

# Interactive Oil Paint Filtering On Mobile Devices \*

Amir Semmo<sup>1</sup>   Matthias Trapp<sup>1</sup>   Sebastian Pasewaldt<sup>2</sup>   Jürgen Döllner<sup>1</sup>

<sup>1</sup>Hasso Plattner Institute, University of Potsdam, Germany\*

<sup>2</sup>Digital Masterpieces GmbH \*

## Abstract

*Image stylization enjoys a growing popularity on mobile devices to foster casual creativity. However, the implementation and provision of high-quality image filters for artistic rendering is still faced by the inherent limitations of mobile graphics hardware such as computing power and memory resources. This work presents a mobile implementation of a filter that transforms images into an oil paint look, thereby highlighting concepts and techniques on how to perform multi-stage nonlinear image filtering on mobile devices. The proposed implementation is based on OpenGL ES and the OpenGL ES shading language, and supports on-screen painting to interactively adjust the appearance in local image regions, e.g., to vary the level of abstraction, brush, and stroke direction. Evaluations of the implementation indicate interactive performance and results that are of similar aesthetic quality than its original desktop variant.*

Categories and Subject Descriptors (according to ACM CCS): I.4.3 [Computer Graphics]: Image Processing and Computer Vision—Enhancement—Filtering

## 1. Introduction

Image abstraction and stylization have become a natural part of a growing number of mobile apps to foster ubiquitous and casual creativity [Win13]. Today, demands for advanced image effects arise that go beyond simple operations for image processing (e.g., color grading, edge detection) to propel the creative production process. The implementation and provision of high-quality image effects on mobile devices, however, is still faced by a number of challenges such as device heterogeneity and the inherent limitations with respect to computing power and memory resources. In particular, with the continuous advancements of the camera hardware used for mobile devices (e.g., resolution, color depth, stereo) the interactive processing of image data becomes an increasingly challenging task. This especially concerns image-based artistic rendering [KCWI13] that typically requires several passes of (non-)linear filtering and highly parallelized GPU implementations for interactive processing.

This work seeks answers to these challenges by the example of interactive oil paint filtering. The primary goal was to implement a variant of the desktop prototype proposed in [SLKD16] that is able to produce aesthetic renditions of similar quality, provides a similar set of interactive features such as image warping and per-pixel parameterizations, and maintains interactive performance to be able to implement interactive tools, such as on-screen painting, that enable creative control over the visual output—which poses a contemporary field of research [Ise16].



**Figure 1:** Comparison between the results of [SLKD16] (left) and the proposed mobile implementation (right) of the oil paint filter.

The mobile implementation proposed in this work is based on a framework that is able to effectively build complex image processing chains that can be efficiently executed on mobile devices. Thereby, a number of functional and logical enhancements are made to the oil paint effect to achieve our objective, including multi-scale image processing, color grading using lookup tables [Sel04], separated filter kernels, and explicit control for shader precision. Together, the enhancements enable interactive performance for color images with full HD resolution and per-pixel parameterizations using a touch-based painting interface to locally adjust the level of abstraction. Side-by-side comparisons with the results of the desktop prototype indicate that the mobile implementation is able to produce stylized images of similar aesthetic quality (Figure 1).

\* <http://www.hpi3d.de> | <http://www.digitalmasterpieces.com>  
This is the authors' version of the work. The definitive version is available at [digilib.eg.org](http://digilib.eg.org)



**Figure 2:** Results of the multi-scale approach that processes MIP map levels and uses joint bilateral upsampling [KCLU07] with the input image (top left). Scale factors: 1.0 (top right), 0.5 (bottom left), 0.25 (bottom right). Equal filter parameters are used for all results.

## 2. Technical Approach

The reader is referred to [SLKD16] for a description of the original filtering stages. At this, the following enhancements were made:

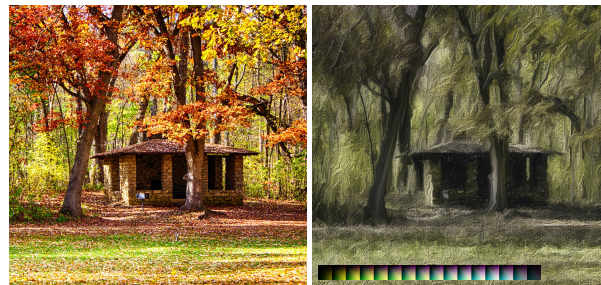
**Separated Filter Kernels.** The oil paint effect introduces wide kernels for Gaussian filtering ( $\sigma \approx 20$ ), leading to a high number of texture fetches to achieve firm color blendings. To this end, separated kernels are used to significantly reduce the number of texture fetches, which represent a performance limiting factor for mobile GPUs. Here, the flow-based implementation of [KD08] is adopted to filter in the gradient direction in the first pass and along the flow curves induced by the tangent field in the second pass.

**Multi-Scale Image Processing.** To further reduce the number of texture fetches, the abstraction in color space may be performed on the MIP map levels while using smaller filter kernels. The filtered output is then used together with the high-resolution input image to perform joint bilateral upsampling [KCLU07] (Figure 2). This may also alleviate the problem of adjusting filter kernel sizes according to the image resolution to achieve a consistent level of abstraction.

**Fast Color Transformation.** Instead of solving a global optimization problem to render images with a reduced color palette, color grading is implemented using lookup tables (LUTs) to enable real-time performance [Sel04] (Figure 3). Here, a pre-defined set of LUTs is exposed to the user to be able to select a desired color mood.

**Texture and Shader Precision.** Float-typed textures are only used if required, e.g., to encode image gradients, otherwise textures with 8-bit precision per channel are used to decrease the memory footprint. Further, medium shader precision is set for all filter stages.

**Local and Dynamic Image Processing.** A touch-based deformation technique is applied to create stylized outputs with more deliberate shape abstraction. Thereby, a virtual regular grid is projected on the image, where the user is able to locally shift grid points by brush-based painting to create effects of local compression and



**Figure 3:** Color grading based on lookup tables [Sel04].

distortion [GRG04]. The painting only affects mesh points in a certain (configurable) radius, so that only parts of an image need to be reprocessed. In addition, the framework maintains a graph of the processing chains to trigger only processing stages that directly or indirectly depend on one of the changed (e.g., painted) parameters.

## 3. Implementation

The oil paint filter was implemented using OpenGL ES and the OpenGL ES Shading Language, and was deployed on Android using a Java-based framework. All test images were processed on a OnePlus Two with a Qualcomm Snapdragon 810 and a Adreno 430 GPU. The size of the filter kernels is currently the limiting factor, while the performance scales with the image resolution. For images with full HD resolution, the filter performs at 8.5 fps (scale factor 0.25), 6 fps (scale factor 0.5), and 2.5 fps (full resolution). To this end, it is subject to interactive processing induced by global parameter adjustments or on-screen painting, e.g., to locally adjust the level of abstraction or brush. For future work, we plan to make the mobile implementation publicly available as part of an app.

## Acknowledgements

This work was partly funded by the Federal Ministry of Education and Research (BMBF), Germany, within the InnoProfile Transfer research group “4DnD-Vis” ([www.4dndvis.de](http://www.4dndvis.de)).

## References

- [GRG04] GOOCH B., REINHARD E., GOOCH A.: Human Facial Illustrations: Creation and Psychophysical Evaluation. *ACM Trans. Graph.* 23, 1 (2004), 27–44. 2
- [Ise16] ISENBERG T.: Interactive NPAR: What Type of Tools Should We Create? In *Proc. NPAR* (2016), pp. 89–96. 1
- [KCLU07] KOPF J., COHEN M. F., LISCHINSKI D., UYTENDAELE M.: Joint Bilateral Upsampling. *ACM Trans. Graph.* 26, 3 (2007). 2
- [KCWI13] KYPRIANIDIS J. E., COLLOMOSSE J., WANG T., ISENBERG T.: State of the 'Art': A Taxonomy of Artistic Stylization Techniques for Images and Video. *IEEE Trans. Vis. Comput. Graphics* 19, 5 (2013), 866–885. 1
- [KD08] KYPRIANIDIS J. E., DÖLLNER J.: Image Abstraction by Structure Adaptive Filtering. In *Proc. EG UK TPCG* (2008), pp. 51–58. 2
- [Sel04] SELAN J.: Using Lookup Tables to Accelerate Color Transformations. In *GPU Gems*. Addison-Wesley, 2004, pp. 381–392. 1, 2
- [SLKD16] SEMMO A., LIMBERGER D., KYPRIANIDIS J. E., DÖLLNER J.: Image Stylization by Interactive Oil Paint Filtering. *Computers & Graphics* 55 (2016), 157–171. 1, 2
- [Win13] WINNEMÖLLER H.: NPR in the Wild. In *Image and Video-Based Artistic Stylisation*. Springer, 2013, pp. 353–374. 1