Live Intrinsic Video

Abhimitra Meka [1] Michael Zollhöfer [1] Christian Richardt [1, 2] Christian Theobalt [1]

[1] Max Planck Institute for Informatics [2] Intel Visual Computing Institute

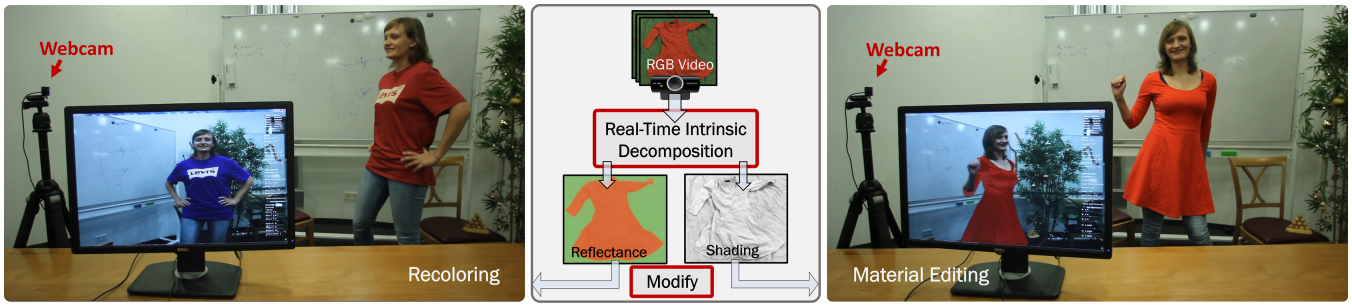


Figure 1: *We present the first approach to tackle the hard intrinsic video decomposition problem at real-time frame rates. The decomposition is the basis for live augmented video applications such as illumination-aware recoloring (left), material editing (right), retexturing and stylization.*

Abstract

Intrinsic video decomposition refers to the fundamentally ambiguous task of separating a video stream into its constituent layers, in particular reflectance and shading layers. Such a decomposition is the basis for a variety of video manipulation applications, such as realistic recoloring or retexturing of objects. We present a novel variational approach to tackle this underconstrained inverse problem at real-time frame rates, which enables on-line processing of live video footage. The problem of finding the intrinsic decomposition is formulated as a mixed variational ℓ_2–ℓ_p-optimization problem based on an objective function that is specifically tailored for fast optimization. To this end, we propose a novel combination of sophisticated local spatial and global spatio-temporal priors resulting in temporally coherent decompositions at real-time frame rates without the need for explicit correspondence search. We tackle the resulting high-dimensional, non-convex optimization problem via a novel data-parallel iteratively reweighted least squares solver that runs on commodity graphics hardware. Real-time performance is obtained by combining a local-global solution strategy with hierarchical coarse-to-fine optimization. Compelling real-time augmented reality applications, such as recoloring, material editing and retexturing, are demonstrated in a live setup. Our qualitative and quantitative evaluation shows that we obtain high-quality real-time decompositions even for challenging sequences. Our method is able to outperform state-of-the-art approaches in terms of runtime *and* result quality – even without user guidance such as scribbles.

Keywords: intrinsic decomposition, reflectance, shading, p-norm, real time, data-parallel optimization, recoloring, retexturing

Concepts: •**Computing methodologies** → **Computational photography; Mixed / augmented reality;**

SIGGRAPH 2016 Technical Paper, July 24–28, 2016, Anaheim, CA
ISBN: 978-1-4503-4279-7/16/07
DOI: http://dx.doi.org/10.1145/2897824.2925907

1 Introduction

Separating a video stream into its reflectance and shading layers is a fundamentally ambiguous and challenging inverse problem, but a solution has many potential applications. The availability of such a decomposition is for example the basis of a large variety of video editing tasks such as realistic recoloring, relighting and texture editing. Having a fast real-time solution to this fundamental problem has big ramifications – especially in the context of augmented reality – since this allows to apply such modifications, in particular photorealistic texture and appearance editing, directly to live video footage.

First, let us consider the simpler problem of computing the decomposition of a single input image. Given an image $\mathbf{I}$ (or single frame of a video), we seek a decomposition at every pixel $\mathbf{x}$, such that the product of reflectance $\mathbf{R}(\mathbf{x}) \in \mathbb{R}^3$ and shading $S(\mathbf{x}) \in \mathbb{R}$ is equal to the corresponding input observation:

$$\mathbf{I}(\mathbf{x}) = \mathbf{R}(\mathbf{x}) \times S(\mathbf{x}). \tag{1}$$

Note that the shading is modeled using the scalar quantity $S(\mathbf{x})$, based on the assumption of a white illuminant, as in previous work. Recovering the reflectance and shading image from such input constraints is ill-posed, since this problem is severely under-constrained. Equation 1 only provides three constraints for the four unknowns that define the reflectance $\mathbf{R}(\mathbf{x})$ and shading $S(\mathbf{x})$. This fundamental ambiguity is an inherent property of all *intrinsic decomposition* problems. Current state-of-the-art approaches tackle this problem by incorporating sophisticated local spatial priors that constrain the solution to a suitable subspace. These priors are based on assumptions about the typical variations encountered in reflectance and shading images. A lot of approaches [Horn 1974, Tappen et al. 2005, Gehler et al. 2011] exploit the smoothness and sparsity that is often encountered in shading and reflectance images, respectively. The reflectance sparsity assumption is especially valid for most man-made objects and scenes, since these are normally composed of a small number of materials, but both assumptions might fail if more complex natural scenes are encountered.

Decompositions of such complex natural scenes can still be obtained based on more powerful discriminative priors learned from collections of training data [Barron and Malik 2015, Zhou et al. 2015]. While these approaches handle natural scenes well, they do not easily generalize to types of scenes not contained in the training data. Similarly, multi-view decomposition approaches cope with the

complexity of natural scenes by exploiting multiple views of the same scene [Laffont et al. 2013, Duchêne et al. 2015], but these are not always available, and difficult to capture for video.

Recently, Lee et al. [2012] and Chen and Koltun [2013] proposed approaches that exploit simultaneously captured depth cues to resolve the ambiguities in the intrinsic decomposition problem. While their results are promising, depth information is often not easily available, especially for legacy video footage or for a live stream captured by a webcam that has to be processed at real-time frame rates.

Current state-of-the-art approaches for the intrinsic image [Shen et al. 2011, Gehler et al. 2011, Zhao et al. 2012, Li and Brown 2014, Bell et al. 2014, Barron and Malik 2015] or video decomposition [Bonneel et al. 2014, Ye et al. 2014, Kong et al. 2014] problem have prohibitively high runtimes of several minutes to hours per frame. This makes the scene-specific parameters of these approaches hard to tune given their slow computation times. Additionally, these approaches are restricted to slow off-line scenarios, where prerecorded data is available in advance. Therefore, it is not possible to apply these techniques in the context of live applications, such as augmented reality, that require real-time processing.

Recently, Bonneel et al. [2014] proposed the first interactive technique that decomposes a video frame in half a second. This technique is unsuitable for the decomposition of live video streams, since it requires a slow off-line pre-processing step to calculate the optical flow of the sequence. Yet, for pre-recorded data, this method offers a significant speed-up compared to previous methods. This impressive improvement in speed now allows for interactive parameter tuning, but still falls one order of magnitude short of the performance required for real-time augmented reality applications. In addition, the method relies on user-provided input in the form of scribbles, which are infeasible to provide in a real-time context.

In this paper, we propose the first approach for real-time intrinsic video decomposition. Our approach obtains temporally coherent decompositions at real-time frame rates without the need for explicit correspondence search. We tackle the resulting variational optimization problem using a specifically tailored data-parallel optimization strategy. High-quality decompositions are obtained even for challenging real-world video sequences at the capturing rate of the input device, without requiring any user input. Our main contributions are as follows:

- The first real-time algorithm to decompose live video streams into high-quality reflectance and shading layers.
- A novel formulation for the intrinsic video decomposition problem that combines local spatial and global spatio-temporal priors tailored to produce high-quality and temporally consistent video decompositions in real time.
- A new data-parallel solver for mixed ℓ_2–ℓ_p-optimization problems based on iteratively reweighted least squares (IRLS).

Our approach does not require user scribbles, unlike many state-of-the-art off-line approaches, yet it achieves comparable and even better results. The possibilities opened up by our live intrinsic video decomposition are demonstrated by several live video editing applications, including material editing, recoloring, retexturing and stylization.

2 Related Work

We constrain our discussion of related work to intrinsic decomposition methods [Barrow and Tenenbaum 1978] computing reflectance and shading layers. Many intrinsic image decomposition techniques were proposed in the past, but only very few video techniques exist that master the additional difficulty of ensuring temporally coherent results. Our approach is the first to run at real-time frame rates.

Retinex and Local Priors Land and McCann [1971] suggested the Retinex approach that locally classifies edges of a grayscale image into shading or reflectance edges based on the assumption that stronger edges correspond to reflectance and weaker to shading variation. Many variants of similar and derived local edge cues have since been used [Jiang et al. 2010], for instance with learned edge classifiers [Bell and Freeman 2001, Tappen et al. 2005]. Retinex assumptions are also often part of more complex non-local methods. Bonneel et al. [2014] decompose edges into their contributing reflectance and shading components instead of simply labeling them. They use local chromaticity cues to guide the separation, and enforce sparsity on reflectance edges and smoothness on illumination edges using a hybrid ℓ_2–ℓ_p-optimization strategy. We use similar local terms, but perform the decomposition directly on image colors instead of gradients, which avoids the integration of the gradient-domain reflectance and shading images. More recently, Bi et al. [2015] use a similar energy, with local color differences in *Lab*-space used to inversely weigh the local sparsity term for reflectance estimation. Methods based only on such local cues produce decent results on simple scenes with a single segmented object, as shown in Grosse et al.'s survey [2009], but produce inaccurate results on many real-world images, as they only coarsely model the physics of image formation and ignore the global structure of the scene. None of the above approaches runs in real time.

Global Priors Retinex-based methods have been extended to include non-local cues to improve the decomposition across an entire image [Gehler et al. 2011, Shen and Yeo 2011]. Shen et al. [2008] and Zhao et al. [2012] show promising results for decomposing structured texture patterns by enforcing constant reflectance for pixels with similar local texture, but the non-local search is computationally expensive. Chang et al. [2014] present a probabilistic model for intrinsic decomposition. Other non-local methods enforce a small number of reflectance surfaces in the scene by clustering the reflectance image [Garces et al. 2012, Bi et al. 2015]. Such complex clustering strategies are very time consuming and not real-time capable. Our approach includes non-local cues in a real-time capable way using a histogram-based clustering approach. Zoran et al. [2015] propose a framework to infer mid-level visual properties and apply it to the intrinsic decomposition task. Other computationally expensive global cues include creating pairwise pixel correspondences across the entire image [Chen and Koltun 2013, Bell et al. 2014]. We propose similar correspondence constraints, which are real-time capable, through a non-local sampling strategy. In combination with our local sparsity term for reflectance, we are able to achieve globally *and* temporally coherent decompositions.

Statistical and Learning-Based Techniques Statistics of real-world geometry and illumination can be learned or modeled to help resolve the inherent ambiguity in intrinsic decomposition [Barron and Malik 2015]. Such approaches are powerful, but often reach their limit on more complex scenes that fall outside of the used training data. Discriminative techniques have also been used to solve the Retinex problem by classifying edges as either a reflectance or shading edge [Bell and Freeman 2001, Tappen et al. 2005]. Recently, Zhou et al. [2015] learned the relative reflectance ordering of image patches from a large annotated dataset to identify surfaces of similar reflectance under different illumination conditions. In spite of such diverse strategies, intrinsic decomposition remains a challenging, ill-posed problem, especially on real-world scenes. Many recent approaches thus resort to user input like scribbles to resolve ambiguities [Bousseau et al. 2009, Shen et al. 2011, Bonneel et al. 2014, Ye et al. 2014]. Even without such user interaction, our approach produces decomposition results, in real-time, that are on par with or even better than results obtained with previous off-line approaches.

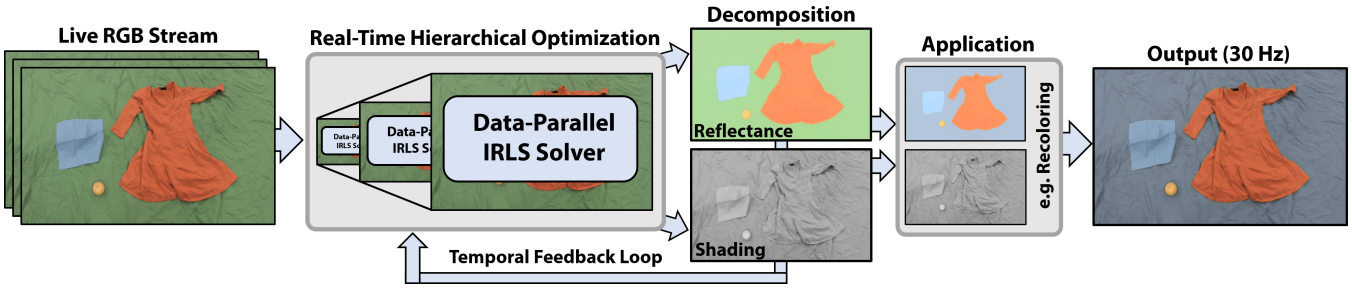


Figure 2: *Overview of our proposed real-time intrinsic decomposition approach.*

Multi-Image and Depth-Based Techniques The highly under-constrained intrinsic decomposition problem benefits from additional information, such as per-pixel depth, temporal information from time lapses, or geometry from multi-view images. Several techniques rely on varying illumination over an image sequence of a static scene, to isolate the temporally constant reflectance from time-varying illumination effects [Weiss 2001, Matsushita et al. 2004, Laffont et al. 2012, Hauagge et al. 2013, Laffont and Bazin 2015]. Geometry cues computed from multi-view imagery are often exploited to construct further priors. Kong et al. [2014] use sequences captured with a moving light source, and use optical flow to find temporal correspondences in dynamic scenes. Surface normals are then used to improve the decompositions. Such approaches break down when lighting is near-constant, as in many real-life scenarios. Laffont et al. [2013] and Duchêne et al. [2015] use multi-view stereo to reconstruct scene geometry and hence estimate environment maps of the scene. Depth information has proven very useful in estimating reflectance and shading, especially under a Lambertian reflectance assumption. Given an RGB-D video stream, illumination estimation and shape-from-shading refinement is feasible in real time [Wu et al. 2014]. Depth information has also been exploited to impose local and global constraints on the shading layer [Lee et al. 2012, Barron and Malik 2013, Chen and Koltun 2013, Hachama et al. 2015], for example by exploiting local normal information. Although depth and other geometric cues are very valuable, they require specific multi-view capture, moving light sources or special camera hardware – all of which are not available for live RGB video. We propose the first approach for real-time, space-time coherent intrinsic decomposition from just a single monocular RGB video.

Intrinsic Video Decomposition Techniques Most discussed techniques are limited to decomposing a single image off-line and yield unacceptable, temporally incoherent results when directly applied to video. Only few approaches explicitly tackle video. Shen et al. [2014] perform intrinsic decomposition only for specific regions in the video, their approach requires user input and has a slow off-line runtime. Ye et al. [2014] propose a multi-pass optimization strategy for intrinsic video decomposition that clusters reflectance pixels and uses optical flow for correspondence across frames. Their approach is fundamentally off-line as it takes more than a minute per video frame. Bonneel et al. [2015] use the temporal regularity of the input video as a guide to stabilize the shading and albedo layers computed by intrinsic decomposition techniques. Bonneel et al. [2014] suggest a fast and flexible method that uses both local and global chromaticity cues. However, since the method operates on grayscale images instead of RGB, the output reflectance image has the same chromaticity as the input image, which is often wrong. Therefore, the approach notably struggles if the assumptions of white light and Lambertian surfaces are violated. In contrast, our method works in the RGB space and is more resilient against violation of these assumptions. The method of Bonneel et al. [2014] requires half a second per frame and an additional slow off-line preprocessing step to calculate optical flow. In contrast, our approach runs completely in real time. We extend recent concepts for real-time non-linear optimization on the GPU [Wu et al. 2014, Zollhöfer et al. 2014, 2015]. In particular, we propose a novel GPU-based optimizer to explicitly handle ℓ_2–ℓ_p-optimization. Previous video techniques also use extensive user input, whereas we obtain similar or even better results in real time without any user interaction.

3 Overview

Given an arbitrary video stream as input, our proposed live intrinsic video decomposition technique extracts the corresponding shading and reflectance streams at real-time rates. Like previous decomposition methods, we assume Lambertian reflectance in the scene, i.e. the reflectance is equal to the albedo of the surface. Figure 2 shows an overview of all building blocks of our approach. We propose a novel mixed ℓ_2–ℓ_p-formulation (see Section 4) for the intrinsic video decomposition problem that leads to decompositions that are both spatially and temporally coherent without the need for an explicit correspondence search. The resulting high-dimensional and non-convex variational optimization problem is robustly and efficiently optimized using a custom-tailored, fully data-parallel, iteratively reweighted least squares (IRLS) solver (see Section 5). Leveraging the computational power of modern graphics hardware, we can compute decompositions at frame rate. The obtained results (see Section 6) show that our approach outperforms the current state of the art qualitatively and quantitatively in terms of accuracy, robustness and runtime performance. We show the real-time capabilities of the proposed approach in a live setup that demonstrates a variety of compelling demo applications (see Section 7), ranging from recoloring to material editing tasks. Finally, we discuss current theoretical and technical limitations (Section 8) and conclude with an outlook (Section 9).

4 Intrinsic Video Decomposition

Intrinsic decomposition problems are commonly tackled by transferring and solving them in the log-domain [e.g. Shen and Yeo 2011]:

$$\mathbf{i}(\mathbf{x}) = \mathbf{r}(\mathbf{x}) + s(\mathbf{x}), \tag{2}$$

where lower-case letters are the log-domain versions of their upper-case counterparts. This explicitly linearizes the constraints and facilitates the use of simpler optimization strategies. Even in the log-domain, the intrinsic decomposition problem is still under-constrained, since all per-pixel decompositions are completely independent. Most existing intrinsic video decomposition techniques rely on user scribbles to provide crucial constraints for solving the heavily under-constrained intrinsic decomposition problem. However, user scribbles are not an option for on-line intrinsic video decomposition approaches, such as ours, as such user input cannot be provided at 30 Hz in a live-streaming setup. We extend previously used reflectance, shading and chromaticity priors to suit our real-time setting. In addition, we propose new global space-time and

reflectance clustering priors designed with real-time computational performance in mind, to solve the under-constrained decomposition problem. Our approach is based on the decomposition energy

$$E(\mathcal{D}) = \sum_{\mathbf{x}} \Big[E_{\text{data}}(\mathbf{x}) + E_{\text{priors}}(\mathbf{x}) + E_{\text{non-local}}(\mathbf{x}) + E_{\text{clustering}}(\mathbf{x}) \Big]. \quad (3)$$

All sub-energies are defined per pixel $\mathbf{x}$. We minimize this energy for every video frame to obtain the decomposition

$$\mathcal{D} = \big[\ldots, \mathbf{r}(\mathbf{x})^\top, \ldots, s(\mathbf{x}), \ldots \big]^\top \quad (4)$$

that stacks the unknown per-pixel reflectance and shading values defined by the vector-valued (RGB) reflectance layer $\mathbf{r}$ and the scalar shading layer s. All unknowns are defined in the log-domain. We assume the image formation model in Equation 2 for defining the decomposition problem. Next, we discuss the particular data terms and prior constraints used in our novel decomposition energy and describe how we efficiently solve the resulting mixed ℓ_2–ℓ_p-optimization problem at real-time rates. To this end, we propose a specifically tailored data-parallel solution strategy in Section 5.

4.1 Data Fitting Term

The output of our optimization is a decomposition of the input video frame (in log-space) into a sum of reflectance and shading components. We enforce this as a soft-constraint via the data fitting term E_{data}. Similar to most previous intrinsic decomposition methods, we assume monochromatic, white illumination; therefore the shading image is scalar-valued. In the log-domain, we enforce the fitting constraint per color channel, i.e. $i_c \approx r_c + s$ for $c \in \{\text{R}, \text{G}, \text{B}\}$. To make the solution more robust to deviations from perfectly white illumination, we apply per-channel perceptual weights ω_c to obtain the final constraint:

$$E_{\text{data}}(\mathbf{x}) = w_{\text{data}} \cdot \omega_{\text{iw}}(\mathbf{x}) \cdot \sum_{c \in \{\text{R,G,B}\}} \omega_c \cdot |i_c(\mathbf{x}) - r_c(\mathbf{x}) - s(\mathbf{x})|^2, \quad (5)$$

where $\{\omega_{\text{R}}, \omega_{\text{G}}, \omega_{\text{B}}\} = \{0.299, 0.587, 0.114\}$ (ITU-R BT.601). In addition, our data term is scaled by the data term weight w_{data}, and the image intensity weight

$$\omega_{\text{iw}}(\mathbf{x}) = 1 - w_{\text{intensity}} \cdot (1 - |\mathbf{I}(\mathbf{x})|), \quad (6)$$

which expresses the empirically confirmed observation that pixels with a higher intensity $|\mathbf{I}(\mathbf{x})|$ provide more reliable decomposition constraints, while low-intensity pixels need to be more strongly regularized to better deal with noise in the input data. In particular for commodity webcams, which have a low signal-to-noise ratio, low intensity pixels need strong regularization. This is adjustable via $w_{\text{intensity}}$.

4.2 Local Prior Terms

We assume that illumination effects such as shading and shadows only affect the intensity of a pixel, but not its chromaticity, $\mathbf{c}(\mathbf{x}) = \mathbf{I}(\mathbf{x}) / |\mathbf{I}(\mathbf{x})|$. Therefore, any large gradient in the chromaticity does not originate in the shading image, but in the reflectance image. This can be interpreted as an intensity-normalized version of Retinex [Land and McCann 1971]. Based on a chromaticity similarity weight $\omega_{\text{cs}}(\mathbf{x}, \mathbf{y})$, we selectively scale the reflectance and shading priors, which are described next, to compute an optimal decomposition:

$$\omega_{\text{cs}}(\mathbf{x}, \mathbf{y}) = \exp\big(-\alpha_{\text{cs}} \cdot \|\mathbf{c}(\mathbf{x}) - \mathbf{c}(\mathbf{y})\|_2\big). \quad (7)$$

Here, we use the empirically determined factor $\alpha_{\text{cs}} = 15$ as it yields the best decomposition results in our experiments. In contrast to Bonneel et al. [2014], we use a smooth discriminator function instead of a hard threshold on chromaticity difference.

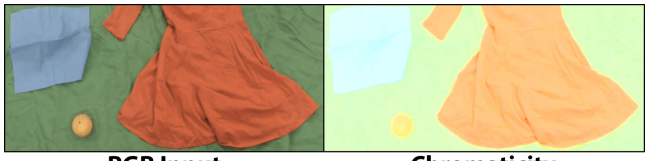


Figure 3: *Chromaticity shift: in practical conditions, especially in dark regions (e.g. folds of the dress), chromaticity changes occur due to indirect illumination effects and finite camera sensitivity.*

Reflectance Sparsity We assume that the reflectance image $\mathbf{r}$ consists of piecewise-constant regions. Such a sparse solution can be obtained by minimizing the p^{th} power of the ℓ_p-norm, with $p \in [0, 2)$, of the local per-pixel reflectance gradients $\nabla\mathbf{r}(\mathbf{x})$. Smaller choices of p yield sparser decompositions. We set $p = 0.8$ in all our experiments. However, as $\mathbf{r}$ is a 3-vector, $\nabla\mathbf{r}$ is a 3×2 matrix, consisting of horizontal and vertical gradients for each color channel. To ensure soft and edge-friendly piecewise constancy of the reflectance image, we do not minimize the ℓ_p-matrix norm directly, but instead separate the gradients along each dimension and minimize their magnitude independently:

$$E_{\text{reflectance}}(\mathbf{x}) = w_{\text{reflectance}} \cdot \sum_{\mathbf{y} \in N(\mathbf{x})} \omega_{\text{cs}}(\mathbf{x}, \mathbf{y}) \cdot \|\mathbf{r}(\mathbf{x}) - \mathbf{r}(\mathbf{y})\|_2^p. \quad (8)$$

Here, $N(\mathbf{x})$ is the 4-pixel neighborhood of pixel $\mathbf{x}$, and the more similar two pixels' chromaticities, as measured by $\omega_{\text{cs}}(\mathbf{x}, \mathbf{y})$, the lower the weight on the reflectance difference. The whole objective is scaled by $w_{\text{reflectance}}$. Note that we express this constraint directly on color values, not on gradients [Bonneel et al. 2014], which benefits real-time performance (see Section 6.5).

Shading Smoothness For purely diffuse surfaces, shading is only a function of the shape of the object. Since objects in natural scenes generally have smooth shapes, we expect the shading image to also be smooth. In addition, neighboring pixels with different chromaticities, as measured by $1 - \omega_{\text{cs}}(\mathbf{x}, \mathbf{y})$, indicate a reflectance edge, where shading smoothness should be more strongly enforced:

$$E_{\text{shading}}(\mathbf{x}) = w_{\text{shading}} \cdot \sum_{\mathbf{y} \in N(\mathbf{x})} (1 - \omega_{\text{cs}}(\mathbf{x}, \mathbf{y})) \cdot |s(\mathbf{x}) - s(\mathbf{y})|^2. \quad (9)$$

Here, w_{shading} is the weight of this prior constraint.

Chromaticity Prior As mentioned earlier, we assume that the chromaticity of the input image is not altered by illumination effects such as shading and shadows. In this case, the chromaticity of the unknown reflectance image $\mathbf{r}$ should be the same as that of the input image. We enforce this using the soft constraint

$$E_{\text{chromaticity}}(\mathbf{x}) = w_{\text{chromaticity}} \cdot \|\mathbf{c}(\mathbf{x}) - \mathbf{c}_{\text{r}}(\mathbf{x})\|_2^2, \quad (10)$$

where $\mathbf{c}$ is the chromaticity of the input video frame, and $\mathbf{c}_{\text{r}}$ is the chromaticity of the reflectance image $\mathbf{r}$.

In a simplified image formation model that only considers direct white illumination and infinite camera precision, chromaticity changes solely occur due to reflectance changes. However, in the real world (especially in low-intensity regions), indirect illumination effects and the camera's finite sensitivity limit this assumption. This leads to shifts in the captured chromaticities (see Figure 3). In brighter regions, the chromaticity is still a good approximation of the reflectance. Therefore, we combine the three priors using the image intensity weight $\omega_{\text{iw}}(\mathbf{x})$, to reduce the influence of the shading and chromaticity priors for dark pixels, to obtain

$$E_{\text{priors}}(\mathbf{x}) = E_{\text{reflectance}}(\mathbf{x}) + \omega_{\text{iw}}(\mathbf{x}) \cdot \big[E_{\text{shading}}(\mathbf{x}) + E_{\text{chromaticity}}(\mathbf{x}) \big]. \quad (11)$$

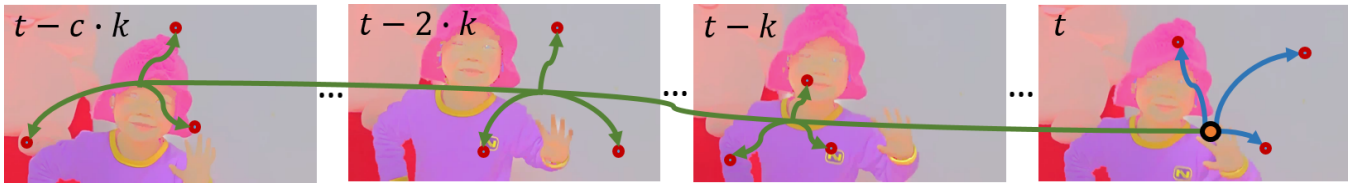


Figure 4: *Spatio-temporal reflectance consistency prior: we apply global consistency constraints in the space (blue) and time (green) domains based on random sampling. If sampled pixels have similar chromaticity, we constrain their reflectances to also be similar.*

4.3 Spatio-Temporal Reflectance Consistency Prior

Many natural and man-made scenes contain multiple, identically colored instances of an object, such as cushions on a sofa. Illumination is also changing over time, causing pixels to increase or decrease in brightness. In these scenarios, it is essential to ensure spatio-temporally consistent reflectances. This is not handled by the constraints described so far, which merely locally enforce piecewise constant reflectance. To ensure spatially and temporally consistent reflectance, we propose a new global, sampling-based, spatio-temporal reflectance consistency constraint, that does not rely on costly space-time correspondence finding, such as optical flow. This allows for real-time performance.

For each pixel $\mathbf{x}$ in the reflectance image, we connect it to N_s randomly sampled pixels $\mathbf{y}_i$. Samples are chosen from reflectance images of the current and previous frames t_i, as illustrated in Figure 4. If the chromaticity of the current pixel is reasonably close to that of the sampled pixel, we constrain their reflectances to be similar:

$$E_\text{non-local}(\mathbf{x}) = w_\text{non-local} \cdot \sum_{i=1}^{N_\text{s}} g_i(\mathbf{x}) \cdot \|\mathbf{r}(\mathbf{x}) - \mathbf{r}_{t_i}(\mathbf{y}_i)\|_2^2 \qquad (12)$$

$$g_i(\mathbf{x}) = \begin{cases} \omega_\text{iw}(\mathbf{x}) & \text{if } \|\mathbf{c}(\mathbf{x}) - \mathbf{c}_{t_i}(\mathbf{y}_i)\|_2 < \tau_\text{cc}, \\ 0 & \text{otherwise.} \end{cases} \qquad (13)$$

Here, τ_cc is a chromaticity consistency threshold. We randomly sample $N_\text{s} = 9$ pixel locations from the current frame t as well as the previous five keyframes (spaced five frames apart). Since darker pixels suffer from shifted chromaticities, we again reduce their contribution based on ω_iw.

The proposed approach, although relying on random sampling, is especially effective when combined with the reflectance sparsity prior. It is very likely that distinct regions of same reflectance are connected by at least a few samples, and the reflectance sparsity prior then spreads the global reflectance consistency constraints to other nearby pixel locations. By creating connections to previous video frames, this term leads to temporally stable decompositions. The number and spacing of the used frames is adjustable: a shorter temporal window may for example be preferable in case of fast motion or illumination changes. Spacing the frames further apart makes our approach more resilient to slow illumination changes. We use a default of five past keyframes spaced five frames apart which proved sufficient for all our test sequences. Note that in contrast to previous work [Bonneel et al. 2014, Kong et al. 2014, Ye et al. 2014], we do not require time-consuming explicit correspondence finding to obtain temporally coherent results.

4.4 Reflectance Clustering Prior

The reflectance sparsity and non-local consistency priors lead us very close to the goal of a sparse distribution of reflectances, by encouraging piecewise constancy and consistent colors for disjoint objects of the same reflectance, respectively. However, there may still be remaining inconsistencies in actually uniform reflectance regions and unwanted temporal changes within the same material. We therefore introduce a per-pixel soft constraint for global reflectance consistency that ensures the reflectance image to be close to the desired result and temporally stable, even without costly spatial correspondence finding. We achieve this by estimating a clustered version of the reflectance image. We first compute a histogram of the reflectance image and find major reflectance clusters. Each pixel's reflectance is then constrained to match the reflectance of its most similar cluster. Specifically, we compute an RGB histogram of the reflectance image with 30^3 uniformly spaced bins, where each bin stores the number of pixels within it, as well as their mean color (see Figure 5). We exponentially average histograms over time to improve the temporal coherence of the reflectance clusters, which we compute by performing weighted k-means clustering on the reflectance histogram. The cluster centers are initialized with the previous frame's clusters, which speeds up convergence, or randomly in the case of the first frame. We also collapse duplicate reflectance clusters with chromaticity differences below the chromaticity consistency threshold τ_cc used before.

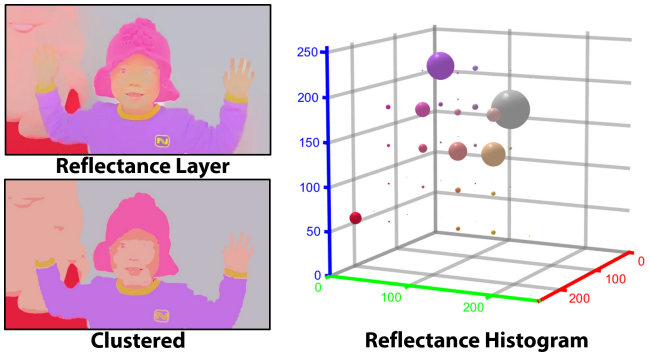


Figure 5: *Reflectance Clustering: The reflectance layer is clustered based on a weighted k-means strategy on the reflectance histogram.*

We then create a clustered reflectance image $\mathbf{r}_\text{cluster}$ using the closest reflectance cluster for each reflectance pixel $\mathbf{r}(\mathbf{x})$ in terms of ℓ_2 distance. This clustered reflectance image could be used directly as the final reflectance image, but any errors in the clustering process would become part of the final result. Instead, we use the clustered reflectance image as a soft constraint that is most strongly applied to dark pixels as these are most unreliable. The reason for this is what we call *chromaticity shift*: large shading variations may cause a shift in chromaticity in the darker regions of the same reflectance surface because of inter-reflections and finite camera sensitivity. We resolve this issue by constraining dark pixels more strongly to be similar to their closest reflectance cluster:

$$E_\text{clustering}(\mathbf{x}) = \omega_\text{clustering}(\mathbf{x}) \cdot \|\mathbf{r}(\mathbf{x}) - \mathbf{r}_\text{cluster}(\mathbf{x})\|_2^2, \qquad (14)$$

$$\omega_\text{clustering}(\mathbf{x}) = w_\text{clustering} \cdot \exp\left(-\alpha_\text{clustering} \cdot |\mathbf{I}(\mathbf{x})|\right), \qquad (15)$$

using the clustering prior weight $w_\text{clustering}$ and empirically determined soft function constant $\alpha_\text{clustering} = 0.4$. Using the clustered

reflectance image to define the decomposition energy is a chicken-and-egg problem, as estimating the clustered image requires the reflectance to be available, whereas estimating the reflectance requires the clustering. To solve this problem, we exploit our coarse-to-fine optimization strategy (see Section 5.4). We perform the clustering on the reflectance estimated on the second-finest level and use it for regularizing the finest level result.

5 Real-Time Optimization

The intrinsic decomposition objective $E(\mathcal{D})\colon \mathbb{R}^{4N} \rightarrow \mathbb{R}$ proposed in Equation 3 is a mixed ℓ_2–ℓ_p-optimization problem in the unknown parameter values $\mathcal{D}$. Here, $N = W \times H$ is the resolution of the input video stream. The parameter vector $\mathcal{D}$ holds the $4N$ unknown pixel values that fully define the intrinsic decomposition, i.e. the per-pixel log-space reflectance $\mathbf{r}(\mathbf{x}) \in \mathbb{R}^3$ and shading $s(\mathbf{x}) \in \mathbb{R}$. The optimal decomposition $\mathcal{D}^*$ is the minimizer of $E(\mathcal{D})$:

$$\mathcal{D}^* = \underset{\mathcal{D}}{\operatorname{argmin}}\, E(\mathcal{D}). \tag{16}$$

This high-dimensional, under-constrained optimization problem is non-linear and non-convex due to the involved ℓ_p-optimization. In addition, this optimization has a large number of unknowns even for small video resolutions, e.g. about 2 million unknowns for a resolution of 800×600 pixels, which have to be optimized under our tight real-time constraint of 30 Hz. Previously, sparse gradient priors [Levin and Weiss 2007, Levin et al. 2007, Joshi et al. 2009, Bonneel et al. 2014] have been tackled on the CPU using an *iteratively reweighted least squares* (IRLS) approach; but not at real-time rates given millions of unknowns. We exploit the computational horsepower of the data-parallel GPU architecture to solve such variational optimization problems at framerate. In contrast to previous work on data-parallel optimization [Wu et al. 2014, Zollhöfer et al. 2014, 2015], which only deals with standard non-linear least squares formulations, we propose a novel solution strategy for general unconstrained ℓ_p-optimization problems. To this end, we devise a custom-tailored data-parallel IRLS solver that allows to solve for up to 2 million unknowns at real-time rates.

5.1 Data-Parallel IRLS Core Solver

IRLS is a widely used optimization strategy [Holland and Welsch 1977]; its key idea is to transform a general unconstrained optimization problem to a sequence of reweighted subproblems:

$$\left\{\mathcal{D}^{(k)} = \underset{\mathcal{D}}{\operatorname{argmin}}\, E^{(k)}\big(\mathcal{D} \mid \mathcal{D}^{(k-1)}\big)\right\}_{k=1}^{K}. \tag{17}$$

The original energy E is successively reweighted based on the previous solution $\mathcal{D}^{(k-1)}$ to obtain new energies $E^{(k)}$. Starting from an initial estimate $\mathcal{D}^{(0)}$, the optimum $\mathcal{D}^* = \mathcal{D}^{(K)}$ of E is found based on K such steps. For the first time, we integrate the IRLS strategy into a data-parallel iterative GPU solver for handling the ℓ_p term in our energy. As a starting point, let us consider a single scalar ℓ_p-residual of the objective. Since we use the p^{th} power of ℓ_p in our energy, it can be written as:

$$\big|r(\mathcal{D}^{(k)})\big|^p. \tag{18}$$

Here, $r(\mathcal{D}^{(k)}) \in \mathbb{R}$ is a general scalar and linear residual. Now let $\mathcal{D}^{(k-1)}$ be the approximate solution computed in the previous iteration step. Then, a suitable reweighting scheme is obtained by approximately splitting Equation 18 into two components:

$$\big|r(\mathcal{D}^{(k)})\big|^p \approx \underbrace{\big|r(\mathcal{D}^{(k-1)})\big|^{p-2}}_{c(\mathcal{D}^{(k-1)})} \cdot \big|r(\mathcal{D}^{(k)})\big|^2 \tag{19}$$

$$= \Big(\sqrt{c\big(\mathcal{D}^{(k-1)}\big)} \cdot r(\mathcal{D}^{(k)})\Big)^2. \tag{20}$$

This factorization is based on the assumption that parameters change slowly $\mathcal{D}^{(k)} \approx \mathcal{D}^{(k-1)}$. The reweighting factor $c\big(\mathcal{D}^{(k-1)}\big)$ is constant during one iteration step, since it only depends on the previous solution. The remaining second factor is a quadratic function of the parameters since the residuals $r(\mathcal{D}^{(k)})$ are linear. Note, reweighting also applies to the case $p = 2$, resulting in $c\big(\mathcal{D}^{(k-1)}\big) = 1$. Thus, we can write the energy $E^{(k)}$ using reweighting factors $c_k(\mathcal{D}^{(k-1)})$:

$$E^{(k)}(\mathcal{D} \mid \mathcal{D}^{(k-1)}) = \sum_{k=1}^{M} \Big(\underbrace{\sqrt{c_k(\mathcal{D}^{(k-1)})} \cdot r_k(\mathcal{D})}_{\hat{r}_k(\mathcal{D} \mid \mathcal{D}^{(k-1)})}\Big)^2. \tag{21}$$

The total number $M = N(13 + N_s)$ of residuals $\hat{r}_k(\mathcal{D} \mid \mathcal{D}^{(k-1)})$ depends on the data fitting term ($3N$ terms), shading smoothness prior (N terms), reflectance sparsity prior ($3N$ terms), chromaticity prior ($3N$ terms), spatio-temporal reflectance coherence prior (NN_s terms) and the reflectance clustering prior ($3N$ terms). To simplify notation further, we stack all M scalar residual terms $\hat{r}_k(\mathcal{D} \mid \mathcal{D}^{(k-1)})$ in a single vector:

$$F^{(k)}(\mathcal{D} \mid \mathcal{D}^{(k-1)}) = \big[\hat{r}_1(\mathcal{D} \mid \mathcal{D}^{(k-1)}), \ldots, \hat{r}_M(\mathcal{D} \mid \mathcal{D}^{(k-1)})\big]^\top. \tag{22}$$

This vector can be interpreted as a high-dimensional vector field $F : \mathbb{R}^N \rightarrow \mathbb{R}^M$ that allows to rewrite $E^{(k)}(\mathcal{D})$:

$$E^{(k)}(\mathcal{D} \mid \mathcal{D}^{(k-1)}) = \Big\|F^{(k)}(\mathcal{D} \mid \mathcal{D}^{(k-1)})\Big\|^2. \tag{23}$$

Since all elements of $F^{(k)}$ are linear functions of the unknowns, the resulting optimization problem is quadratic, hence convex:

$$\mathcal{D}^{(k)} = \underset{\mathcal{D}}{\operatorname{argmin}} \Big\|F^{(k)}\big(\mathcal{D} \mid \mathcal{D}^{(k-1)}\big)\Big\|_2^2. \tag{24}$$

We find the global optimum of the sequential sub-problems by setting the partial derivatives to zero. The resulting highly over-constrained linear system ($M \gg N$) is solved in the least-squares sense. Previous work [Weber et al. 2013, Wu et al. 2014, Zollhöfer et al. 2014, 2015] demonstrated the feasibility of data-parallel preconditioned conjugate gradient (PCG) for the fast solution of such problems. We use a similar GPU-based PCG approach to exploit the sparsity pattern of the system matrix. Entries of the system matrix are computed on the fly (and only if they are required) during PCG iterations, and are never explicitly stored. As preconditioner, we employ inverse diagonal preconditioning. The proposed strategy is highly efficient and already provides real-time performance for a moderate amount of unknowns. However, since our objective has millions of unknown parameters, real-time optimization is not directly feasible with the proposed core solver. To alleviate this problem, we propose a local-global optimization approach that exploits the regular grid structure of the image domain to partition the problem into small local sub-problems. Each small sub-problem can then be solved efficiently in shared GPU memory based on the presented core solver.

5.2 Local–Global Optimization Approach

Instead of solving the global joint optimization problem directly, we subdivide the domain into small square subdomains and locally

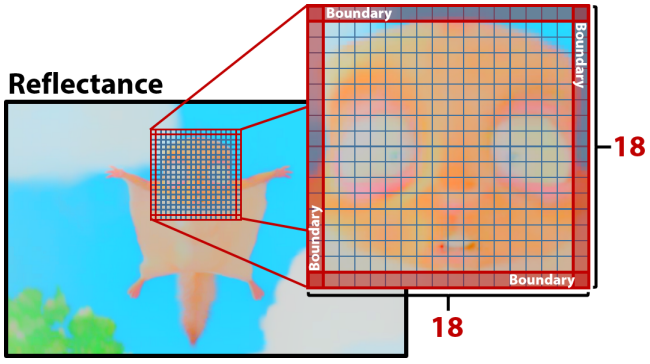


Figure 6: *Subdomains of our local–global optimization approach.*

perform the optimization on each of these. Afterwards, the updates obtained in this local step are exchanged, and the whole procedure is iterated. For a start, let us consider the energy without the global reflectance consistency constraint. We describe a strategy to incorporate this energy term later in Section 5.3. The evaluation of all other objectives requires locally at most a one-ring pixel neighborhood. We solve each sub-problem independently by one thread block on the GPU and aim to keep the complete state of the solver close to the associated multiprocessor, i.e. in shared memory and registers.

In each subdomain, we first cache the input data and current decomposition to shared memory. In this step, we include a one-ring boundary. We enforce Neumann constraints on this boundary to decouple the sub-problems. The size of the local subdomains is set based on the available L1 cache on the used GPU. We use 16×16 subdomains, see Figure 6. Including the boundary pixels, this leads to overlapping 18×18 regions that are loaded to shared memory. The local per-domain problem is solved via the proposed IRLS strategy. After solving the local problems, the subdomain decomposition result is written back to global memory to facilitate data exchange between regions. For the 16×16 inner subregions, one thread per pixel writes the obtained new shading and reflectance values to global memory. Values on the boundary are not written back, as they are part of the inner subregion of an adjacent subdomain. This can be interpreted as a variant of the *Schwarz Alternating Procedure* [Zhao 1996] for domain decomposition problems. Note that in our implementation, IRLS steps and Schwarz iterations are directly interleaved. We write to global memory out-of-place, leading to deterministic results (fully *Additive Schwarz*), which are independent of GPU scheduling. This is in contrast to Wu et al. [2014] and Zollhöfer et al. [2015], where a blend between an *Additive* and *Multiplicative* strategy has been proposed. We found that our approach leads to temporally more coherent results if only a fixed limited number of iterations is performed. Sub-domains are shifted virtually after each iteration step based on a Halton sequence to improve convergence.

5.3 Adding the Spatio-Temporal Reflectance Prior

Up to now, we did not consider the spatio-temporal reflectance prior in the optimization strategy. This energy term does not directly fit the proposed local–global sub-domain optimization strategy due to its global nature, since sample points are randomly distributed in the video volume. This introduces a coupling between the local subproblems. Note that the optimization strategy proposed by Wu et al. [2014] and Zollhöfer et al. [2015] can not handle this situation. We follow a two-fold strategy do deal with this problem. First, we treat these connections similar to the boundary by imposing Neumann constraints for values outside of the processed sub-domain. This allows to cache these values dynamically to registers before the local sub-domain optimization commences. Second, we assume unidirectionality of the constraints, i.e. only the reflectance value at the currently processed pixel $\mathbf{r}(\mathbf{x})$ in Equation 12 is an unknown and the target $\mathbf{r}_{t_i}(\mathbf{y}_i)$ is assumed to be constant. Informally speaking, pixels only see their drawn samples, but do not know if they have been sampled by others. Therefore, the partial derivatives do not depend on the target, and a constant amount of N_s values per thread has to be cached. We keep these values in registers. Cached values are updated over the non-linear IRLS iterations. This decouples the sub-domain systems from each other and allows for a data-parallel optimization as proposed earlier.

5.4 Nested Hierarchical Optimization

For the solution strategy proposed so far, error reduction stalls after the high-frequency error components have been resolved. Low frequency errors are only slowly resolved, since the propagation of updates over long spatial distances requires many iteration steps. This is a common problem of all iterative solution strategies. To alleviate this problem, we run the proposed iterative local–global optimization approach in a nested coarse-to-fine loop based on a Gaussian pyramid. Since low frequency errors are of higher frequency on the coarser resolution levels, all frequency components of the error can be efficiently handled, hence leading to fast convergence. We solve the optimization on every level and use a prolongation operator to obtain a suitable starting value for the next finer level. Prolongation is based on bi-linear interpolation of pixel data. Currently, we use a hierarchy with three to four levels depending on the input resolution. This turned out to be sufficient for good convergence rates. On the coarsest level, we perform a frame-by-frame initialization based on the assumption that reflectance and shading have the same magnitude. Therefore, we set $\mathbf{r}(\mathbf{x}) = \mathbf{i}(\mathbf{x})/2$ and $s(x) = |\mathbf{i}(\mathbf{x})|/2$. We only apply the reflectance clustering prior (Section 4.4) on the finest pyramid level, and use the reflectance image computed on the second-finest pyramid level to compute the reflectance clusters.

6 Results

We tested our approach on several challenging real and synthetic datasets to evaluate its robustness, accuracy and runtime behavior in comparison to the current state of the art. Our test datasets consist of some real sequences (GIRL[1], TOY[1], DOWNSTAIRS[1], OBJECTS[1], HOUSE[2], CART[2]) and some synthetic sequences (SQUIRREL[1], CATHEDRAL[1], SANMIGUEL[2]) provided by existing intrinsic video decomposition techniques. In addition, we apply our approach to several of our live video streams captured by a webcam for demonstrating various applications. We perform both a qualitative and quantitative analysis of our results in comparison to the intrinsic video decomposition methods of Bonneel et al. [2014] and Ye et al. [2014]. Our method deals better with illumination effects such as shadows and shading than previous approaches, while being orders of magnitude faster. In the quantitative comparisons, we consistently obtain smaller decomposition errors than current state-of-the-art video techniques.

In most experiments, we used the following fixed set of parameters to instantiate our intrinsic decomposition energy: $w_{\text{reflectance}} = 0.5$, $w_{\text{intensity}} = p = 0.8$, $w_{\text{cs}} = 1$, and $w_{\text{data}} = w_{\text{shading}} = w_{\text{chromaticity}} = w_{\text{non-local}} = w_{\text{clustering}} = 10$. Note that our approach works out of the box for all sequences evaluated by us, with resolutions ranging from 640×360 to 960×540, including the live video footage in the same resolution range. Drastic deviation from this range may require parameter adjustments. Since the intrinsic decomposition problem is ambiguous, we globally scale our reflectance layer based on a single scalar (the shading layer is scaled inversely) to match

[1] http://media.au.tsinghua.edu.cn/yegenzhi/IntrinsicVideo.htm
[2] http://liris.cnrs.fr/~nbonneel/intrinsic.htm

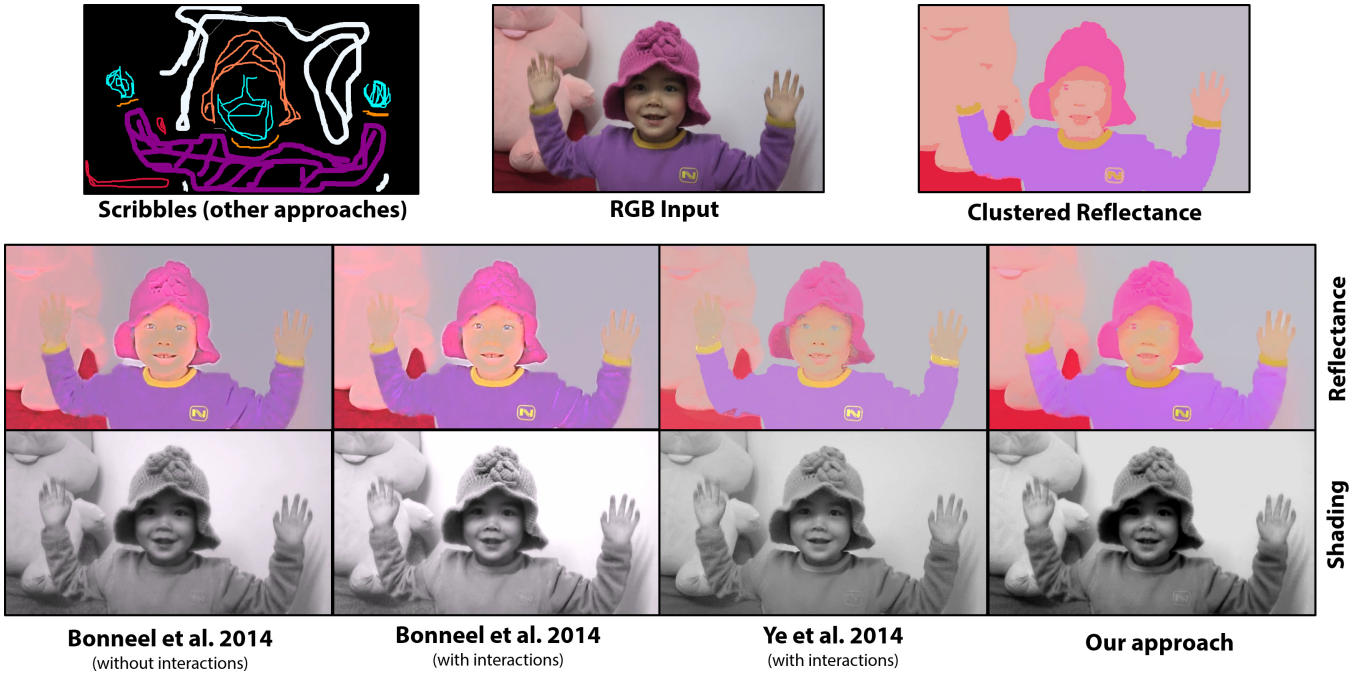


Figure 7: *State-of-the-art comparison to Bonneel et al. [2014] and Ye et al. [2014] on the* GIRL *sequence. Our approach obtains comparable or even higher-quality decompositions than previous approaches (less shading in the reflectance layer), while being orders of magnitude faster (10× faster than Bonneel et al. and 1800× faster than Ye et al.) and not requiring user input in the form of scribbles.*

the perceived brightness of previous state-of-the-art approaches. Note, the scaled results are still valid decompositions. We refer to the accompanying video for further results on the complete video sequences. The temporal consistency of our decomposition results can best be judged from these.

6.1 Qualitative Evaluation

We start with a qualitative comparison to the state-of-the-art approaches of Bonneel et al. [2014] and Ye et al. [2014] in Figure 7. Our approach obtains reflectance layers of higher quality, particularly in the more uniform regions (see the hat). The other two approaches more strongly bake shading variation into the reflectance map. We also separate the input (see creases of the shirt) better into its reflectance and shading components. This is possible due to our novel spatio-temporal prior. Note, the other methods require intricate user interaction, in the form of constant reflectance scribbles in the first frame of the video, to obtain reasonable decomposition results, whereas our approach is fully automatic and orders of magnitude faster (10× faster than Bonneel et al., 1800× faster than Ye et al.). In addition, the method of Bonneel et al. [2014] operates on grayscale images instead of RGB data. Therefore, the output reflectance has the same chromaticity as the input. This leads to artifacts if the assumption of white light or Lambertian surface is violated.

Our global spatio-temporal prior ensures that reflectance values of spatially or temporally distant objects with the same appearance are similar in the decomposition. This becomes especially apparent in the TOY sequence (see Figure 8), which contains several toy blocks with similar appearance. The previous state-of-the-art approaches struggle with this challenging scenario. In particular, they are unable to uniformly decompose the blue colored blocks and end up with a lot of shading detail in the reflectance layer. Note again, our method is orders of magnitude faster than these approaches and does not require user input in the form of scribbles.

6.2 Quantitative Evaluation

We use established error metrics [Grosse et al. 2009] to compare our results to ground truth data:

1. MSE (*mean squared error*) measures the average of the squared per-pixel deviations from the ground truth. In case of color images, we average over all channels.

2. LMSE (*local mean squared error*) measures the average MSE over a set of overlapping patches. The intensity of each patch is scaled by a single scalar value to minimize the per-patch MSE value. The metric is normalized so that an estimate of all zeros has the maximum possible score of 1. We use a patch size of 10×10.

3. DSSIM (*structural dissimilarity index*) is an information theoretic metric that measures the perceived change in structural information between two images.

We compute and state each metric separately for the reflectance and shading images, and also report the average as final result.

Figure 9 compares our results on the synthetic SANMIGUEL sequence to the approach of Bonneel et al. [2014]. Our approach achieve higher quality decompositions, especially in the foliage and on the background walls. The complex illumination pattern on the leaves is difficult to decompose with previous state-of-the-art approaches, even with user interaction in the form of scribbles. We are able to obtain decompositions of better quality fully automatically even in this challenging scenario. Note, our approach is also an order of magnitude faster. Figure 10 shows the per-frame MSE, LMSE and DSSIM results as plots for the complete sequence. We obtain consistently lower decomposition errors in almost all frames of the sequence. The increased temporal stability of our approach, compared to Bonneel et al. [2014], can be seen in the smaller variance of the error plots. The errors over the complete sequence are summarized in Table 1, separately for shading and reflectance layers,

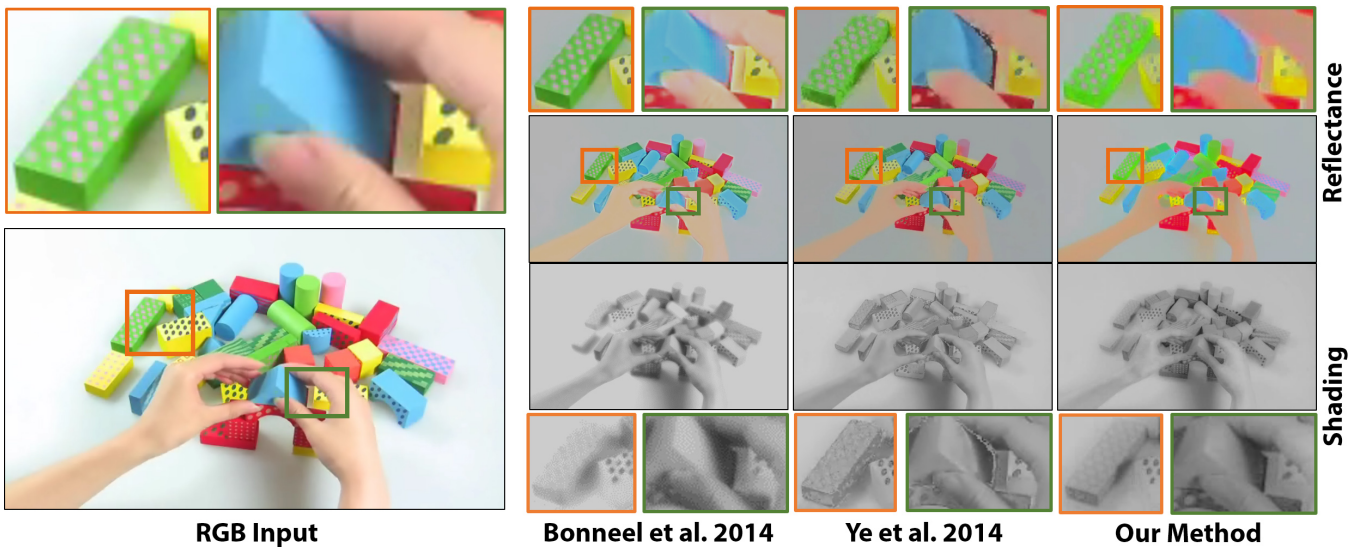


Figure 8: *State-of-the-art comparison to Bonneel et al. [2014] and Ye et al. [2014] on the* TOY *sequence. Our approach obtains decompositions of higher quality than previous approaches (less shading in the reflectance layer, sharper shading layer, less artifacts), while being orders of magnitude faster (10× faster than Bonneel et al., 200× faster than Ye et al.) and not requiring user input in the form of scribbles.*

Table 1: *Quantitative comparison on the* SANMIGUEL *sequence: our decompositions obtain a lower error (bold) than previous work.*

	MSE			LMSE			DSSIM		
Approach	shading	reflectance	mean	shading	reflectance	mean	shading	reflectance	mean
Bonneel et al. [2014] (no scribbles)	0.0063	0.0258	0.0161	0.1564	0.1332	0.1447	0.2794	0.3226	0.3011
Bonneel et al. [2014] (scribbles)	0.0030	0.0166	0.0097	0.0886	0.1029	0.0947	0.1753	0.2898	0.2302
Our approach	0.0028	**0.0151**	**0.0089**	**0.0309**	**0.0622**	**0.0461**	**0.1304**	**0.2566**	**0.1915**
Our approach (w/o non-local prior)	**0.0027**	0.0173	0.0099	0.0421	0.0961	0.0688	0.1367	0.2693	0.2014

and averaged, and also indicate the superior performance of our approach, even without using user scribbles.

6.3 Evaluation on 'Intrinsic Images in the Wild' Dataset

We additionally evaluate our approach on the 'Intrinsic Images in the Wild' benchmark dataset of Bell et al. [2014]. Towards this goal, we disable the temporal consistency prior term in our formulation, decompose the 5,230 individual images in the dataset and evaluate the *weighted human disagreement rate* (WHDR), which compares the manual annotations on the images with the decomposed reflectance images. We obtain a $\text{WHDR}_{10\%}$ score of 31.4%. Note that our technique is not meant to compete with traditional intrinsic *single-image* decomposition techniques, as we address a different set of challenges in intrinsic decomposition of *live videos*.

6.4 Influence of the Different Energy Terms

Our intrinsic decomposition approach obtains high-quality results due to our carefully crafted decomposition energy function. Next, we evaluate the relative importance of the different objective terms. Figure 11 shows the reflectance images for different instantiations of our decomposition energy, where we successively disabled certain components by setting the respective weight to zero (see also video). The best decomposition results are obtained by our full combined energy. The chromaticity prior helps to keep the output reflectance close to the input's chromaticity leading to more saturated results. The clustering prior is particularly useful in decomposing the challenging dark shadow regions in the image accurately. Without it, illumination effects such as shadows and shading become part of

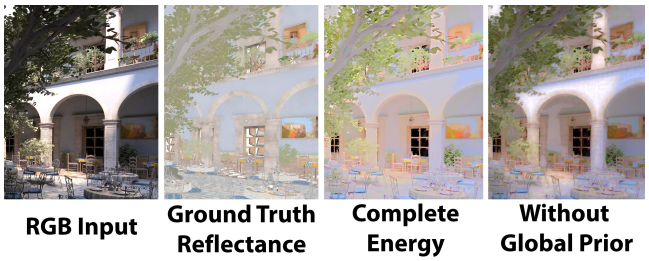


Figure 12: *Influence of different priors on the* SANMIGUEL *sequence. The sampling-based global spatial prior constraint helps to remove shading variations from the reflectance layer.*

the reflectance layer. The spatio-temporal prior ensures the global consistency of the reflectance layer, even for disconnected regions of the same material. In addition, it leads to temporally coherent results. The added global spatial consistency can even better be judged from the SANMIGUEL sequence (see Figure 12). Note that the background wall in the courtyard, the floor and the leaves, all incorrectly contain illumination and shadows if this prior is not applied. The lower error in the ground-truth comparison (see Table 1) also reflects this difference in quality. Therefore, all proposed priors contribute significantly to the accuracy of the obtained decomposition results.

6.5 Runtime and Convergence

Figure 13 shows the convergence behavior of our novel nested IRLS approach. The staircase pattern corresponds to the number of hierarchy levels (5 in this case). For this experiment, we used 7 non-linear

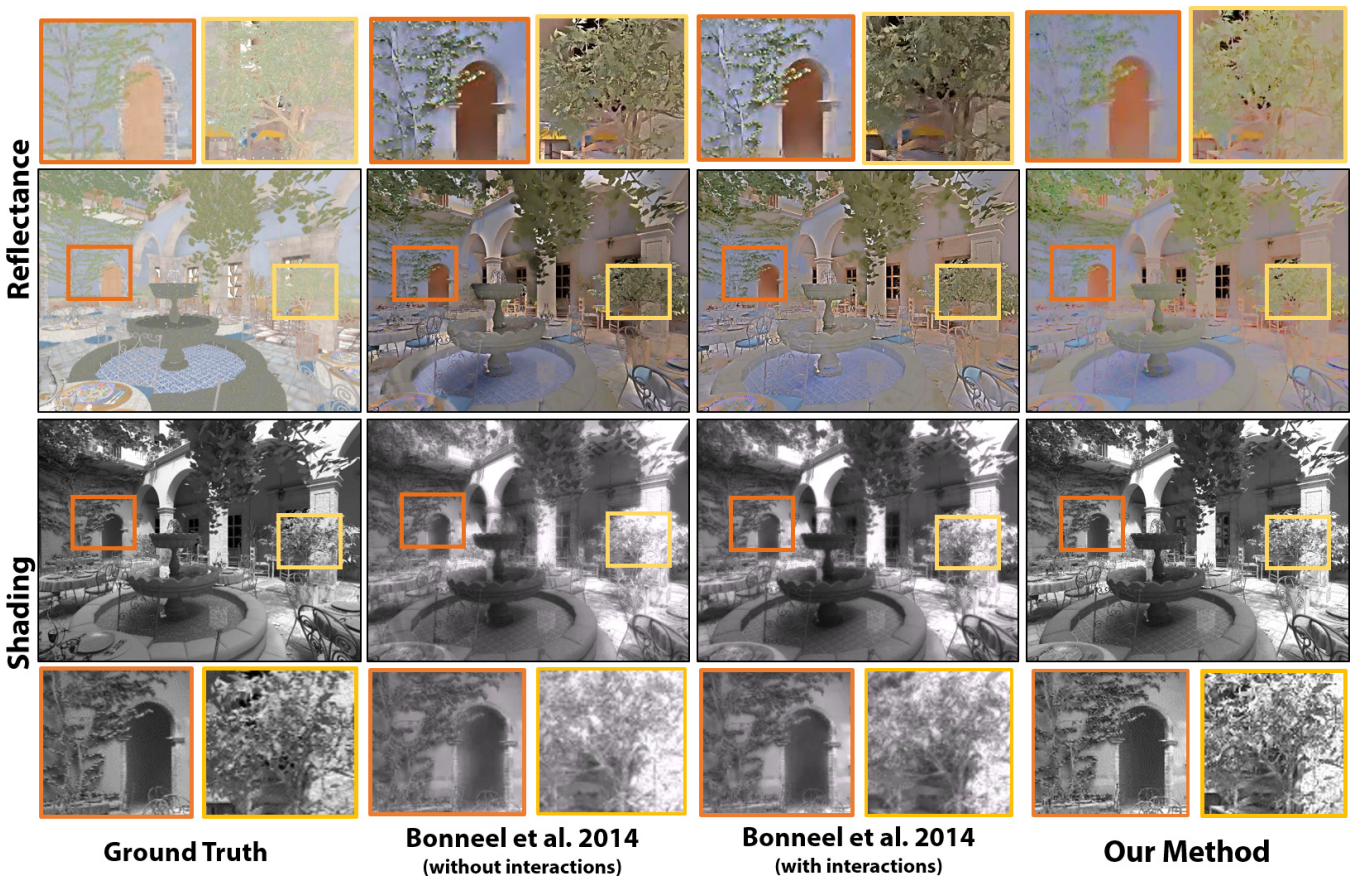


Figure 9: *Ground-truth comparison on the* SANMIGUEL *sequence. Our approach obtains decompositions that more closely match the ground truth. Bonneel et al.'s result artificially blurs the shading layer and contains small-scale shading in the reflectance layer. Even user-provided scribbles do not alleviate this issue. Our approach is also one order of magnitude faster and can be applied to live video data.*

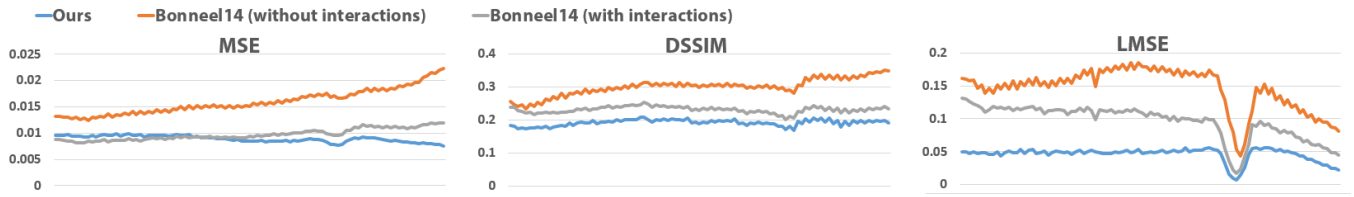


Figure 10: *Quantitative evaluation: our approach obtains lower MSE, DSSIM and LMSE errors than the approach of Bonneel et al. [2014] on the* SANMIGUEL *sequence, while also being one order of magnitude faster and not relying on user input in the form of scribbles.*

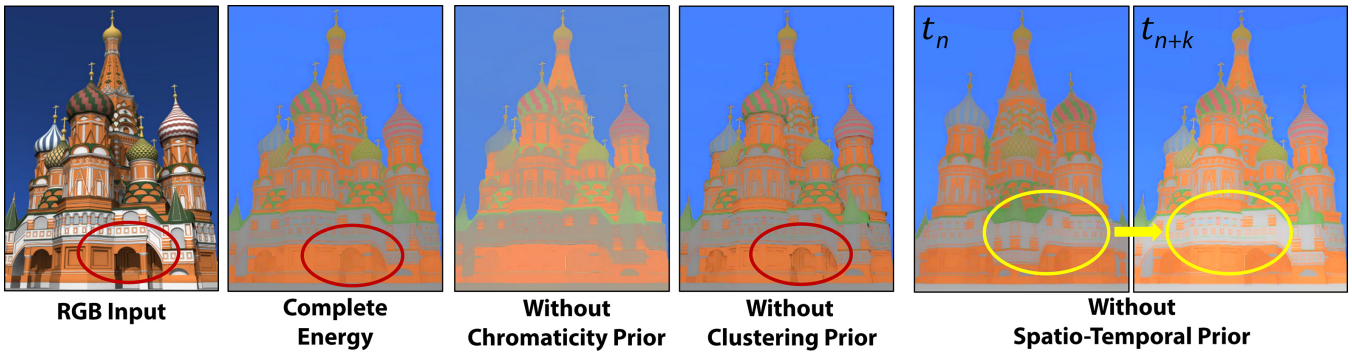
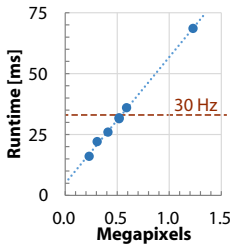


Figure 11: *Influence of energy terms: reflectance result on the* CATHEDRAL *sequence. The best reflectance image is obtained with our full energy. Without the* ***chromaticity prior****, the output reflectance color deviates from the input. The* ***clustering prior*** *removes shading variation from the reflectance layer (red circles). Without the* ***spatio-temporal prior****, the decomposition is temporally unstable (yellow circles).*

Table 2: *Runtime performance for different input resolutions.*

Sequence	Resolution	Time
HOUSE	1024 × 576	36.0 ms
GIRL	960 × 540	31.8 ms
DOWNSTAIRS	960 × 540	31.5 ms
TOY	640 × 360	16.1 ms
SQUIRREL	854 × 480	26.0 ms
SANMIGUEL	1280 × 960	68.6 ms
Live	640 × 480	22.1 ms



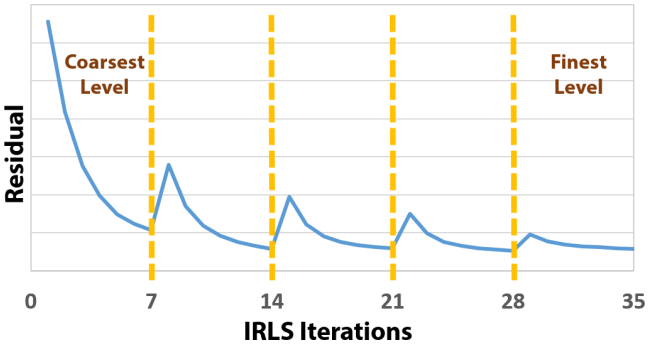


Figure 13: *Convergence: The residual error is always decreasing.*

IRLS iterations per level with 8 PCG steps each. As can be seen, our IRLS approach converges on each hierarchy level in about 4 iteration steps. Due to the used hierarchy, global convergence is fast and all error frequencies are efficiently resolved. Since convergence on a single level is reached after only a few iteration steps, in the following, we set the number of IRLS iterations to 4; all other settings are kept unchanged. This is a good trade-off between accuracy and runtime performance. We give the mean per-frame runtime of our approach for seven sequences with different input resolutions in Table 2. Runtime is essentially linear in the number of pixels in the video, and we achieve frame rates of more than 30 Hz for input resolutions up to 950×540. In particular, live sequences at VGA resolution are processed in less than 23 ms, which guarantees real-time feedback. All timings have been measured on a commodity Nvidia GTX Titan graphics card.

7 Applications

Our approach, for the first time, enables high-quality intrinsic decompositions in real time. This real-time capability is the basis for a large variety of video editing applications, which we showcase in a live setup. Our live setup is based on a commodity webcam (*Logitech HD Pro C920*), which captures RGB video at 30 Hz. We use a color resolution of 640×480 for all applications. The camera's exposure, white balance and focal length were manually set to a fixed value. The quality of our live decompositions and the live editing results can best be judged from the accompanying video.

7.1 Dynamic Reflectance Recoloring

This demo showcases the realistic recoloring of different objects in live video footage. For each captured frame, we first compute the intrinsic decomposition and apply chromaticity keying to the reflectance layer to select a subregion for which a different reflectance value is set. Note that in the recolored composite (see Figure 14), shading variations are realistically preserved. The real-time setting enables immediate visual feedback, even if parameters are changed.

Figure 14: *Reflectance recoloring on the* GIRL *sequence. We recolor the girl's shirt in real time using our intrinsic decomposition approach. Note that the shading detail is preserved.*

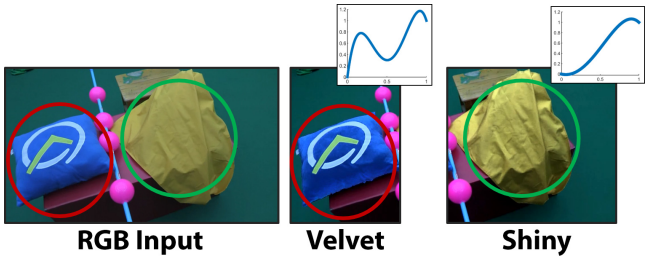


Figure 15: *Editing material appearances on the* OBJECTS *sequence. The cushion looks like velvet (red circles), and the cloth is modified to appear shinier (green circles). The blue curves show the tone mapping applied to the corresponding regions of the shading layer to achieve the effect in each case.*

7.2 Editing Material Appearances

This application demonstrates the modification of material properties other than reflectance at real-time rates; we borrow the term material editing from Ye et al. [2014], who showed similar effects in an off-line setup. We apply tone mapping to a selected region of the shading layer that has been computed in real time. The tone mapping function is provided interactively by the user based on a sparse set of control points. Based on this, we can for example change the appearance of different objects in live video footage (see Figure 15). The cushion is modified to have a velvet surface, whereas in the second image, the cloth is made to appear more shiny. Note, the reflectance of the objects is not influenced by this operation, since the editing is performed in the shading domain.

7.3 Realistic Texture Replacement

We demonstrate real-time illumination-aware retexturing of live video footage. In contrast to the two previous examples, which applied a constant color or appearance change to a chroma-keyed region, this demo requires temporal correspondences. To this end, we use the feature-based PTAM [Klein and Murray 2007] technique which tracks the camera's motion based on a set of sparse feature points in real time. Retexturing is performed by applying a reflectance layer texture to the handled planar surfaces. Note however, arbitrary objects can be handled easily if a corresponding proxy ge-

Figure 16: *Realistic texture replacement: we add a virtual painting (left), and apply a brick texture (right). The textures realistically interact with the illumination (red circles). With a naïve texturing approach, shadows are lost. See this result in motion in our video.*

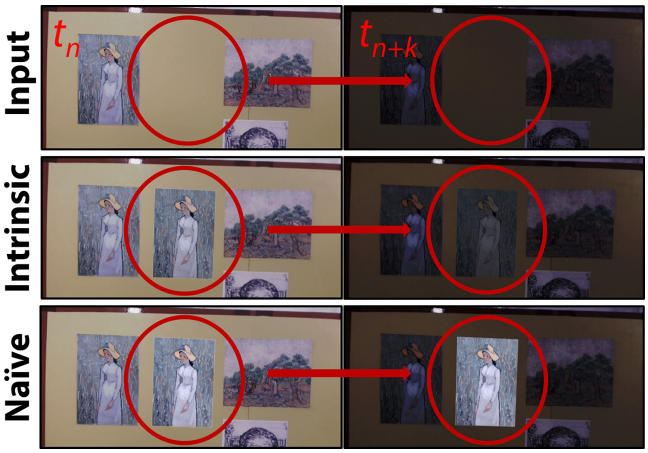


Figure 17: *Realistic texture replacement: we add a virtual painting to the wall. The textures realistically reflect the illumination change (red circles) caused by dimming the lights. Note that a naïve texturing approach leads to unrealistic results.*

ometry is available. In Figure 16, we add a Van Gogh painting ("Girl in White, 1890") to the scene. Our intrinsic retexturing method adds shadows and lighting, which are part of the scene, to the texture in real time. This allows for photorealisitic results. The naïvely added texture, i.e. replacing the texture in the non-decomposed RGB video, does not interact with the illumination, hence making it appear synthetic. We also retexture the notice board with a brick texture. In Figure 17, we dim the light source. Our approach properly relights the synthetic texture. Note, the virtual paintings and bricks are correctly and realistically interacting with the real-world illumination. In contrast, naïve retexturing leads to unrealistic results. In Figure 18, we add a leaf texture to the side of a carton. Note the different shading on the added decal, depending on which side of the box it is placed. Please also refer to the accompanying video.

7.4 Live Video Abstraction & Stylization

Next, we demonstrate abstraction and artistic stylization of live video footage. Abstraction of images and video has been shown to be an important tool in recognition and memory tasks [Winnemöller et al. 2006]. Our reflectance video stream does not contain shading

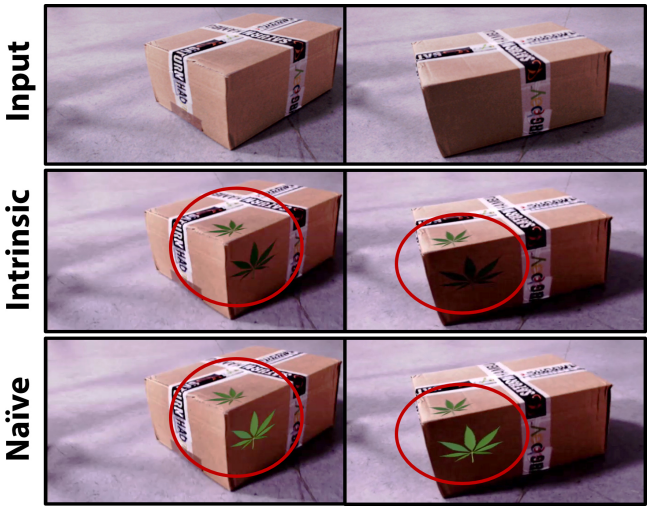


Figure 18: *Realistic texture replacement: we add two virtual decals to a box. Intrinsic texturing realistically interacts with the real-world shading. Note that naïve texturing leads to unrealistic results.*

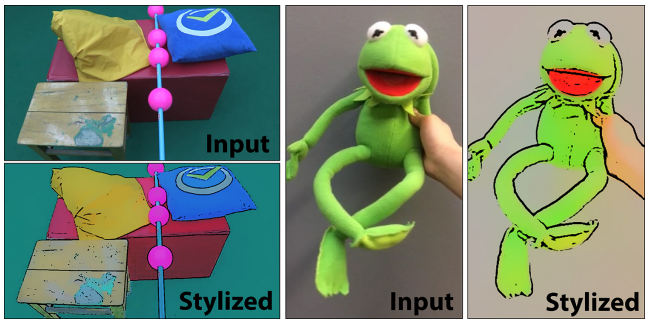


Figure 19: *Live video stylization using a cartoon-style effect.*

information and hence already captures an abstract version of the scene. By increasing the contrast of major edges of the shading layer, and suppressing low-contrast regions, a nice cartoon-style effect can be achieved. To this end, we apply a difference-of-gradient (DoG) filter [Winnemöller et al. 2006] to the shading layer and then recombine it with the reflectance layer (see Figure 19). The spatial scale, sensitivity and sharpness of the resulting edges can all be controlled interactively by the user. Unlike previous video abstraction techniques, our method is directly applied to the shading layer, hence enforcing only the shading edges, not edges between albedo regions which are often also stylized in previous methods.

8 Discussion

We demonstrated the first approach for intrinsic decomposition of live video streams at real-time framerates. While we achieve high-quality results on par or surpassing the current state-of-the-art off-line methods in terms of robustness, accuracy and runtime, we make some simplifying assumptions to make this hard inverse problem tractable. Note that these assumptions are common to almost all state-of-the-art intrinsic decomposition approaches, even to the off-line methods. In the following, we discuss the main assumptions:

Monochromatic Illumination: All illuminants are assumed to emit pure white light, a reasonable assumption for many real-world scenes. Therefore, a perceived change in chromaticity can be directly attributed to a change in material reflectance.

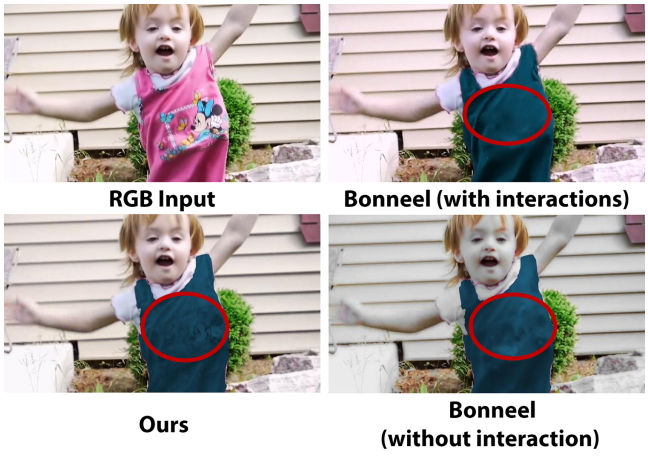


Figure 20: *Recoloring of highly textured objects: we obtain comparable recoloring results to the approach of Bonneel et al. [2014] with default parameters and no user interaction (bottom). With additional scribble-based user interaction, Bonneel et al. obtain results with fewer texture copy artifacts (top right). Note that our approach is one order of magnitude faster and does not use any user input, since this is infeasible in the proposed live video editing context.*

Diffuse Reflectance: All objects in the scene are assumed to have a purely diffuse reflectance. This is a soft assumption since our method handles non-diffuse objects gracefully, as long as the material is not highly specular.

Sparse Reflectance: We assume the scene to be comprised of a relatively small number of uniformly colored surface patches. In natural scenes with high-frequency texture or smooth color gradients, this assumption might be violated. We show one such example in the context of recoloring in Figure 20.

Direct Illumination: We only consider direct illumination effects. Complex multi-bounce illumination such as caustics or color bleeding are not explicitly handled and might be mistaken for reflectance variation.

Despite these simplifying assumptions, our approach produces plausible decomposition results at previously unseen frame rates and without any user interaction.

9 Conclusion

We presented the first approach to compute intrinsic decompositions of monocular live video footage in real time. High-quality and temporally coherent decompositions are obtained without the need for an explicit correspondence search. Real-time optimization is possible due to a carefully crafted data-parallel solver for general ℓ_2–ℓ_p-optimization problems. We demonstrated the capabilities of our approach on live video footage as well as on synthetic data. The qualitative and quantitative evaluation shows that our approach is on par with or even outperforms current state-of-the-art techniques in terms of robustness, accuracy and runtime.

We believe that the real-time capabilities of our intrinsic decomposition approach will pave the way for many novel augmented reality applications that build on top of the presented realistic recoloring, relighting and texture editing functionality. In the future, we want to relax some of the made assumption to make our approach applicable to an even wider range of settings, such as colored multi-bounce illumination, highly specular surfaces or textured objects. The incorporation of depth information into the optimization process will help to resolve some of the inherent ambiguities of the intrinsic decomposition problem, leading to even more accurate results.

Acknowledgments We thank the anonymous reviewers for their helpful feedback, and Nicolas Bonneel and Yebin Liu for providing the data for comparisons to their state-of-the-art techniques. We thank Franziska Müller for patiently assisting with the live video editing demo. This work was supported by the ERC Starting Grant CapReal (335545).

References

BARRON, J. T., AND MALIK, J. 2013. Intrinsic scene properties from a single RGB-D image. In *CVPR*.

BARRON, J. T., AND MALIK, J. 2015. Shape, illumination, and reflectance from shading. *IEEE Transactions on Pattern Analysis and Machine Intelligence 37*, 8, 1670–1687.

BARROW, H. G., AND TENENBAUM, J. M. 1978. Recovering intrinsic scene characteristics from images. Tech. Rep. 157, AI Center, SRI International.

BELL, M., AND FREEMAN, W. T. 2001. Learning local evidence for shading and reflection. In *CVPR*.

BELL, S., BALA, K., AND SNAVELY, N. 2014. Intrinsic images in the wild. *ACM Transactions on Graphics 33*, 4 (July), 159:1–12.

BI, S., HAN, X., AND YU, Y. 2015. An l_1 image transform for edge-preserving smoothing and scene-level intrinsic decomposition. *ACM Transactions on Graphics 34*, 4 (July), 78:1–12.

BONNEEL, N., SUNKAVALLI, K., TOMPKIN, J., SUN, D., PARIS, S., AND PFISTER, H. 2014. Interactive intrinsic video editing. *ACM Transactions on Graphics 33*, 6 (November), 197:1–10.

BONNEEL, N., TOMPKIN, J., SUNKAVALLI, K., SUN, D., PARIS, S., AND PFISTER, H. 2015. Blind video temporal consistency. *ACM Transactions on Graphics 34*, 6 (November), 196:1–9.

BOUSSEAU, A., PARIS, S., AND DURAND, F. 2009. User-assisted intrinsic images. *ACM Transactions on Graphics 28*, 5 (December), 130:1–10.

CHANG, J., CABEZAS, R., AND FISHER III, J. W. 2014. Bayesian nonparametric intrinsic image decomposition. In *ECCV*.

CHEN, Q., AND KOLTUN, V. 2013. A simple model for intrinsic image decomposition with depth cues. In *ICCV*.

DUCHÊNE, S., RIANT, C., CHAURASIA, G., MORENO, J. L., LAFFONT, P.-Y., POPOV, S., BOUSSEAU, A., AND DRETTAKIS, G. 2015. Multiview intrinsic images of outdoors scenes with an application to relighting. *ACM Transactions on Graphics 34*, 5 (October), 164:1–16.

GARCES, E., MUNOZ, A., LOPEZ-MORENO, J., AND GUTIERREZ, D. 2012. Intrinsic images by clustering. *CGF 31*, 4, 1415–1424.

GEHLER, P. V., ROTHER, C., KIEFEL, M., ZHANG, L., AND SCHÖLKOPF, B. 2011. Recovering intrinsic images with a global sparsity prior on reflectance. In *NIPS*.

GROSSE, R., JOHNSON, M. K., ADELSON, E. H., AND FREEMAN, W. T. 2009. Ground truth dataset and baseline evaluations for intrinsic image algorithms. In *ICCV*.

HACHAMA, M., GHANEM, B., AND WONKA, P. 2015. Intrinsic scene decomposition from RGB-D images. In *ICCV*.

HAUAGGE, D., WEHRWEIN, S., BALA, K., AND SNAVELY, N. 2013. Photometric ambient occlusion. In *CVPR*.

HOLLAND, P. W., AND WELSCH, R. E. 1977. Robust regression using iteratively reweighted least-squares. *Communications in Statistics – Theory and Methods 6*, 9 (September), 813–827.

HORN, B. K. P. 1974. Determining lightness from an image. *Computer Graphics and Image Processing 3*, 4, 277–299.

JIANG, X., SCHOFIELD, A. J., AND WYATT, J. L. 2010. Correlation-based intrinsic image extraction from a single image. In *ECCV*.

JOSHI, N., ZITNICK, C., SZELISKI, R., AND KRIEGMAN, D. 2009. Image deblurring and denoising using color priors. In *CVPR*.

KLEIN, G., AND MURRAY, D. 2007. Parallel tracking and mapping for small AR workspaces. In *ISMAR*.

KONG, N., GEHLER, P. V., AND BLACK, M. J. 2014. Intrinsic video. In *ECCV*.

LAFFONT, P.-Y., AND BAZIN, J.-C. 2015. Intrinsic decomposition of image sequences from local temporal variations. In *ICCV*.

LAFFONT, P.-Y., BOUSSEAU, A., PARIS, S., DURAND, F., AND DRETTAKIS, G. 2012. Coherent intrinsic images from photo collections. *ACM Transactions on Graphics 31*, 6 (November), 202:1–11.

LAFFONT, P.-Y., BOUSSEAU, A., AND DRETTAKIS, G. 2013. Rich intrinsic image decomposition of outdoor scenes from multiple views. *IEEE Transactions on Visualization and Computer Graphics 19*, 2 (February), 210–224.

LAND, E. H., AND MCCANN, J. J. 1971. Lightness and retinex theory. *Journal of the Optical Society of America 61*, 1, 1–11.

LEE, K. J., ZHAO, Q., TONG, X., GONG, M., IZADI, S., LEE, S. U., TAN, P., AND LIN, S. 2012. Estimation of intrinsic image sequences from image+depth video. In *ECCV*.

LEVIN, A., AND WEISS, Y. 2007. User assisted separation of reflections from a single image using a sparsity prior. *IEEE Transactions on Pattern Analysis and Machine Intelligence 29*, 9 (September), 1647–1654.

LEVIN, A., FERGUS, R., DURAND, F., AND FREEMAN, W. T. 2007. Image and depth from a conventional camera with a coded aperture. *ACM Transactions on Graphics 26*, 3 (July), 70.

LI, Y., AND BROWN, M. S. 2014. Single image layer separation using relative smoothness. In *CVPR*.

MATSUSHITA, Y., LIN, S., KANG, S., AND SHUM, H.-Y. 2004. Estimating intrinsic images from image sequences with biased illumination. In *ECCV*.

SHEN, L., AND YEO, C. 2011. Intrinsic images decomposition using a local and global sparse representation of reflectance. In *CVPR*.

SHEN, L., TAN, P., AND LIN, S. 2008. Intrinsic image decomposition with non-local texture cues. In *CVPR*.

SHEN, J., YANG, X., JIA, Y., AND LI, X. 2011. Intrinsic images using optimization. In *CVPR*.

SHEN, J., YAN, X., CHEN, L., SUN, H., AND LI, X. 2014. Retexturing by intrinsic video. *Information Sciences 281*, 726–735.

TAPPEN, M. F., FREEMAN, W. T., AND ADELSON, E. H. 2005. Recovering intrinsic images from a single image. *IEEE Transactions on Pattern Analysis and Machine Intelligence 27*, 9, 1459–1472.

WEBER, D., BENDER, J., SCHNOES, M., STORK, A., AND FELLNER, D. 2013. Efficient GPU data structures and methods to solve sparse linear systems in dynamics applications. *Computer Graphics Forum 32*, 1, 16–26.

WEISS, Y. 2001. Deriving intrinsic images from image sequences. In *ICCV*.

WINNEMÖLLER, H., OLSEN, S. C., AND GOOCH, B. 2006. Real-time video abstraction. *ACM Transactions on Graphics 25*, 3 (July), 1221–1226.

WU, C., ZOLLHÖFER, M., NIESSNER, M., STAMMINGER, M., IZADI, S., AND THEOBALT, C. 2014. Real-time shading-based refinement for consumer depth cameras. *ACM Transactions on Graphics 33*, 6 (November), 200:1–10.

YE, G., GARCES, E., LIU, Y., DAI, Q., AND GUTIERREZ, D. 2014. Intrinsic video and applications. *ACM Transactions on Graphics 33*, 4 (July), 80:1–11.

ZHAO, Q., TAN, P., DAI, Q., SHEN, L., WU, E., AND LIN, S. 2012. A closed-form solution to Retinex with nonlocal texture constraints. *IEEE Transactions on Pattern Analysis and Machine Intelligence 34*, 7 (July), 1437–1444.

ZHAO, H. K. 1996. *Generalized Schwarz Alternating Procedure for Domain Decomposition*. University of California, Los Angeles.

ZHOU, T., KRÄHENBÜHL, P., AND EFROS, A. 2015. Learning data-driven reflectance priors for intrinsic image decomposition. In *ICCV*.

ZOLLHÖFER, M., NIESSNER, M., IZADI, S., RHEMANN, C., ZACH, C., FISHER, M., WU, C., FITZGIBBON, A., LOOP, C., THEOBALT, C., AND STAMMINGER, M. 2014. Real-time non-rigid reconstruction using an RGB-D camera. *ACM Transactions on Graphics 33*, 4 (July), 156:1–12.

ZOLLHÖFER, M., DAI, A., INNMANN, M., WU, C., STAMMINGER, M., THEOBALT, C., AND NIESSNER, M. 2015. Shading-based refinement on volumetric signed distance functions. *ACM Transactions on Graphics 34*, 4 (July), 96:1–14.

ZORAN, D., ISOLA, P., KRISHNAN, D., AND FREEMAN, W. T. 2015. Learning ordinal relationships for mid-level vision. In *ICCV*.