Oil Paint Filtering using Color Palettes for Colorization

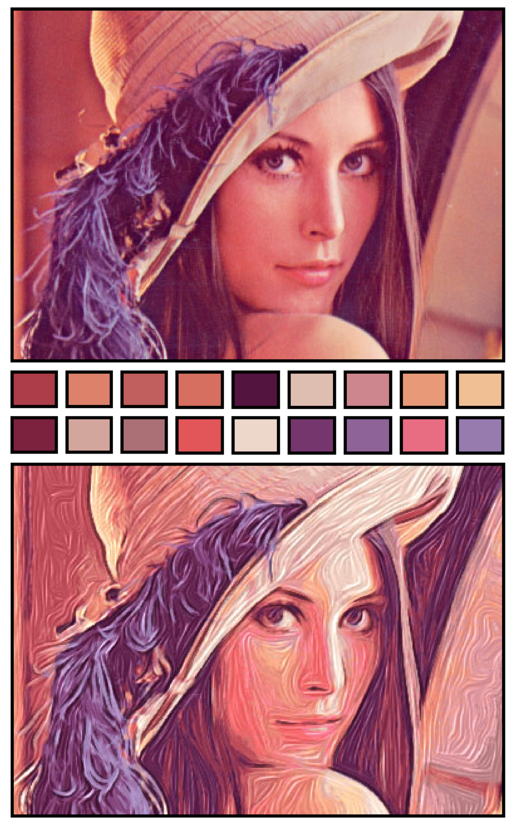
Amir Semmo Jürgen Döllner

Hasso Plattner Institute, University of Potsdam, Germany

1 Abstract

Most techniques for *image-based artistic rendering* are based on edge-preserving filtering to reduce image details without loss of salient structures, e.g., by locally weight averaging in the color domain and range, or by solving an optimization problem for image decomposition [Kyprianidis et al. 2013]. However, these approaches are typically limited in not being able to process image contents according to a pre-defined color palette, which is often a desired feature by artists to have creative control over the filtered output.

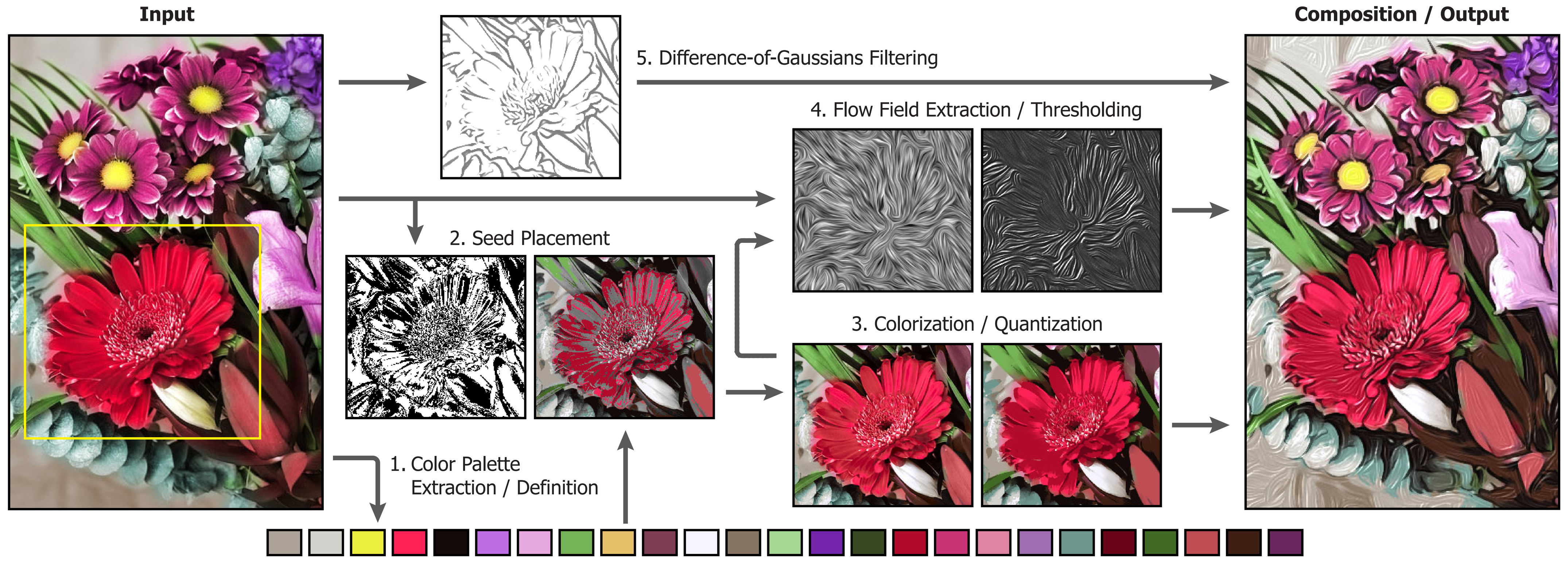
We present an inverse filtering approach that derives color palettes from an input image and uses them for feature-aware colorization, which we demonstrate by a novel oil paint effect. First, dominant colors are derived from the input image via color and entropy-based metrics. Seed pixels are then determined and propagated to the remaining pixels by adopting the optimization framework of Levin et al. [2004] to simulate the way artists paint with a reduced color palette. Finally, the quantized output is combined with flow-based computations [Kyprianidis et al. 2013] to simulate paint texture. Our technique leads to homogeneous outputs in the color domain and gives artists interactive control over the color definitions used for visualization.



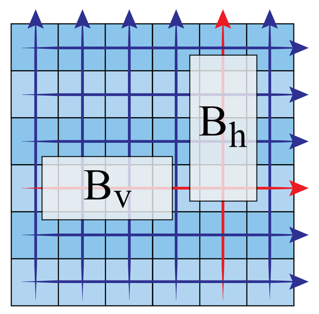
2 Related Work

Previous methods simulate brushes, media and surfaces to produce watercolor, oil and pastel [Hegde et al. 2013]. Typically, these approaches adapt colors of brush strokes by locally weight averaging in the color domain. In our approach instead, a global palette of dominant colors is derived from the input image, and colorization by optimization [Levin et al. 2004] is performed to provide color harmony. The colorization technique by Levin et al. has also been used for multi-scale image decomposition [Subr et al. 2009], but without directing the abstraction according to a global color palette. Our colors are derived from image regions with constant color tone via a scoring system, preferring a broad spectrum to enhance image contrasts. We do not simulate brush strokes per se, but combine the quantized color output with flow-based computations to simulate paint texture [Hertzmann 2002].

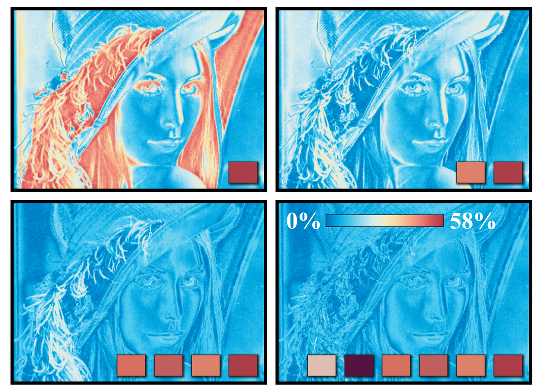
3 Method



1. Color Palette Definition – A global palette $\mathcal{C}$ is computed with colors iteratively derived from local image regions. We introduce a scoring system that weights image regions according to average color lightness L, difference ω to colors already defined in the palette, and normalized entropy η. Minimizing the score $\mathcal{S}(R)$ over all regions R yields an optimal region for a color guess (computed via averaging):



$$B = \min_{R \in I_D} \mathcal{S}(R) \text{ with } \mathcal{S}(R) = \frac{\eta(R) \cdot |R|}{\left(\sum_{p \in R} \omega(p)^2\right) \cdot \left(\sum_{p \in R} \sqrt{L(p)}\right)}$$



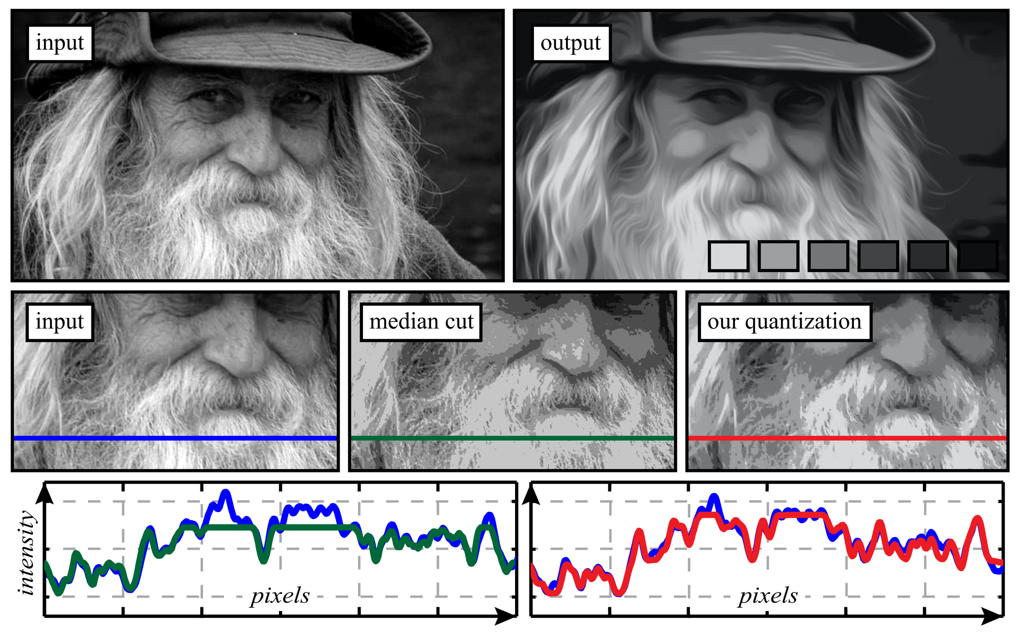
Our approach dynamically evaluates $\mathcal{S}(R)$ for regions along the vertical and horizontal scanlines of the input image. The color difference is computed in CIE-Lab (ΔE) after each iteration:

$$\omega(p) = \min_{c \in \mathcal{C}} \Delta E(p, c).$$

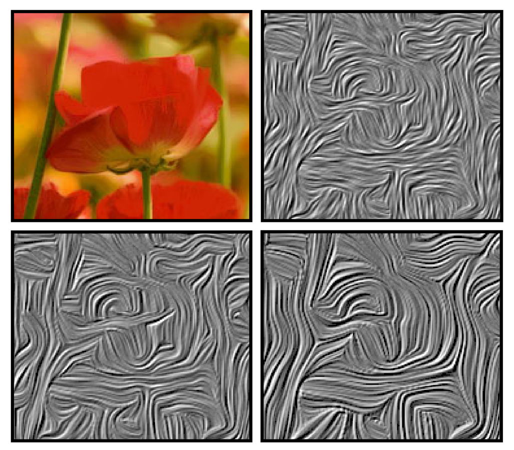
2. Placement of Color Seeds – The color difference (ΔE) between pixels of the input image and the color palette is thresholded. If it falls below a given threshold, the pixel is is marked as a seed pixel and is replaced by the dominant color. The seed pixels are used as "scribbles" for the colorization stage and simulate the initial placement of paint on a canvas.

image contours colored seed pixels blurred seed pixels

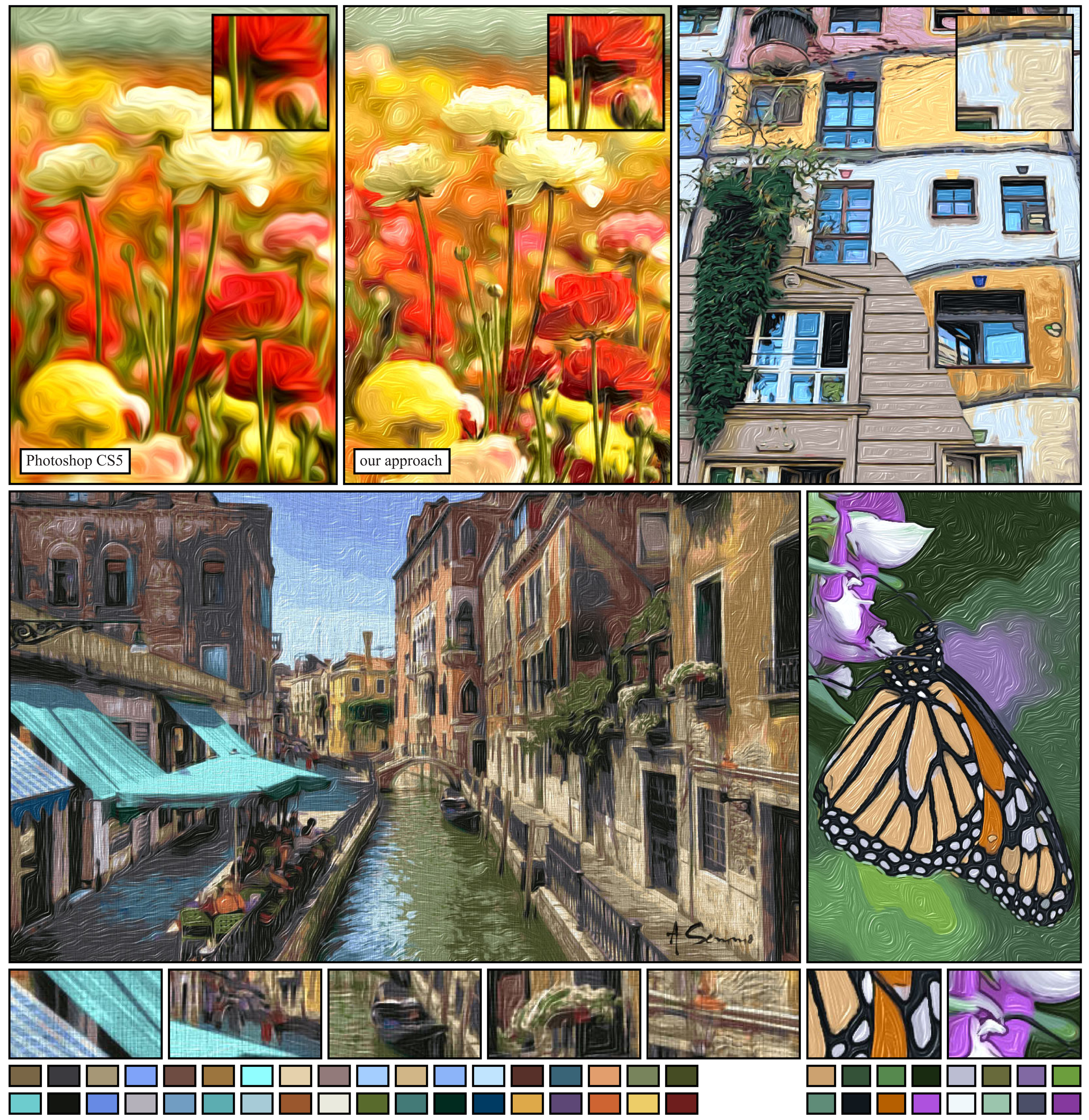
3. Colorization and Quantization – The input image is converted to CIE-Lab color space and used with the seed pixels to parameterize the technique of Levin et al. [2004]; dominant colors are propagated to the remaining pixels by solving an optimization problem, i.e., pixels with similar intensities should have similar colors. We introduce a second pass for quantization: the colorized output is converted to HSV color space, where the *hue* values are used to propagate the luminance values of the seed pixels, the result is then combined with the color components of the first pass. Contrary to quantization based on median interpolation [Heckbert 1982], our technique preserves dominant color tones better.



4. Paint Texture – Using the *smoothed structure tensor* [Kyprianidis et al. 2013], the flow field of the colorized output is thresholded by its magnitude to yield a paint texture [Hertzmann 2002]. Optional difference-of-Gaussians filtering is performed on the input image to include contour lines.



4 Results



5 Implementation and References

We have implemented the color palette extraction on the CPU using C++, and the remaining stages on the GPU using CUDA. A 800×600 pixel image is processed on an Intel® Xeon™ $4\times$ 3.06 GHz and NVidia® GTX 760 in ~ 50 seconds. The flow computation and difference-of-Gaussians filtering perform in real time for images with HD resolution, and may be interactively refined via a painting interface. The major strength of our technique is that it enables users to easily redefine color tones. We plan to build on this functionality for a user-defined color theme adjustment.

Heckbert, P. 1982. Color Image Quantization for Frame Buffer Display. *SIGGRAPH Comput. Graph. 16*, 3, 297–307.

Hegde, S., Gatzidis, C., and Tian, F. 2013. Painterly rendering techniques: a state-of-the-art review of current approaches. *Comp. Anim. Virtual Worlds 24*, 1, 43–64.

Hertzmann, A. 2002. Fast Paint Texture. In *Proc. NPAR*, 91–96.

Kyprianidis, J. E., Collomosse, J., Wang, T., and Isenberg, T. 2013. State of the 'Art': A Taxonomy of Artistic Stylization Techniques for Images and Video. *IEEE Trans. Vis. Comput. Graphics 19*, 5, 866–885.

Levin, A., Lischinski, D., and Weiss, Y. 2004. Colorization Using Optimization. *ACM Trans. Graph. 23*, 3, 689–694.

Subr, K., Soler, C., and Durand, F. 2009. Edge-preserving Multiscale Image Decomposition based on Local Extrema. *ACM Trans. Graph. 28*, 5, 147:1–147:9.

Amir Semmo, M.Sc.
amir.semmo@hpi.de
+49(0)331 5509 3909

Computer Graphics Systems Group
Hasso Plattner Institute
www.hpi3d.de

Prof. Dr. Jürgen Döllner
office-doellner@hpi.de

Hasso Plattner Institute
Prof.-Dr.-Helmert-Str. 2–3
D-14482 Potsdam, Germany

www.hpi3d.de

HPI Hasso Plattner Institut
IT Systems Engineering | Universität Potsdam

4DnD·Vis

INNOPROFILE UNTERNEHMEN REGION
Die BMBF-Innovationsinitiative Neue Länder

SPONSORED BY THE Federal Ministry of Education and Research