction. This is especially true for the clouds who are something between texture and image. The impression of dynamic leaves moving in the wind is still conveyed, in contrast to the DCI without textures.*

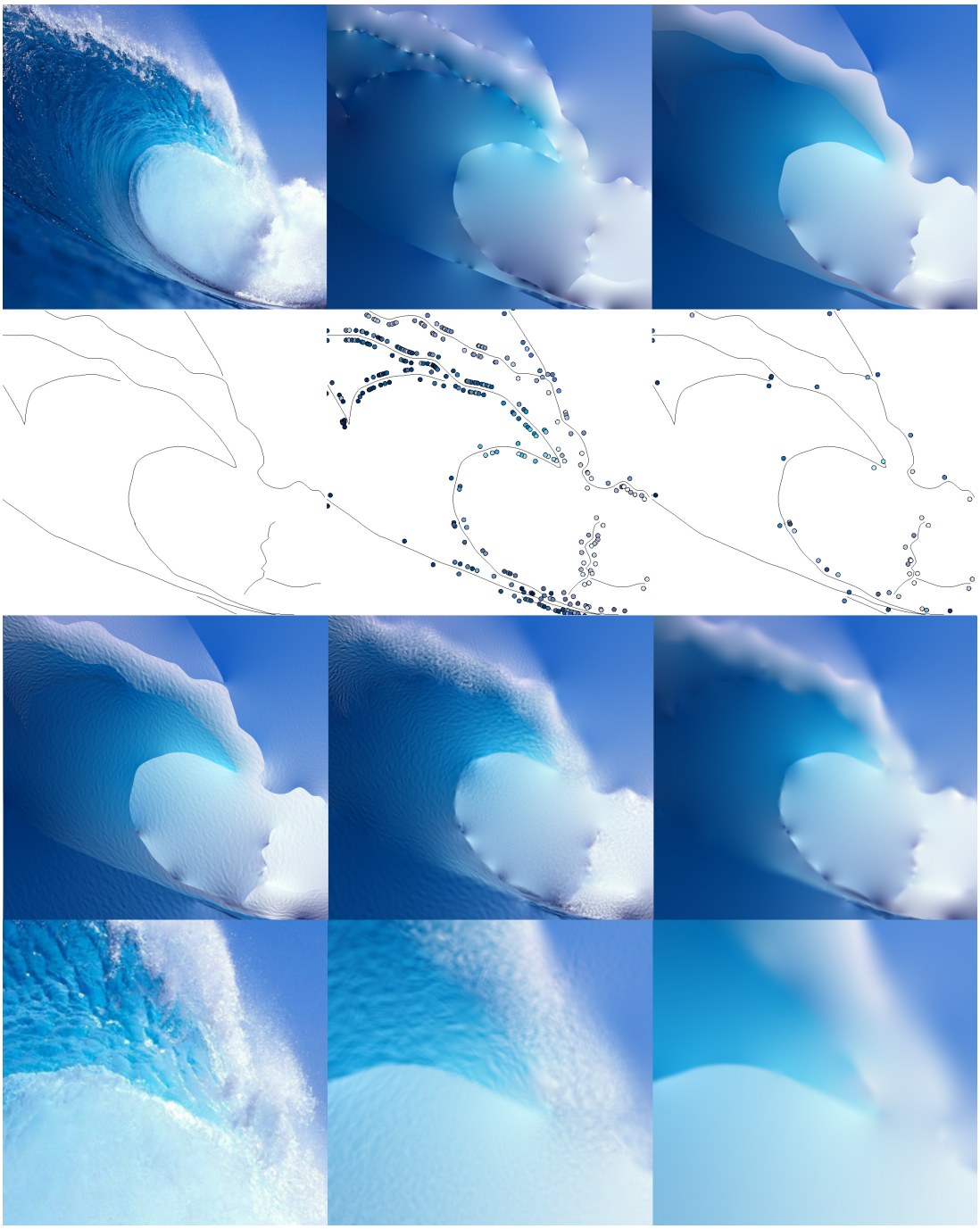

**Figure 8:** *The wave example. Top row (from left to right): original image, result by [Orzan et al. 08], our result. Second row (from left to right): input curves, result by [Orzan et al. 08] (327 color points), our result (55 color points). Third row (from left to right): result of the automatic noise fitting, final result after manual editing, final result without textures (but with colors and blur edited). Bottom row (from left to right): original image closeup, final textured closeup, closeup without textures.*
*This is an experimental example where the image was abstracted with very few strokes. Note how the Gabor noise increases the impression of water compared to the diffusion curve image without textures. This is especially the case for the closeup.*

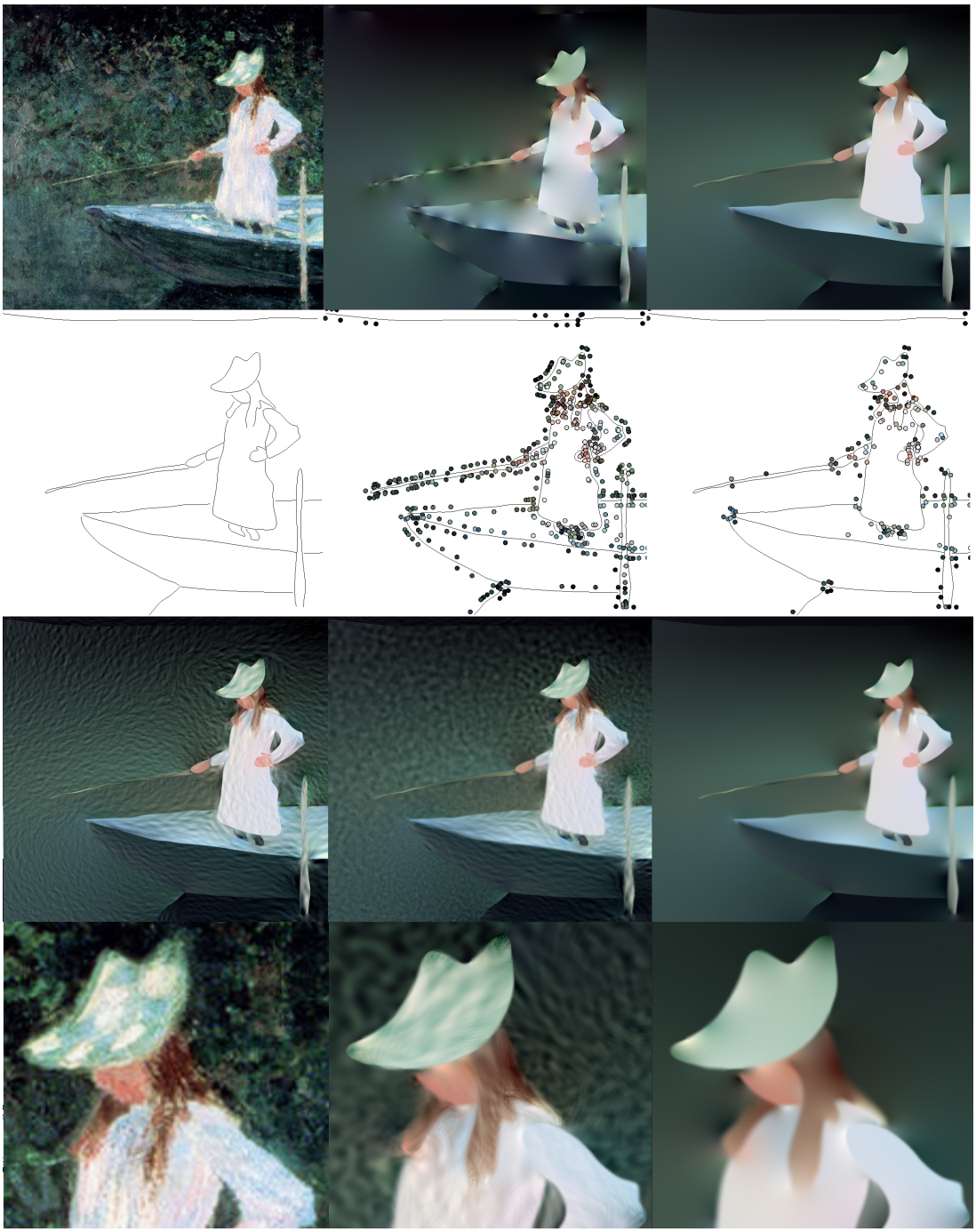

**Figure 9:** *The "In the Rowing Boat" example (original image again by Claude Monet). Top row (from left to right): original image, result by [Orzan et al. 08], our result. Second row (from left to right): input curves, result by [Orzan et al. 08] (549 color points), our result (176 color points). Third row (from left to right): result of the automatic noise fitting, final result after manual editing, final result without textures (but with colors and blur edited). Bottom row (from left to right): original image closeup, final textured closeup, closeup without textures.*
*In this example, the texture estimated by the automatic algorithm again appears too directional in the background regions.*

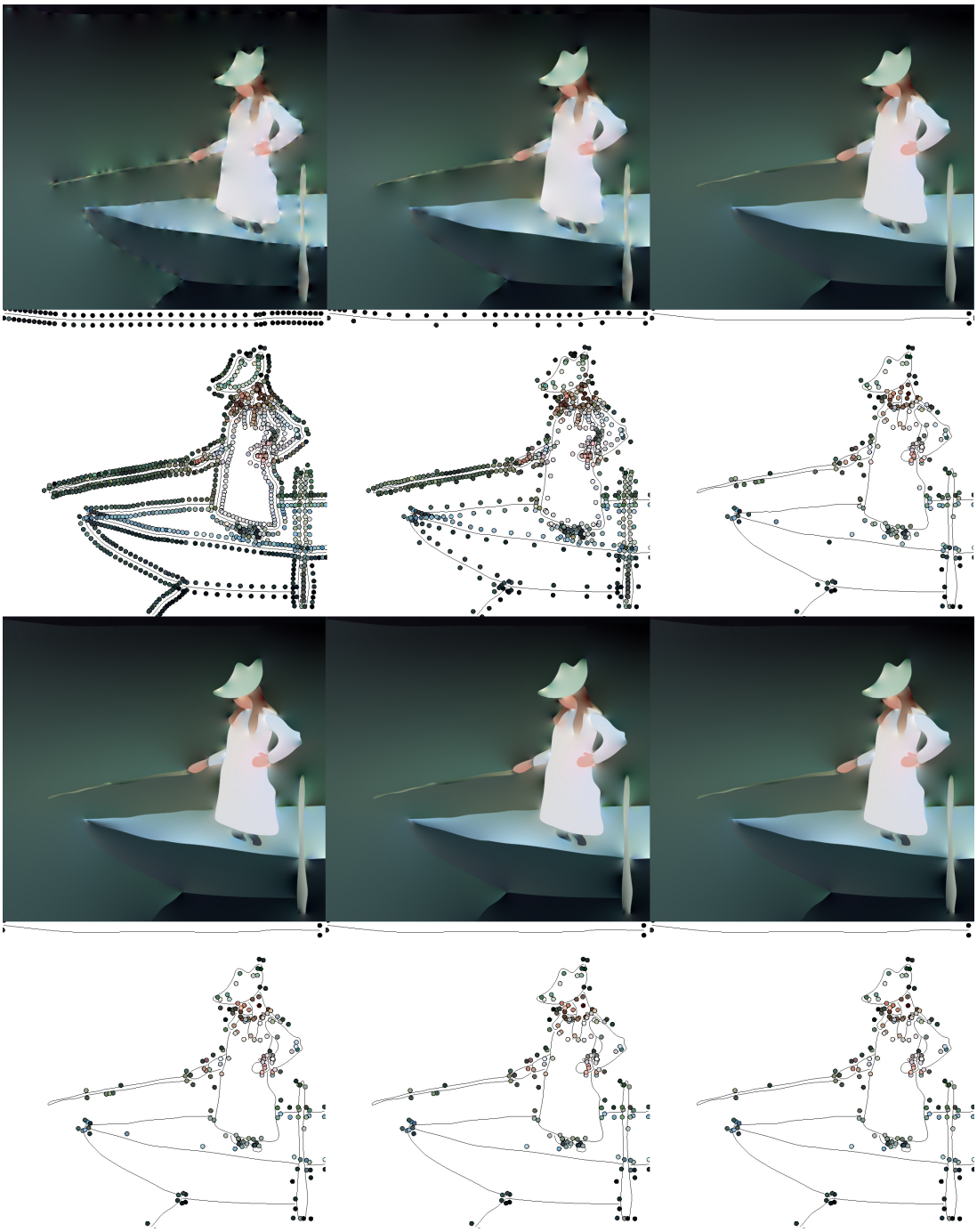

**Figure 10:** *Individual color point simplification steps for the "In the Rowing Boat" example. One row shows the result, the row below the respective individual color points. First and second row (from left to right): 1130 dense color points, 675 sparse color points, 224 simplified color points. Third and last row (from left to right): 185, 178 and 176 simplified color points. At 176 points the algorithm terminated.*

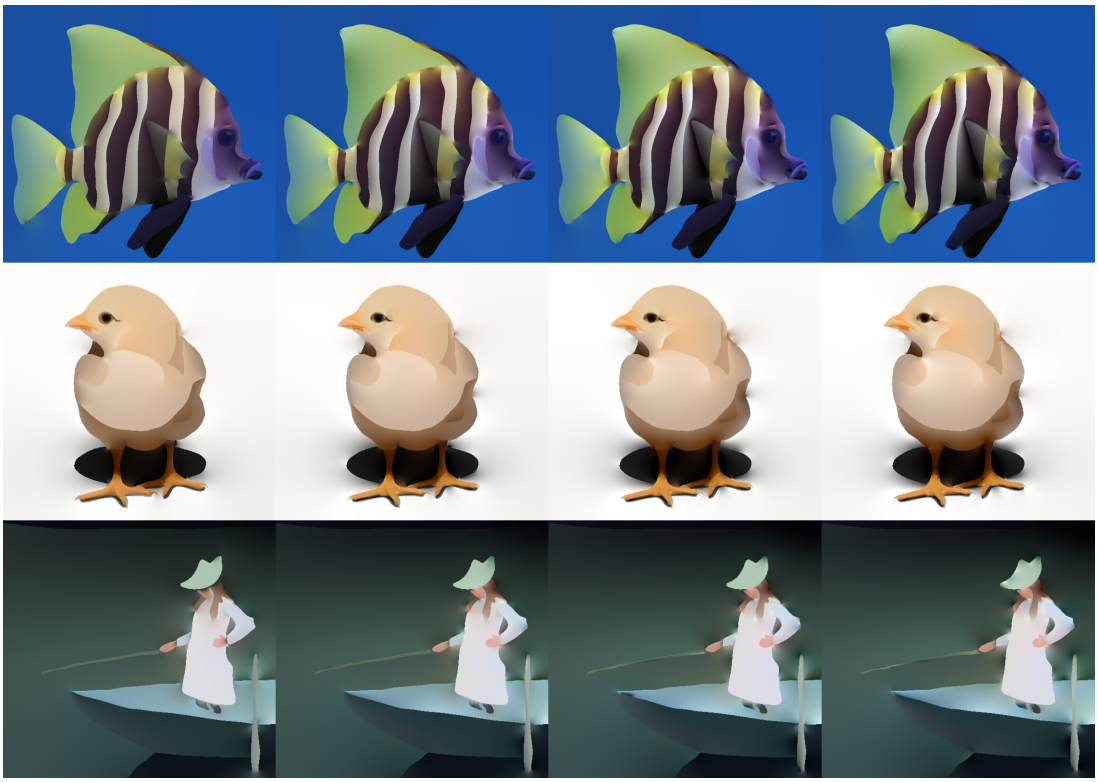

**Figure 11:** *This image compares the color fitting for a different number of colors l stored in each pixel during the diffusion process as explained in Section 3.2 in the paper. The first row shows the fish example, the second the chicken and the third the "In the Rowing Boat" example. The left column shows the fitting using a single color l per pixel. In this case, the diffusion does actually not happen as each color point represents just the average color of all nearby pixels. The number of color points for each row is 261, 205 and 116, respectively. Here the curve color fitting is fairly coarse and the image appears flat and dull. This can be observed, for example, at the head of the fish and the eye of the chicken, where details are missing. In addition, large visible errors occur as color bleeding, for example, below the hat of the child in the third row. In the second column l has been increased to two colors per pixel. The accuracy increases, leading to more color points (324, 251 and 153, respectively) and less errors. This trend continues in the third column where l has been increased to three colors per pixel. This lead to 345, 266 and 170 color points, respectively. The last column shows the results for $l = 4$, which resulted in 365, 275 and 178 color points and again a slight increase in accuracy. However, the differences to column three are very subtle, which let us conclude that four colors per pixel seem sufficient in practice.*

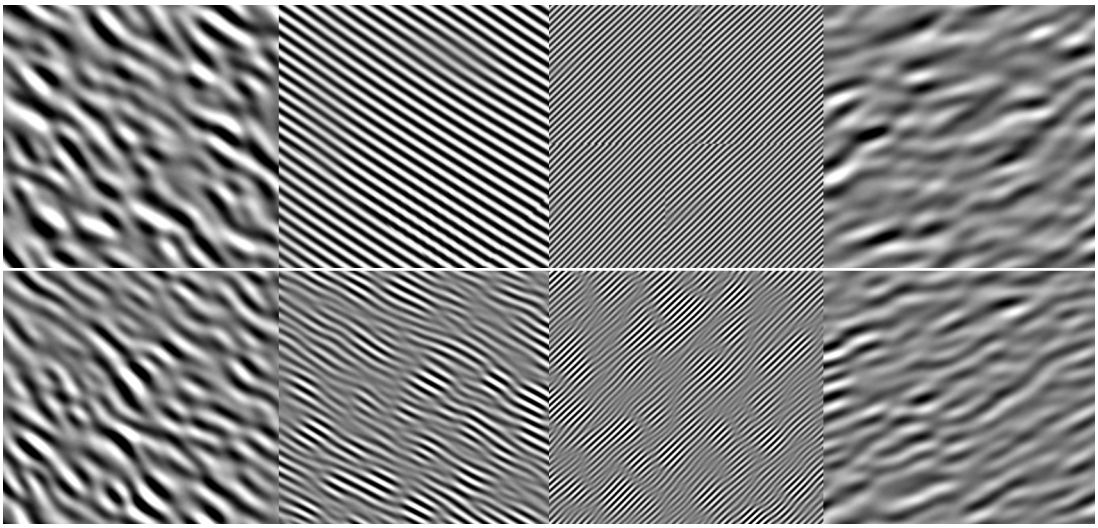

**Figure 12:** *Evaluation of the noise fitting process for different noise scales of directional Gabor noise. The top row shows four input Gabor noise images and the bottom row the results fitted by our system. Four curves tighly frame each original image (not visible here). For these examples the noise fitting works fairly well: all scales and directions are nicely reproduced.*

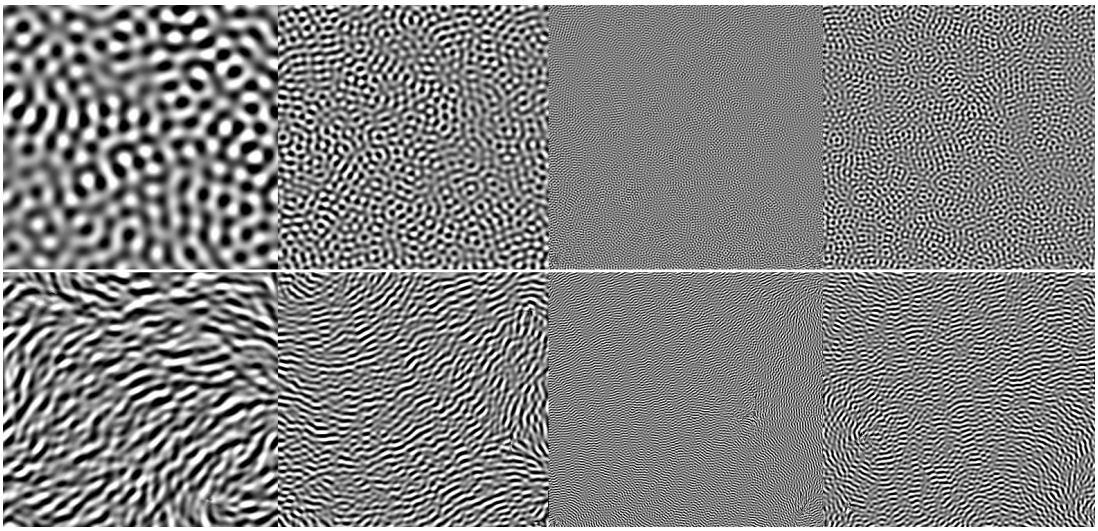

**Figure 13:** *Evaluation of the noise fitting process for different noise scales of isotropic Gabor noise. Again, the top row shows four input Gabor noise images and the bottom row the results fitted by our system. While the scale was reproduced fairly well, the fitted noise appears too directional. One possible reason for this is our usage of uniformly sized large Gabor kernels for rendering, which better reproduce directional structures.*

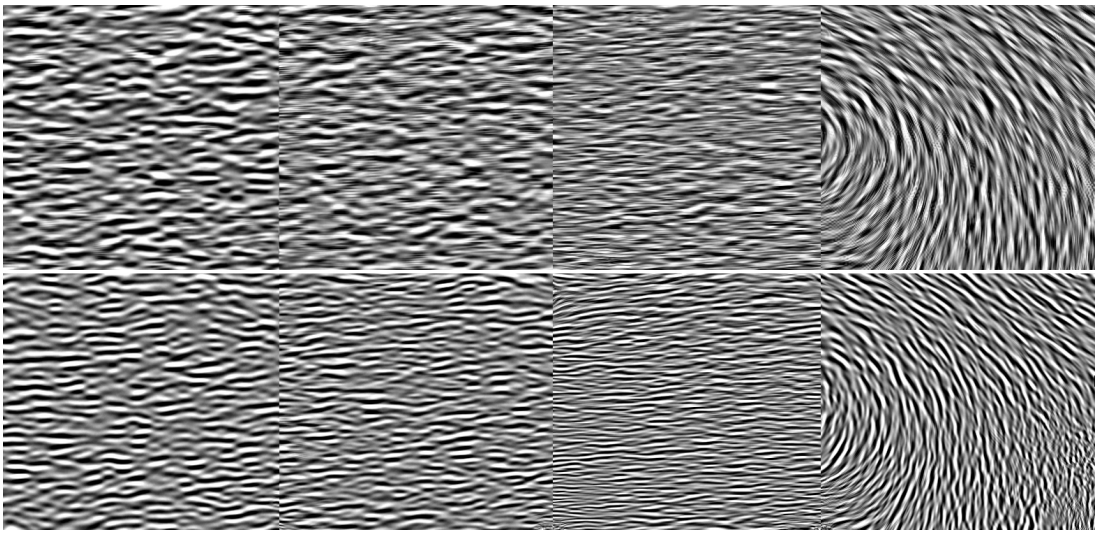

**Figure 14:** *Evaluation of the noise fitting process for noise scale variation of directional Gabor noise. Again, the top row shows four Gabor noise input images and the bottom row the results fitted by our system onto the curves framing the image. In the first tree images we gradually increased the noise scale variation in the input. Apparently, the maximum likelihood estimation seems not sensitive enought. The fitting process retains basically the avarage noise size and hardly any variation, except for the third image, where some variation is retained. A better multi-scale noise analysis might improve the fitting. In the fourth image, direction and scale is varied. While the direction is nicely reproduced, the scale variation is also underestimated in this example.*

**Figure 15:** *This image demonstrates the rendering quality depending on the number of Gabor kernels splatted. Please zoom in. The top image shows the Lena, the bottom image the chicken example. From left to right: 16, 46, 128 and 256 splats per cell. 16 splats per cell render at 16.5 frames per second, but the quality is rather low. 46 splats have been found ideal for editing as it renders with 12 frames per second and thus provides interactive feedback. It provides good preview quality. There are regions that are dominated by one frequency and direction, although the variation should be higher. This can be noticed on the shawl and hair of Lena, and the feathers of the chicken. With 128 splats the frame rate drops to 6.5 frames per second. However, the quality is increased significantly. 256 splats provide some subtle improvements but the frame rate drops to only 4 frames per second.*

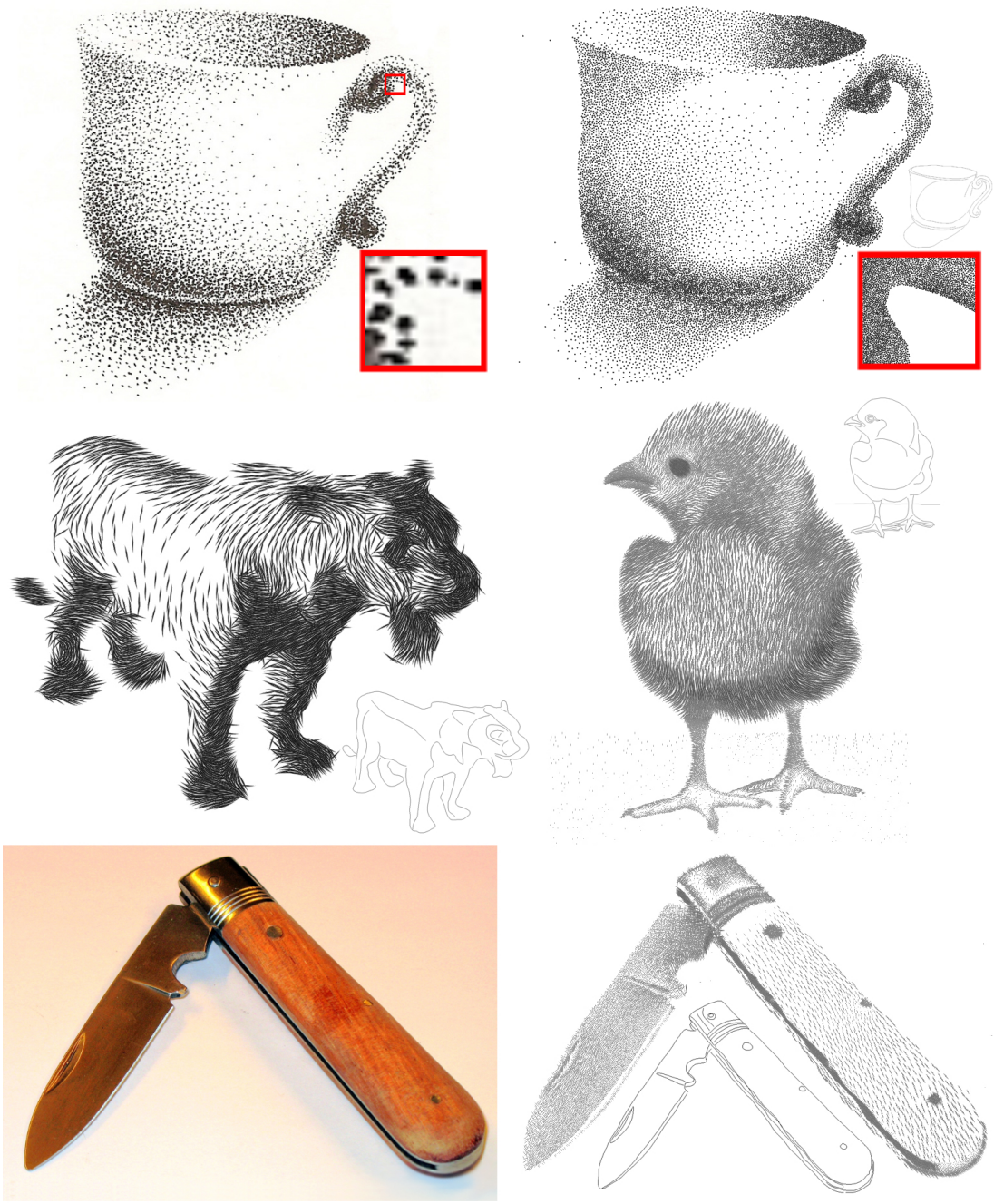

**Figure 16:** *Various stippling and hatching examples. Top row: original stippled cup (left) and stippled from diffusion curves (right). Middle row: tiger (left) and chicken (right), both hatched from diffusion curves. Bottom row: original knife (left) and hatched (right) from diffusion curves. Darker regions are cross-hatched here.*

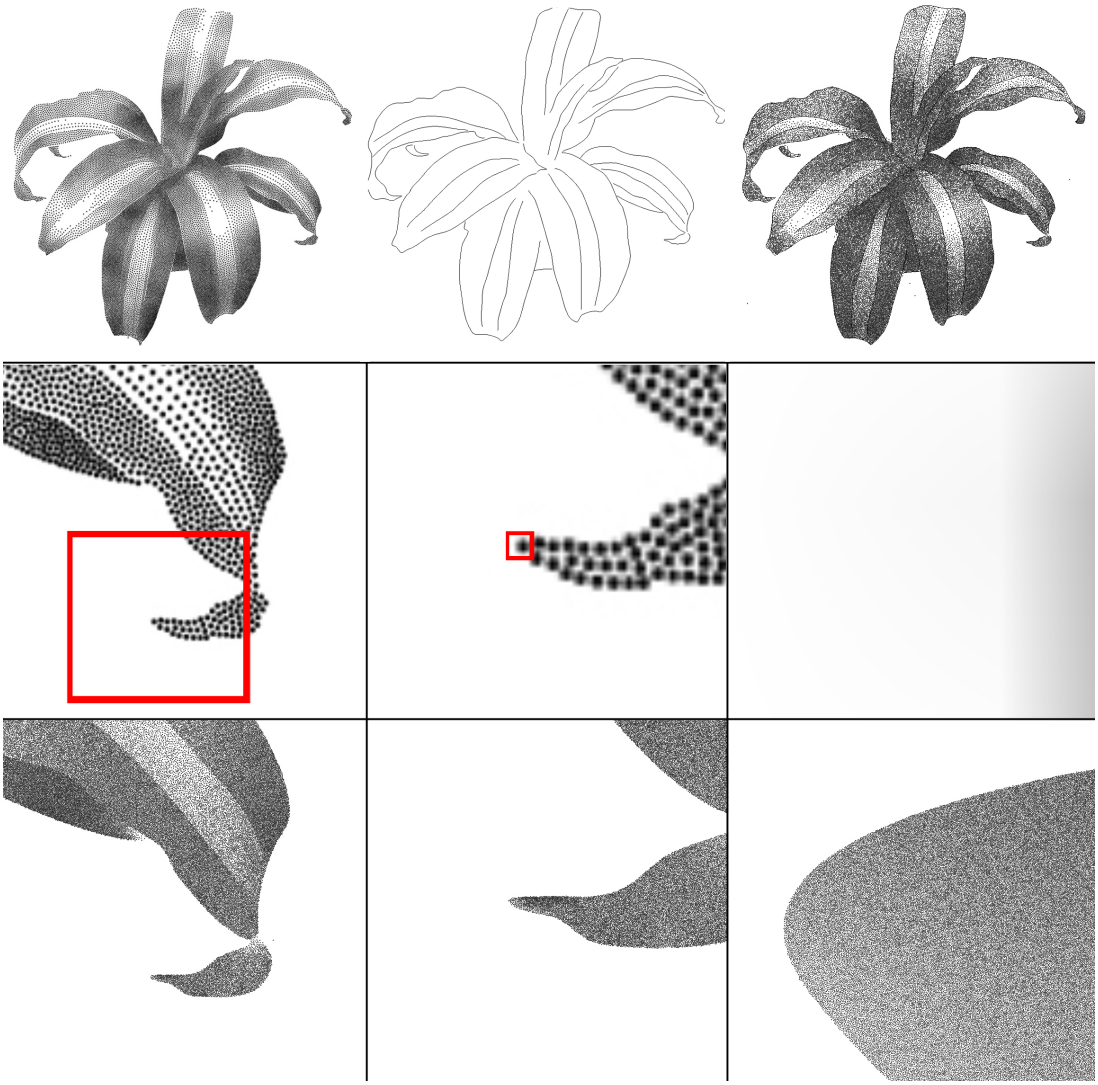

**Figure 17:** *Top row (from left to right): stippled corn input image (left), diffusion curves (middle), and the stippled image (right) with curves overlayed. The second row shows three zoom levels of the imput image, where the right image is an extreme closeup with a size of approximately one pixel. The third row shows the same zoom levels as the second row using diffusion curves and blue noise stippling with recursive Wang tiling [Kopf et al. 06]. Note how the silhouette remains sharp independly from the zoom level.*