ISSN 1213-6972 *Volume 25, Number 2, 2017*

Journal of WSCG

An international journal of algorithms, data structures and techniques for computer graphics and visualization, surface meshing and modeling, global illumination, computer vision, image processing and pattern recognition, computational geometry, visual human interaction and virtual reality, animation, multimedia systems and applications in parallel, distributed and mobile environment.

EDITOR – IN – CHIEF

Václav Skala

Vaclav Skala – Union Agency

Journal of WSCG

Editor-in-Chief: Vaclav Skala
c/o University of West Bohemia
Faculty of Applied Sciences
Univerzitni 8
CZ 306 14 Plzen
Czech Republic
http://www.VaclavSkala.eu

Managing Editor: Vaclav Skala

Printed and Published by:
Vaclav Skala - Union Agency
Na Mazinach 9
CZ 322 00 Plzen
Czech Republic

Hardcopy: ***ISSN 1213 – 6972***
CD ROM: ***ISSN 1213 – 6980***
On-line: ***ISSN 1213 – 6964***

Journal of WSCG

Editor-in-Chief

Vaclav Skala

c/o University of West Bohemia
Faculty of Applied Sciences
Department of Computer Science and Engineering
Univerzitni 8
CZ 306 14 Plzen
Czech Republic

http://www.VaclavSkala.eu
Journal of WSCG URLs: http://www.wscg.eu or http://wscg.zcu.cz/jwscg

Editorial Advisory Board
MEMBERS

WSCG 2017

International Program Committee

Adzhiev, Valery (United Kingdom)
Anderson, Maciel (Brazil)
Benes, Bedrich (United States)
Bilbao, Javier,J. (Spain)
Bourke, Paul (Australia)
Daniel, Marc (France)
de Geus, Klaus (Brazil)
Drechsler, Klaus (Germany)
Feito, Francisco (Spain)
Ferguson, Stuart (United Kingdom)
Galo, Mauricio (Brazil)
Giannini, Franca (Italy)
Gobbetti, Enrico (Italy)
Gudukbay, Ugur (Turkey)
Juan, M.-Carmen (Spain)
Kenny, Erleben (Denmark)
Kim, Jinman (Australia)
Kim, HyungSeok (Korea)
Lobachev, Oleg (Germany)
Molla, Ramon (Spain)
Montrucchio, Bartolomeo (Italy)
Muller, Heinrich (Germany)
Murtagh, Fionn (United Kingdom)
Pan, Rongjiang (China)
Pedrini, Helio (Brazil)
Platis, Nikos (Greece)
Ramires Fernandes, Antonio (Portugal)
Richardson, John (United States)
Ritter, Marcel (Austria)
Rojas-Sola, Jose Ignacio (Spain)
Sanna, Andrea (Italy)
Segura, Rafael (Spain)
Skala, Vaclav (Czech Republic)
Sousa, A.Augusto (Portugal)
Szecsi, Laszlo (Hungary)
Teschner, Matthias (Germany)
Tokuta, Alade (United States)
Umetani, Nobuyuki (Japan)
Wu, Shin-Ting (Brazil)
Wuensche, Burkhard,C. (New Zealand)
Wuethrich, Charles (Germany)
Yao, Junfeng (China)

WSCG 2017

Board of Reviewers

Richardson, John (United States)
Ritter, Marcel (Austria)
Rodrigues, Joao (Portugal)
Rojas-Sola, Jose Ignacio (Spain)
Sanna, Andrea (Italy)
Schwaerzler, Michael (Austria)
Segura, Rafael (Spain)
Serano, Ana (Spain)
Sik-Lanyi, Cecilia (Hungary)
Sommer, Bjorn (Germany)
Sousa, A.Augusto (Portugal)
Szecsi, Laszlo (Hungary)
Teschner, Matthias (Germany)
Todt, Eduardo (Brazil)
Tokuta, Alade (United States)
Tytkowski, Krzysztof (Poland)
Umetani, Nobuyuki ()
Umlauf, Georg (Germany)
Vanderhaeghe, David (France)
Vidal, Vincent (France)
Vierjahn, Tom (Germany)
Wu, Shin-Ting (Brazil)
Wuensche, Burkhard,C. (New Zealand)
Wuethrich, Charles (Germany)
Yao, Junfeng (China)
Yoshizawa, Shin (Japan)
YU, Qizhi (United Kingdom)
Zhao, Qiang (China)

Journal of WSCG

Vol.25, No.2, 2017

Contents

Accelerating Discrete Wavelet Transforms on Parallel Architectures

David Barina Michal Kula Michal Matysek Pavel Zemcik

Centre of Excellence IT4Innovations
Faculty of Information Technology
Brno University of Technology
Bozetechova 1/2, Brno
Czech Republic
{ibarina,ikula,imatysek,zemcik}@fit.vutbr.cz

ABSTRACT

The 2-D discrete wavelet transform (DWT) can be found in the heart of many image-processing algorithms. Until recently, several studies have compared the performance of such transform on various shared-memory parallel architectures, especially on graphics processing units (GPUs). All these studies, however, considered only separable calculation schemes. We show that corresponding separable parts can be merged into non-separable units, which halves the number of steps. In addition, we introduce an optional optimization approach leading to a reduction in the number of arithmetic operations. The discussed schemes were adapted on the OpenCL framework and pixel shaders, and then evaluated using GPUs of two biggest vendors. We demonstrate the performance of the proposed non-separable methods by comparison with existing separable schemes. The non-separable schemes outperform their separable counterparts on numerous setups, especially considering the pixel shaders.

Keywords

Discrete wavelet transform, Image processing, Synchronization, Graphics processors

1 INTRODUCTION

The discrete wavelet transform became a very popular image processing tool in last decades. A widespread use of this transform has resulted in a development of fast algorithms on all sorts of computer systems, including shared-memory parallel architectures. At present, the GPU is considered as a typical representative of such parallel architectures. In this regard, several studies have compared the performance of various 2-D DWT computational approaches on GPUs. All of these studies are based on separable schemes, whose operations are oriented either horizontally or vertically. These schemes comprise the convolution and lifting. The lifting requires fewer arithmetic operations as compared with the convolution, at the cost of introducing some data dependencies. The number of operations should be proportional to a transform performance. However, also the data dependencies may form a bottleneck, especially on shared-memory parallel architectures.

In this paper, we show that the fastest scheme for a given architecture can be obtained by fusing the corresponding parts of the separable schemes into new structures. Several new non-separable schemes are obtained in this way. More precisely, the underlying operations of these schemes can be associated with neither horizontal nor vertical axes. In addition, we present an approach where each scheme can be adapted to a particular platform in order to reduce the number of operations. This possibility was completely omitted in existing studies. Our reasoning is supported by extensive experiments on GPUs using OpenCL and pixel shaders (fragment shaders in OpenGL terminology). The presented schemes are general, and they are not limited to any specific type of DWT. To clarify the situation, they all compute the same values.

The rest of this paper is organized as follows. Section Background formally introduces the problem definition. Section Related Work briefly presents the existing separable approaches. Section Proposed Schemes presents the proposed non-separable schemes. Section Optimization Approach discusses the optimization approach that reduces the number of operations. Section Evaluation evaluates the performance on GPUs in the pixel shaders and OpenCL framework. Eventually, Section Conclusions closes the paper. This section is followed by Section Appendix for readers not familiar with signal-processing notations.

2 BACKGROUND

Since the separable schemes are built on the one-dimensional transform, a widely-used z-transform is used for the description of underlying FIR filters. The transfer function of the filter (g_k) is the polynomial

$$G(z) = \sum_k g_k z^{-k},$$

where the k refers to the time axis. Below in the text, the one-dimensional transforms are used in conjunction with two-dimensional signals. For this case, the transfer function of the filter $\left(g_{k_m,k_n}\right)$ is defined as the bivariate polynomial

$$G(z_m, z_n) = \sum_{k_m} \sum_{k_n} g_{k_m,k_n} z_m^{-k_m} z_n^{-k_n},$$

where the subscript m refers to the horizontal axis and n to the vertical one. The $G^*(z_m, z_n) = G(z_n, z_m)$ is a polynomial transposed to a polynomial $G(z_m, z_n)$. A shortened notation G is only written in order to keep the notation readable.

A discrete wavelet transform is a signal-processing tool which is suitable for the decomposition of a signal into low-pass and high-pass components. In detail, the single-scale transform splits the input signal into two components, according to a parity of its samples. Therefore, the DWT is described by 2×2 matrices. As shown by Mallat [10], the transform can be computed by a pair of filters followed by subsampling by a factor of 2. The filters are referred to as $\mathrm{G}_0, \mathrm{G}_1$. The transform can also be represented by the polyphase matrix

$$\begin{bmatrix} \mathrm{G_1}^{(o)} & \mathrm{G_1}^{(e)} \\ \mathrm{G_0}^{(o)} & \mathrm{G_0}^{(e)} \end{bmatrix}, \tag{1}$$

where the polynomials $\mathrm{G}^{(e)}$ and $\mathrm{G}^{(o)}$ refer to the even and odd terms of G. This polyphase matrix defines the convolution scheme. To avoid misunderstandings, it is necessary to say that, in this paper, column vectors are transformed to become another columns. For example, $\mathrm{y} = \mathrm{Mx}$ and $\mathrm{y} = \mathrm{M_2 M_1 x}$ are transforms represented by one and two matrices, respectively. Following the algorithm by Sweldens [14, 4], the convolution scheme in (1) can be factored into a sequence

$$\prod_k \begin{bmatrix} 1 & \mathrm{U}^{(k)} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ \mathrm{P}^{(k)} & 1 \end{bmatrix} \tag{2}$$

of K pairs of short filterings, known as the lifting scheme. The filters employed in (2) are referred to as the lifting steps. Usually, the first step $\mathrm{P}^{(k)}$ in the kth pair is referred to as the predict and the second one $\mathrm{U}^{(k)}$ as the update. The lifting scheme reduces the number of operations by up to half. Since this paper is mostly focused on a single pair of steps, the superscript (k) is omitted in the text below. Note that the number of operations is calculated as the number of distinct (in a column) terms of all polynomials in all matrices, excluding units on diagonals.

Considering the shared-memory parallel architectures, the processing of single or several samples is mapped to independent processing units. In order to avoid race conditions during data exchange, the units must use some synchronization method (barrier). In the lifting scheme, the barriers are required before the lifting steps. In the convolution scheme, the barrier is only required before starting the calculation. In this paper, the barriers are indicated by the | symbol. For example, $\mathrm{M_2}|\mathrm{M_1}$ are two adjacent lifting steps separated by the barrier. For simplicity, the number of barriers is also called the number of steps in the text below.

The 2-D transform is defined as a tensor product of 1-D transforms. Consequently, the transform splits the signal into a quadruple of wavelet coefficients. Therefore, the 2-D DWT is described by 4×4 matrices. See Section Appendix for details. Following the pioneering paper of Mallat [10], the 1-D transforms are applied in both directions sequentially. By its nature, this scheme can be referred to as the separable convolution. The calculations in a single direction are performed in a single step. This means two steps for the two dimensions. The scheme can formally be described as

$$\mathbf{N}^V \,\middle|\, \mathbf{N}^H \,\middle|,$$

where $\mathbf{N}^H$ and $\mathbf{N}^V$ are 1-D transforms in horizontal and in vertical direction. For the well-known Cohen-Daubechies-Feauveau (CDF) wavelet with 9/7 samples, such as used in the JPEG 2000 standard, these matrices are graphically illustrated in Figure 1. Here, only the horizontal part is shown. Particularly, the filters in the figure are of sizes 9 and 7 taps. The 🔴, 🟢, 🔵, and ⚫ circles represent the quadruple of wavelet coefficients. Figures shown are for illustration purpose only.

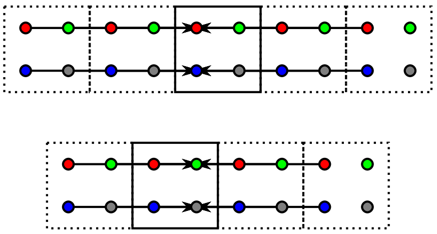


Figure 1: Horizontal part of the separable convolution scheme for the CDF 9/7 wavelet. Two appropriately chosen pairs of matrix rows are depicted in separate subfigures. The arrows are pointing to the destination operand and denote a multiply–accumulate operation, with multiplication by a real constant. The arrows in the same row overlap.

Another scheme used for 2-D transform is the separable lifting. Similarly to the previous case, the predict and update lifting steps can be applied in both directions sequentially. Moreover, horizontal and vertical steps can be arbitrarily interleaved thanks to the linear nature of the filters. Therefore, the scheme is defined as

$$\mathrm{S}_\mathrm{U}^V \,|\, \mathrm{S}_\mathrm{U}^H \,|\, \mathrm{T}_\mathrm{P}^V \,|\, \mathrm{T}_\mathrm{P}^H \,|,$$

wherein the predict steps T always precede the update steps S. The above mapping corresponds to a single P and U pair of lifting steps. For multiple pairs, the scheme is separately applied to each such pair. In order to describe 2-D matrices, the lifting steps must be extended into two dimensions as

$$\begin{bmatrix}\mathrm{G}\\ \mathrm{G}^*\end{bmatrix} = \begin{bmatrix}G\ (z_m, z_n)\\ G^*(z_m, z_n)\end{bmatrix} = \begin{bmatrix}G(z_m)\\ G(z_n)\end{bmatrix}.$$

Then, the individual steps are defined as

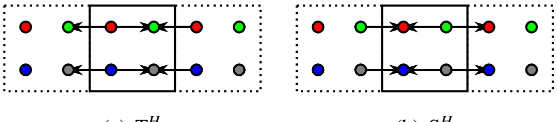
$$\mathrm{T}_\mathrm{P}^H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ \mathrm{P} & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & \mathrm{P} & 1 \end{bmatrix},$$

$$\mathrm{T}_\mathrm{P}^V = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ \mathrm{P}^* & 0 & 1 & 0 \\ 0 & \mathrm{P}^* & 0 & 1 \end{bmatrix},$$

$$\mathrm{S}_\mathrm{U}^H = \begin{bmatrix} 1 & \mathrm{U} & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \mathrm{U} \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$$\mathrm{S}_\mathrm{U}^V = \begin{bmatrix} 1 & 0 & \mathrm{U}^* & 0 \\ 0 & 1 & 0 & \mathrm{U}^* \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

For the CDF wavelets, the matrices are also illustrated in Figure 2, again showing the horizontal part only.



(a) T_P^H (b) S_U^H

Figure 2: The horizontal part of the separable lifting scheme for the CDF wavelets.

3 RELATED WORK

This section briefly reviews papers that motivated our research. So far, several papers have compared the performance of the separable lifting and separable convolution schemes on GPUs. Especially, Tenllado *et al.* [15] compared these schemes on GPUs using pixel shaders. The authors mapped data to 2-D textures, constituted by four floating-point elements. They concluded that the separable convolution is more efficient than the separable lifting scheme in most cases. They further noted that fusing several consecutive kernels might significantly speed up the execution, even if the complexity of the resulting fused pixel program is higher.

Kucis *et al.* [8] compared the performance of several recently published schedules for computing the 2-D DWT using the OpenCL framework. All of these schedules use separable schemes, either the convolution or lifting. In more detail, the work compares a convolution-based algorithm proposed in [5] against several lifting-based methods [2, 16] in the horizontal part of the transform. The authors concluded that the lifting-based algorithm of Blazewicz *et al.* [2] is the fastest method. Furthermore, Laan *et al.* [16] compared the performance of their separable lifting-based method against a convolution-based method. They concluded that the lifting is the fastest method. The authors also compared the performance of implementations in CUDA and pixel shaders, based on the work of Tenllado [15]. The CUDA implementation proved to be the faster choice. In this regard, the authors noted that a speedup in CUDA occurs because the CUDA effectively makes use of on-chip memory. This use is not possible in pixel shaders, which exchange the data using off-chip memory. Other important separable approaches can be found in [11, 6, 13, 12].

This paper is based on the previous works in [1, 9]. In those works, we introduced several non-separable schemes for calculation of 2-D DWT. However, we have not considered important structures, such as polyconvolutions. We contribute this consideration with this paper. Moreover, differences and similarities between the separable schemes and their non-separable counterparts are homogeneously discussed here. All these schemes are also thoroughly analyzed and evaluated.

Considering the present papers, we see that a possible fusion of separable parts into new non-separable structures is not considered. Therefore, we investigate on this promising technique in the following sections.

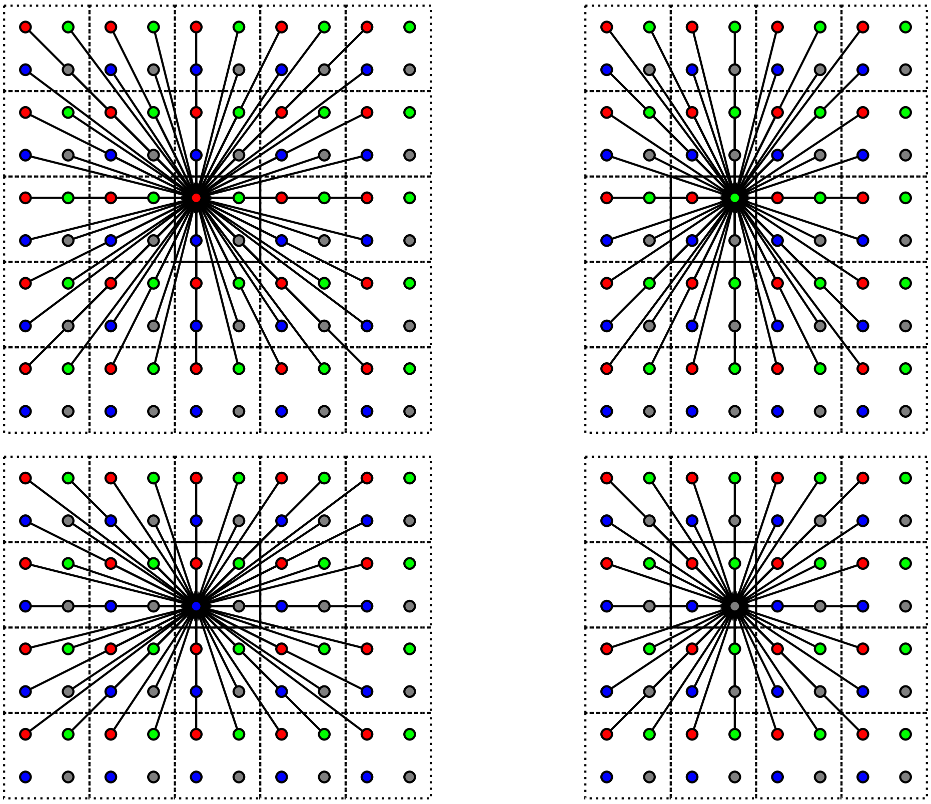


Figure 3: Non-separable convolution scheme for the CDF 9/7 wavelet. The individual rows of N are depicted in separate subfigures. The sizes are from top to bottom and left to right: 9 × 9, 7 × 9, 9 × 7, 7 × 7.

4 PROPOSED SCHEMES

As stated above, the existing approaches did not study the possibility of a partial fusion of lifting polyphase matrices. This section presents three alternative non-separable schemes for the calculation of the 2-D transform. The contribution of this paper starts with this section. To avoid confusion, please note that the proposed schemes compute the same values as the original ones.

The non-separable convolution scheme is a counterpart to the separable convolution. Unlike the separable scheme, all horizontal and vertical calculations are performed in a single step

$$\mathbf{N}\,\big|,$$

where $\mathbf{N} = \mathbf{N}^V\mathbf{N}^H$ is a product of 1-D transforms in horizonal and vertical directions. The drawback of this scheme is that it requires the highest number of arithmetic operations. For the CDF 9/7 wavelet, the matrix is graphically illustrated in Figure 3. Here, the 2-D filters are of sizes 9 × 9, 7 × 9, 9 × 7, and 7 × 7. These sizes make the calculation computationally demanding. Aside from the GPUs, this approach was earlier discussed in Hsia *et al.* [7].

In order to reduce computational complexity, it would be a good idea to construct some smaller non-separable steps. Indeed, the non-separable convolution can be broken into smaller units, referred here to as the non-separable polyconvolutions. For a single pair of lifting steps, the scheme follows from the mapping

$$\mathrm{N}_{\mathrm{P,U}}\,\big|,$$

where

$$\mathrm{N}_{\mathrm{P,U}} = \begin{bmatrix} \mathrm{V^*V} & \mathrm{V^*U} & \mathrm{U^*V} & \mathrm{U^*U} \\ \mathrm{V^*P} & \mathrm{V^*} & \mathrm{U^*P} & \mathrm{U^*} \\ \mathrm{P^*V} & \mathrm{P^*U} & \mathrm{V} & \mathrm{U} \\ \mathrm{P^*P} & \mathrm{P^*} & \mathrm{P} & 1 \end{bmatrix}$$

and $\mathrm{V} = \mathrm{PU} + 1$. For the CDF wavelets, the scheme is graphically illustrated in Figure 4. In this case, the employed filters are of sizes 5 × 5, 3 × 5, 5 × 3, and 3 × 3. Note that only half of the operations are required specifically for the CDF 9/7 wavelet, compared to the non-separable convolution. For the sake of completeness, it should be pointed out that it is also possible to formulate the separable polyconvolution scheme. In our experiments, this one was however not proven to be useful concerning the performance.

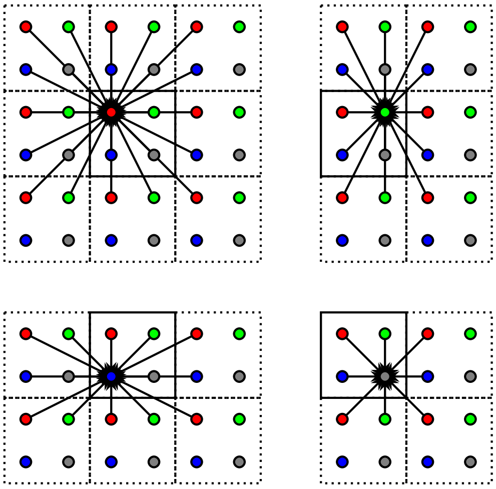


Figure 4: Non-separable polyconvolution scheme for the CDF wavelets. The individual rows of N are depicted in separate subfigures.

By combining the corresponding horizontal and vertical steps of the separable lifting scheme, the non-separable lifting scheme is formed. The number of operations has slightly been increased. The scheme consists of a spatial predict and spatial update step, thus two steps in total for each pair of the original lifting steps. Formally, for each pair of P and U, the scheme follows from

$$\mathrm{S_U} \,\big|\, \mathrm{T_P} \,\big|,$$

where

$$\mathrm{T_P} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ \mathrm{P} & 1 & 0 & 0 \\ \mathrm{P}^* & 0 & 1 & 0 \\ \mathrm{PP}^* & \mathrm{P}^* & \mathrm{P} & 1 \end{bmatrix},$$

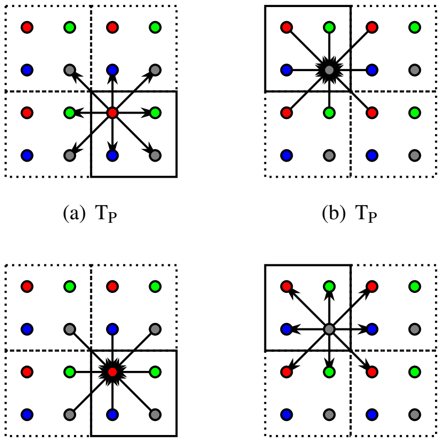
$$\mathrm{S_U} = \begin{bmatrix} 1 & \mathrm{U} & \mathrm{U}^* & \mathrm{UU}^* \\ 0 & 1 & 0 & \mathrm{U}^* \\ 0 & 0 & 1 & \mathrm{U} \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Note that the spatial filters in PP* and UU* may be computationally demanding, depending on their sizes. However, the situation is always better than in the previous two cases. For the CDF wavelets, the scheme is graphically illustrated in Figure 5.

5 OPTIMIZATION APPROACH

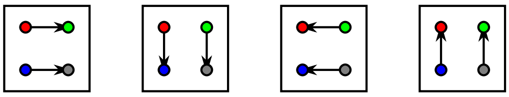
This section presents an optimization approach that reduces the number of operations, while the number of steps remains unaffected. Such an approach was not covered in existing studies.

Regardless of the underlying platform, an important observation can be made. A very special form of the operations guarantees that the processing units never access the results belonging to their neighbors. These operations comprise only constants. Since the convolution is a linear operation, the polynomials can be pulled out of the original matrices, and calculated in a different step. Formally, the original polynomials are split as $\mathrm{P} = \mathrm{P}_0 + \mathrm{P}_1$ and $\mathrm{U} = \mathrm{U}_0 + \mathrm{U}_1$. The P_0 and U_0 are constant. As a next step, the P_0 and U_0 are substituted into the separable lifting scheme. The separable lifting scheme was chosen because it has the lowest number of operations. This part is illustrated in Figure 6. In contrast, the P_1 and U_1 are kept in original schemes. These two steps are then computed without any barrier. The observation is further exploited to adapt schemes for a particular platform.



(a) $\mathrm{T_P}$ (b) $\mathrm{T_P}$ (c) $\mathrm{S_U}$ (d) $\mathrm{S_U}$

Figure 5: Non-separable lifting scheme for the CDF wavelets.



(a) $\mathrm{T}^H_{\mathrm{P}_0}$ (b) $\mathrm{T}^V_{\mathrm{P}_0}$ (c) $\mathrm{S}^H_{\mathrm{U}_0}$ (d) $\mathrm{S}^V_{\mathrm{U}_0}$

Figure 6: Separable lifting scheme with the polynomials P_0 and U_0.

Now, the improved schemes for the shaders and OpenCL are briefly described. These schemes exploit the above-described observation with the polynomials P_0 and U_0 . On recent GPUs, OpenCL schemes also omit memory barriers due to the SIMD-32 architecture. Note that the non-separable polyconvolution scheme makes sense only when $K > 1$, which is the case of the CDF 9/7 wavelet. Implementations in the pixel shaders map input and output data to 2-D textures. There is no possibility to retain some results in registers, and the results are exchanged through textures in off-chip memory. Considering the OpenCL implementations, a data format is not constrained. The image is divided into overlapping blocks and on-chip memory shared by all threads in a block is utilized to exchange the results. Additionally, some results are passed in registers.

This paper explores the performance for three frequently used wavelets, namely, CDF 5/3, CDF 9/7 [3], and DD 13/7 [14]. Their fundamental properties are listed in Table 1: number of steps and arithmetic operations. Note that the number of operations is commonly proportional to a transform performance. Additionally, the number of steps correspond to the number of synchronizations on parallel architectures, which also form a performance bottleneck.

Table 1: The total number of steps and arithmetic operations for the optimized schemes.

(a) CDF 5/3

scheme		steps	operations	
			OpenCL	shaders
separable	convolution	2	20	22
separable	lifting	4	16	16
non-separable	convolution	1	23	39
non-separable	lifting	2	18	18

(b) CDF 9/7

scheme		steps	operations	
			OpenCL	shaders
separable	convolution	2	56	58
separable	polyconv.	4	40	56
separable	lifting	8	32	32
non-separable	convolution	1	152	200
non-separable	polyconv.	2	46	62
non-separable	lifting	4	36	36

(c) DD 13/7

scheme		steps	operations	
			OpenCL	shaders
separable	convolution	2	60	60
separable	lifting	4	32	32
non-separable	convolution	1	203	228
non-separable	lifting	2	50	50

6 EVALUATION

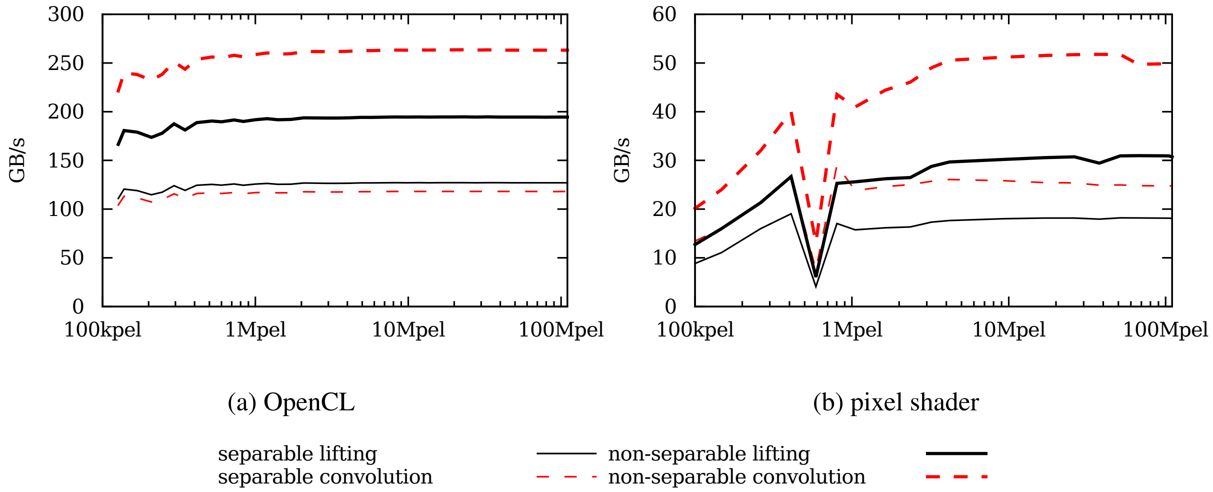
The experiments in this paper were performed on GPUs of the two biggest vendors NVIDIA and AMD using the OpenCL and pixel shaders. In these experiments, only a transform performance was measured, usually in gigabytes per second (GB/s). The host system does not help in the calculation, i.e. with respect to padding or pre/post-processing. Results for only two GPUs are shown for the sake of brevity: AMD Radeon HD 6970 and NVIDIA Titan X. Their technical parameters are summarized in Table 2.

Table 2: Specifications of the evaluated GPUs.

label	AMD 6970	NVIDIA Titan X
model	Radeon HD 6970	Titan X (Pascal)
multiprocessors	24	28
total processors	1 536	3 584
processor clock	880 MHz	1 417 MHz
performance	2 703 GFLOPS	10 157 GFLOPS
memory clock	1 375 MHz	2 500 MHz
bandwidth	176 GB/s	480 GB/s
on-chip memory	32 KiB	96 KiB

The implementations were created using the DirectX HLSL and OpenCL. The HLSL implementation is used on the NVIDIA Titan X, whereas the OpenCL implementation on the AMD 6970. The results of the performance comparison are shown in Figures 7, 8, and 9. The value on the x-axis is the image resolution in kilo/megapixels (kpel or Mpel). Except for the convolutions for the DD 13/7 wavelet, the non-separable schemes always outperform their separable counterparts. For CDF wavelets, having short lifting filters, the non-separable (poly)convolutions have a better performance than the non-separable lifting scheme. Unfortunately, for the DD 13/7 wavelet, which is characterized by a high number of operations in lifting filters, the results are not conclusive. Considering the implementation in pixel shaders, similar results were also achieved on other GPUs, including NVIDIA unified architectures and AMD GPUs based on Graphics Core Next (GCN) architecture. Whereas for the OpenCL implementation, the non-separable schemes are only proved to be useful for very long instruction word (VLIW) architectures.

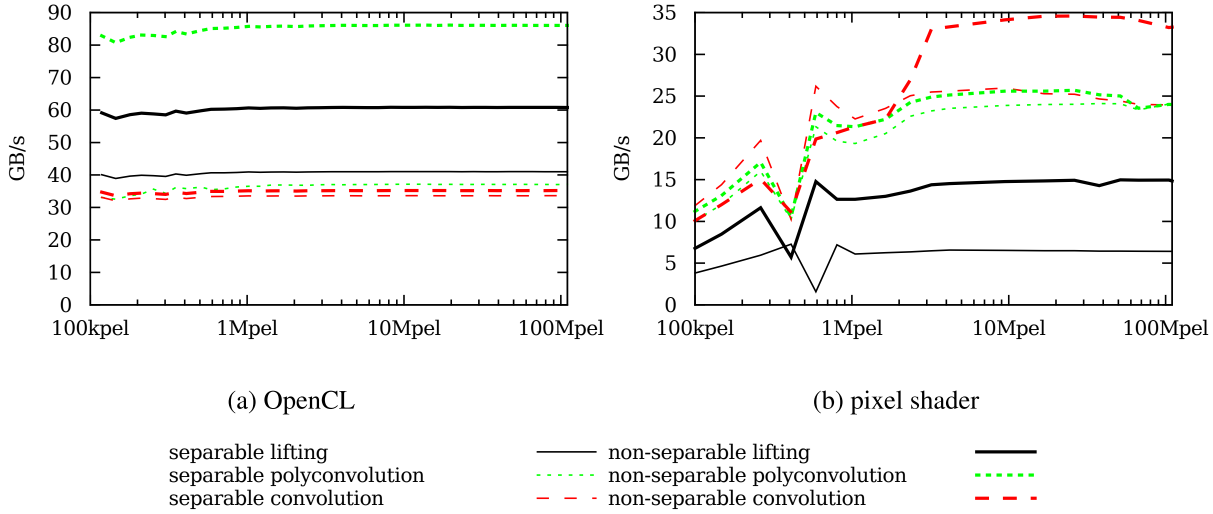
Looking at the experiments with the pixel-shader implementations, some transients can be seen at the beginning of the plots (in lower 2 Mpel region). We concluded that these transients are caused by a suboptimal use of cache system, or alternatively by some overhead made by the graphics API. It should be interesting to show some measures provided by an OpenCL profiler. Our profiling revealed that the implementations exhibit only an occupancy 95.24 %. This occupancy is caused by making use of 256 threads in OpenCL work groups and due to maximal number 1344 of threads in multiprocessors (256 times 5 work groups is 1280 out of 1344).



(a) OpenCL

(b) pixel shader

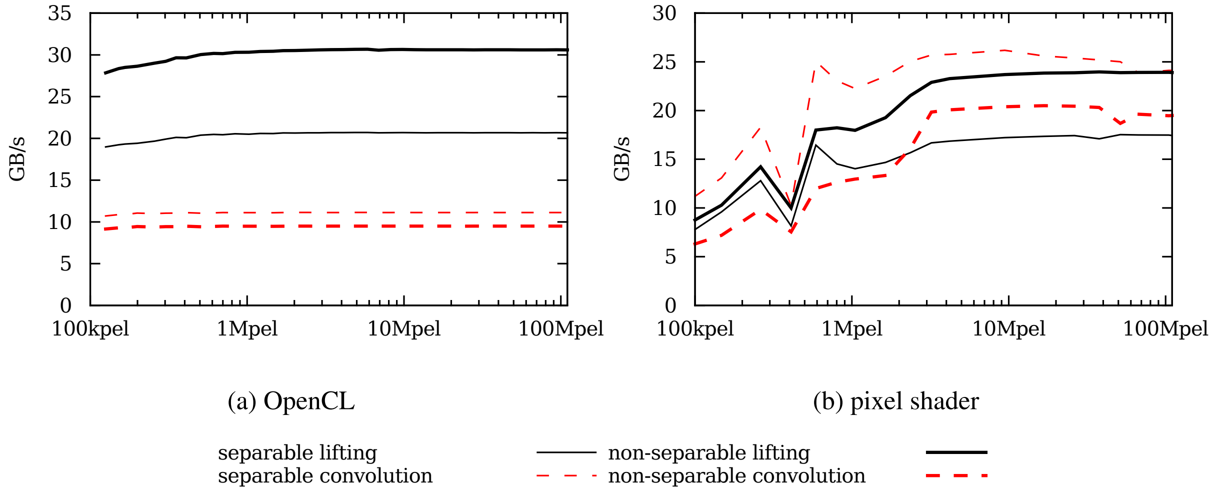
Figure 7: Performance for the CDF 5/3 wavelet.



(a) OpenCL

(b) pixel shader

Figure 8: Performance for the CDF 9/7 wavelet.



(a) OpenCL

(b) pixel shader

Figure 9: Performance for the DD 13/7 wavelet.

7 CONCLUSIONS

This paper presented and discussed several non-separable schemes for the computation of the 2-D discrete wavelet transform on parallel architectures, exemplarily on modern GPUs. As an option, an optimization approach leading to a reduction in the number of operations was presented. Using this approach, the schemes were adapted on the OpenCL framework and pixel shaders. The implementations were then evaluated using GPUs of the two biggest vendors. Considering OpenCL, the schemes exploit features of recent GPUs, such as warping. For CDF wavelets, the non-separable schemes exhibit a better performance than their separable counterparts on both the OpenCL and pixel shaders.

In the evaluation, we reached the following conclusions. Fusing several consecutive steps of the schemes might significantly speed up the execution, irrespective of their higher complexity. The non-separable schemes outperform their separable counterparts on numerous setups, especially considering the pixel shaders. All of the schemes are general and they can be used on any discrete wavelet transform. In future work, we plan to focus on general-purpose processors and multi-scale transforms.

Acknowledgements

This work has been supported by the Ministry of Education, Youth and Sports from the National Programme of Sustainability (NPU II) project IT4Innovations excellence in science (no. LQ1602), and the Technology Agency of the Czech Republic (TA CR) Competence Centres project V3C – Visual Computing Competence Center (no. TE01020415).

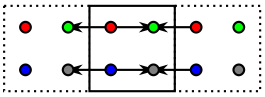
APPENDIX

For readers who are not familiar with signal-processing notations, a relationship between polyphase matrices and data-flow diagrams is explained here. The 2-D discrete wavelet transform divides the image into four polyphase components. Therefore, the 4×4 matrices of Laurent polynomials are used to describe the 2-D discrete wavelet transform. These matrices are commonly referred to as the polyphase matrices. The Laurent polynomials correspond to 2-D FIR filters, that define the transform. In most cases, the transform is described using a sequence of such matrices. One particular matrix thus defines a step of calculation in this case.

For example, the matrix

$$\mathrm{T}_{\mathrm{P}}^{H} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ \mathrm{P} & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & \mathrm{P} & 1 \end{bmatrix}$$

maps four polyphase components to another four components, while using two 2-D FIR filters represented by the polynomials P. Moreover, when we substitute a particular polynomial, say $P(z) = -1/2(1+z^{-1})$, into the matrix, the mapping gets a specific shape. Such a substitution illustrated by the data-flow diagram in Figure 10. The solid arrows correspond to multiplication by $-1/2$ along with subsequent summation.



(a) $\mathrm{T}_{\mathrm{P}}^{H}$

Figure 10: Visual representation of the polyphase matrix. The four polyphase components are represented by color circles.

REFERENCES

[1] Barina, D., Kula, M., and Zemcik, P. Parallel wavelet schemes for images. *Journal of Real-Time Image Processing*, in press. doi: 10.1007/s11554-016-0646-3.

[2] Blazewicz, M., Ciznicki, M., Kopta, P., Kurowski, K., and Lichocki, P. *Two-Dimensional Discrete Wavelet Transform on Large Images for Hybrid Computing Architectures: GPU and CELL*, pages 481–490. Springer, 2012. ISBN 978-3-642-29737-3. doi: 10.1007/978-3-642-29737-3_53.

[3] Cohen, A., Daubechies, I., and Feauveau, J.-C. Biorthogonal bases of compactly supported wavelets. *Communications on Pure and Applied Mathematics*, 45(5):485–560, 1992. ISSN 1097-0312. doi: 10.1002/cpa.3160450502.

[4] Daubechies, I. and Sweldens, W. Factoring wavelet transforms into lifting steps. *Journal of Fourier Analysis and Applications*, 4(3):247–269, 1998. ISSN 1069-5869. doi: 10.1007/BF02476026.

[5] Galiano, V., Lopez, O., Malumbres, M. P., and Migallon, H. Improving the discrete wavelet transform computation from multicore to GPU-based algorithms. In *Int. Conf. on Computational and Mathematical Methods in Science and Engineering*, pages 544–555, June 2011. ISBN 978-84-614-6167-7.

[6] Galiano, V., Lopez, O., Malumbres, M., and Migallon, H. Parallel strategies for 2D discrete wavelet transform in shared memory systems and GPUs. *The Journal of Supercomputing*, 64(1): 4–16, 2013. ISSN 0920-8542. doi: 10.1007/s11227-012-0750-5.

[7] Hsia, C. H., Guo, J. M., Chiang, J. S., and Lin, C. H. A novel fast algorithm based on smdwt for visual processing applications. In *IEEE International Symposium on Circuits and Systems*, pages 762–765, May 2009. doi: 10.1109/ISCAS.2009.5117860.

[8] Kucis, M., Barina, D., Kula, M., and Zemcik, P. 2-D discrete wavelet transform using GPU. In *International Symposium on Computer Architecture and High Performance Computing Workshop*, pages 1–6. IEEE, Oct. 2014. ISBN 978-1-4799-7014-8. doi: 10.1109/SBAC-PADW.2014.13.

[9] Kula, M., Barina, D., and Zemcik, P. New non-separable lifting scheme for images. In *IEEE International Conference on Signal and Image Processing*, pages 292–295. IEEE, 2016. ISBN 978-1-5090-2375-2. doi: 10.1109/SIPROCESS.2016.7888270.

[10] Mallat, S. A theory for multiresolution signal decomposition: the wavelet representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(7):674–693, 1989. ISSN 0162-8828. doi: 10.1109/34.192463.

[11] Matela, J. GPU-based DWT acceleration for JPEG2000. In *Annual Doctoral Workshop on Mathematical and Engineering Methods in Computer Science*, pages 136–143, 2009. ISBN 978-80-87342-04-6.

[12] Quan, T. M. and Jeong, W.-K. A fast discrete wavelet transform using hybrid parallelism on GPUs. *IEEE Transactions on Parallel and Distributed Systems*, 27(11):3088–3100, Nov. 2016. ISSN 1045-9219. doi: 10.1109/TPDS.2016.2536028.

[13] Song, C., Li, Y., Guo, J., and Lei, J. Block-based two-dimensional wavelet transform running on graphics processing unit. *IET Computers Digital Techniques*, 8(5):229–236, Sept. 2014. ISSN 1751-8601. doi: 10.1049/iet-cdt.2013.0141.

[14] Sweldens, W. The lifting scheme: A custom-design construction of biorthogonal wavelets. *Applied and Computational Harmonic Analysis*, 3 (2):186–200, 1996. ISSN 1063-5203. doi: 10.1006/acha.1996.0015.

[15] Tenllado, C., Setoain, J., Prieto, M., Pinuel, L., and Tirado, F. Parallel implementation of the 2D discrete wavelet transform on graphics processing units: Filter bank versus lifting. *IEEE Transactions on Parallel and Distributed Systems*, 19(3):299–310, 2008. ISSN 1045-9219. doi: 10.1109/TPDS.2007.70716.

[16] van der Laan, W. J., Jalba, A. C., and Roerdink, J. B. T. M. Accelerating wavelet lifting on graphics hardware using CUDA. *IEEE Transactions on Parallel and Distributed Systems*, 22(1):132–146, Jan. 2011. ISSN 1045-9219. doi: 10.1109/TPDS.2010.143.

Micro-expression detection using Integral Projections

Hua Lu Kidiyo Kpalma Joseph Ronsin

Université Bretagne Loire
UMR 6164,F-35708, Rennes, France

hua.lu@insa-rennes.fr kidiyo.kpalma@insa-rennes.fr Joseph.ronsin@insa-rennes.fr

ABSTRACT

Automatic detection of micro-expression from a video is the first step in the micro-expression analysis. In this paper, we present a method addressing the micro-expression detection problem based on the differences in the Integral Projection (IP) of sequential frames. The method can detect the temporal location of the micro-expression. It involves an observation of the Chi-squared distance of the IP to measure the difference between frames. The main advantage of using IP for micro-expression detection is its low computation cost, which brings an important merit in real-time application. To evaluate our method, experiments are completed on two micro-expression databases namely CASME and CASME II. Results on these two datasets show that the proposed method obtains positive promising results with much less computation time against state-of-the-art methods.

Keywords

Micro-expression detection, Integral Projection, Chi-squared distance

1 INTRODUCTION

As a non-verbal behavior, the facial expression plays an important role in our daily life. People can express their feelings by making facial expressions and can also communicate with others by reading facial expressions. During the past two decades, the macro-expression analysis has been paid huge attention in many fields. Psychologists studied the human psychology conveyed by the changing facial expression and computer scientists relied on the digital process to analyze the expression. With the progress of technology, automatic macro-expression detection and recognition can be achieved in real-time and has been successfully applied into business [Del15]. Most of the expressions which are called macro-expressions can be easily observed by naked eyes. However, researchers found some expressions appearing and fading quickly and often easily neglected by naked eyes, these are named as micro-expressions.

Approaches addressing the issues related to micro-expression have been considerably studied in the past 50 years [Hag66]. Recently, an increasing attention has been paid to micro-expression detection and recognition due to its potential application in medicine, business and many other fields, such as catching lies during diagnose, negotiation or interrogation [Ekm69]. Compared to macro-expression, a micro-expression lasts only 1/25s to 1/5s, and moreover, its subtle appearance in part of the face makes naked eyes-based detection and recognition difficult to achieve. Thus, computer vision analysis offers a potential solution.

Micro-expression approaches in computer vision area consist of detecting and classifying them from videos. This inspires a series of approaches on micro-expression analysis integrating computer-aided techniques. Most works of the micro-expression analysis concentrate on the classification step [Hua15], [Liu15], and few works have been devoted to the detection, which is the foundation of the analysis. So far, several methods have been developed for this detection, such as method based on 2D/3D histogram of oriented gradients, local binary patterns and optical flow which will be presented and then used for comparison with the proposed method.

Polikovsky et al. [Pol09] divided the face into different facial regions and used the 3D histogram of oriented gradients descriptor (3D HOG) for feature extraction. The recognition applied the k-means method to cluster the extracted features of each region. The results showed good performance rates (all over 80 %) in the regions of the forehead, between the eyes and lower nose. However, the experiments were conducted in a small dataset that only contained 13 posed micro-expressions instead of the spontaneous micro-expressions. Davison et al. [Dav15] used 2D histogram of oriented gradients (HOG) to extract the features of each frame. The Chi-squared distance measure was applied to compute dissimilarity between the sequence frames. However, in paper [Dav15], all detected micro-movements up to 100 frames (200 fps) were classified as true positive including blinks and the eye gaze, without comparing the ground truth of the micro-expression.

Moilanen et al. [Moi14] adopted local binary patterns (LBP) to extract the features from the divided blocks of the face. The method relied on calculating the dissimilarity of features for each block by using Chi-squared distance. The detection experiments were conducted on the spontaneous facial micro-expression datasets in order to solve the problem in practice.

Shreve et al. [Shr14] developed a method for the segmentation of macro- and micro-expression frames by calculating the deformation of facial skin using optical flow (OF). The optical flow is a well-known motion estimation technology and can well spot the subtle movement, but its calculation costs expensively computation time.

Besides, some papers addressing the problem of the detection by training a model to determine if a sequence does or does not contain a micro-expression. Pfister et al. [Pfi11] extracted spatio-temporal local texture features from video sequences and used machine learning algorithms (SVM, MKL, RF) for classification. Xia et al. [Xia16] utilized an adaboost model to compute the initial probability for each frame and the correlation between frames in order to generate a random walk (RW) model. The random walk model was used to calculate the deformation correlation between frames and provide the probability of having micro-expressions in a sequence.

Instead of developing a training model, we propose a new micro-movement detection method by invoking the IP as a feature descriptor to characterize changes in the divided blocks of the face. The IP feature is extracted on each individual block. The Chi-squared distance is used to measure the IP feature dissimilarity between frames so as to observe for possible micro-expression in the frame sequence. The proposed method is evaluated on two widely used datasets through experimental comparison with some popular feature extractors such as the OF, LBP and HOG. The proposed method is an unsupervised model. One of the main advantages of our model is its computation complexity: our model can obtain better or comparable results than the existing models using the OF, LBP and HOG, but requiring much less computation time.

The rest of the paper is organized as follows: Section 2 outlines the background on micro-expression and the IP computation. Section 3 describes the procedure of micro-movement detection and experimental results are discussed in Section 5. Finally Section 6 concludes the paper with the discussions.

2 BACKGROUND

2.1 Micro-expression

A micro-expression is a brief, spontaneous expression, which reveals the true feeling that people try to hide and suppress. Regardless of whether they are macro or micro, facial expressions are dynamic temporal process that match the time and duration of facial deformations and are described with three important concepts: onset, apex and offset [Bet12], [Wei09]. The onset is the point at which the expression starts to show up, the apex is the instant when the deformation of the expression reaches the peak and the offset represents the instant when the expression fades away. Hence, the micro-expression detection is a temporal segmentation of videos, which includes locating the micro-expression appearance instant and providing the duration between the onset and offset. Fig. 1 present an image sequence of the micro-expression which is labeled by 'disgust' in CASME II. Due to the limit space, five frames are presented including 1th(onset), 15th, 29th (apex), 57th and 86th (offset).

Obviously, the duration of the micro-expression is variable. In general, the micro-expression lasts from 1/25s to 1/5s [Ekm69] but other papers show the lasting time of micro-expressions can extend to 1/2s [Mat00], [Mat11]. In our method, the duration from 1/25s to 1/2s will be considered as the guideline.

2.2 Integral Projection

Due to the difficulty for people to read micro-expressions, it is necessary to find appropriate methods for catching subtle and rapid changes of the face. The Integral Projection is presented in the following and it holds as a useful technique for the extraction of facial features. As IP can be extremely effective in determining the position of features, Brunelli et al. [Bru93] applied IP on the human face recognition. In a recent work [Hua15], a combinational method of the IP and LBP was chosen for micro-expression recognition thanks to its ability for providing the shape property of facial images.

IP is a simple and rapid feature extraction method which can reduce the 2D Image features to a simply 1D data. Let $\Omega \subset \mathbb{R}^2$ be the image domain and $I : \Omega \times D \rightarrow \mathbb{R}$ be a sequence of gray level images, where $D \subset \mathbb{R}$ is the time space. At each point $(x,y) \in \Omega$ and at time t, the intensity value is denoted by $I(x,y,t)$,

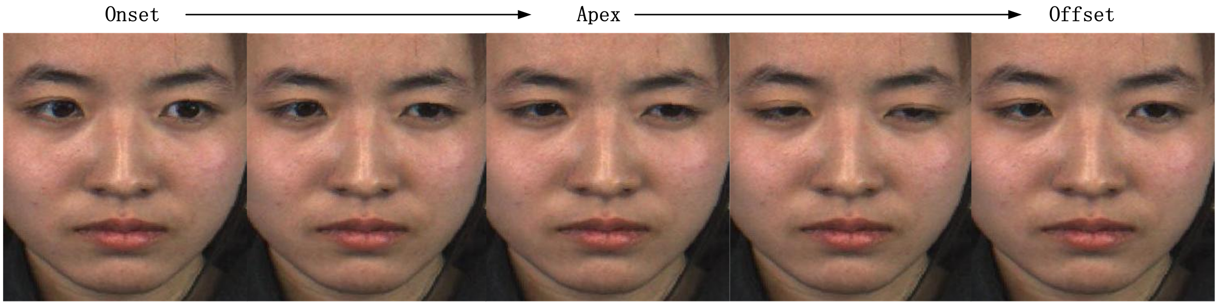


Figure 1: An example of a micro-expression sequence. Five frames are presented including 1th(onset), 15th, 29th (apex), 57th and 86th (offset).

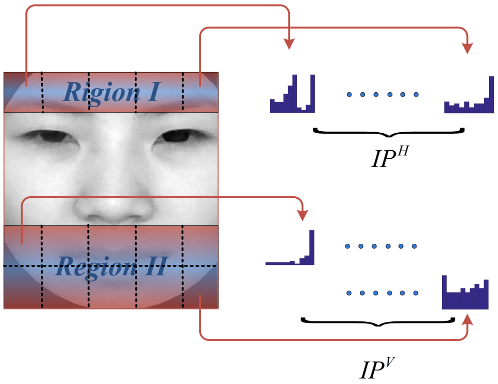


Figure 2: An example of the IP histograms. Plots in the first (resp. second) row correspond to the horizontal (resp. vertical) IP function from each block.

and the typical formula of the IP function can be expressed as:

$$IP_t^H(x) = \frac{1}{y_2 - y_1} \int_{y_1}^{y_2} I(x,y,t)dy, \quad (1)$$

$$IP_t^V(y) = \frac{1}{x_2 - x_1} \int_{x_1}^{x_2} I(x,y,t)dx, \quad (2)$$

where IP_t^H and IP_t^V are the horizontal and vertical integral projection vectors in the rectangle $[x_1,x_2] \times [y_1,y_2]$ at time t, respectively. Fig. 2 shows examples of the IP histogram curves (horizontal and vertical).

3 PROPOSED METHOD

The flow chart of the proposed method is summarized in Fig. 3 and will be detailed in the following steps. Two main parts are presented in this flow chart: one part is the global pre-processing and featuring, and the other is the extraction of the micro-expression. The part I includes tracking, registering, cropping, masking, blocking the face, the IP features extraction, chi-squared distance analysis, thresholding, peak detection. The part II consists of the micro-expression extraction.

3.1 Face tracking and Registration

Determining the existence and locations of faces in each frame of the sequence is the first step for micro-expression detection. The crucial point of this step is the key points detection for cropping the face. In this section, we choose the Supervised Descent method [Xio13] for facial expression points tracking, from which we can obtain 49 facial key points to register and crop the face.

Since the algorithm depends on careful positioning of the face, the alignment step is necessary for the purpose of keeping eyes in horizontal line. By using the facial key points located on the inner eye corners to calculate the angle $\theta \in [0,\pi)$ between the line of the two eyes and a horizontal line, face alignment operation can be performed such that $\theta = 0$.

3.2 Crop, Mask and Divide face into blocks

The nasal spine point is considered as the fixed point for cropping the face. The regions containing inhomogeneous background, clothes, hairs, and eyes which may influence the micro-expression detection results will be removed by a face mask. Thus, one can focus on the regions which only contain useful information. During the process of calculating the IP over the whole face, some important spatial information may be missed due to global merging of observations and hence giving difficulties to identify subtle changes of face. Therefore, in order to obtain the accurate spatial information for the detection of micro-expression, two blocked regions of interest (ROIs) are defined for IP computing: Region I and Region II respectively involving N and $2N$ blocks, as shown in Fig. 2. The number N will be discussed in section 4.1.2. IP can be calculated in each block to locate small movement for micro-expression analysis.

3.3 Feature Extraction Using IP

Once obtained the cropped and blocked face regions, the IP histograms for each block is computed and then fused for the corresponding region. For the blocks in region I, horizontal IP will describe better the change of the facial skin such as the quick movement of the eyebrow. For the blocks in region II, the micro-movement of the mouth will be well featured by the vertical IP.

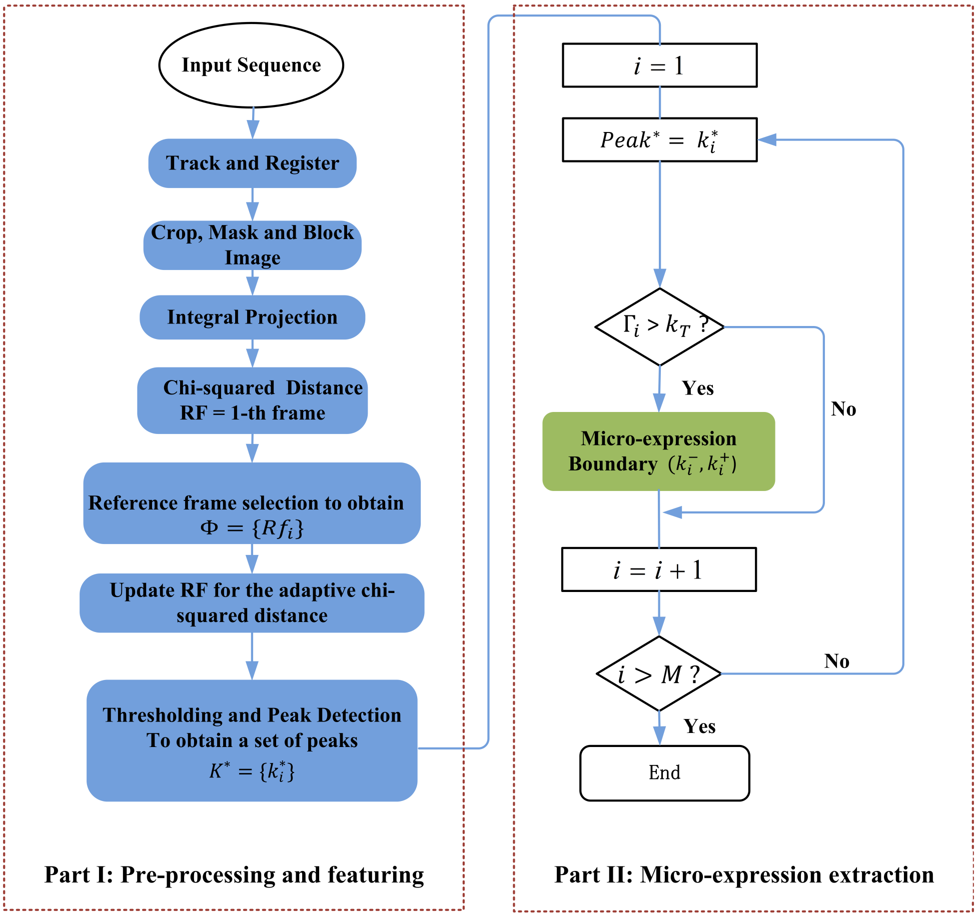

Input Sequence
Track and Register
Crop, Mask and Block Image
Integral Projection
Chi-squared Distance RF = 1-th frame
Reference frame selection to obtain $\Phi = \{Rf_i\}$
Update RF for the adaptive chi-squared distance
Thresholding and Peak Detection To obtain a set of peaks $K^* = \{k_i^*\}$
Part I: Pre-processing and featuring
$i = 1$
$Peak^* = k_i^*$
$\Gamma_i > k_T$?
Yes
No
Micro-expression Boundary (k_i^-, k_i^+)
$i = i + 1$
$i > M$?
No
Yes
End
Part II: Micro-expression extraction

Figure 3: Flow diagram of the proposed algorithm.

Thus, the horizontal IP feature the region I, while the vertical IP feature the region II, as shown in Fig. 2.

3.4 Feature difference analysis

For feature difference analysis, scanning the sequence, subtraction will be performed between a reference frame (RF) and each successive frame denoted as current frame (CF). This reference frame must be a neutral face or onset frame of a temporal facial expression for highlighting differences along the sequence. Differences will be observed from integral projections. IP_t^H and IP_t^V features are extracted from each frame at each block of the two regions, followed by chi-squared distance computation [Moi14] to measure the dissimilarity between the ROIs of the CF and the ROIs of the RF. The chi-squared distance is an efficient method to compute the distance between the features. Given two IP features of P and Q, the chi-squared distance is defined by

$$\chi^2(P,Q) = \frac{1}{2}\sum_{i=1}^{n} \frac{(P(i)-Q(i))^2}{P(i)+Q(i)}, \tag{3}$$

where n is the length of the IP features.

The regions I and II generate two chi-squared distance sequences which are denoted by S_1 and S_2, respectively. The computation of $S_j(j=1,2)$ for the k-th frame can be expressed as

$$S_j(k) = \chi^2(P_0^j, P_k^j) \quad \forall k \in [1, L-1], \tag{4}$$

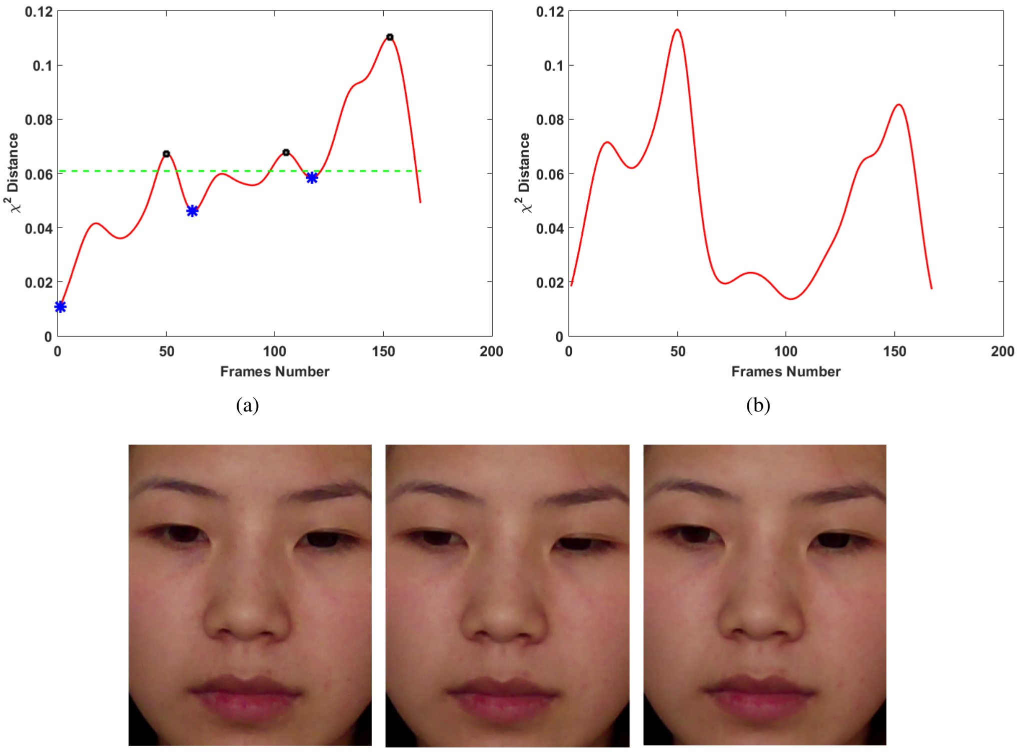
where P_0^j and P_k^j are the IP features of the reference frame and the k-th frame at the regions I $(j=1)$ and II $(j=2)$. The chi-squared distance S used for micro-movement detection is computed by

$$S(k) = \frac{1}{2}(S_1(k)+S_2(k)), \tag{5}$$

which involves the mean values of the normalization of the sequences S_1 and S_2 at the respective location. Normalize the sequences S_1 and S_2 respectively by the values of $\sqrt{\sum_k S_1^2(k)}$ and $\sqrt{\sum_k S_2^2(k)}$.

3.5 Reference frames selection

For very long videos segmentation, it is necessary to select different RFs since taking the first frame as the RF will lead to accumulating errors along the sequence. To solve this problem, a new reference frame selection method is proposed. Before RF selection, one needs to apply low-pass filtering in order to eliminate high frequency details that may influence the result. We give an



(a)

(b)

(c)

Figure 4: An example of the detection of Φ. **(a)** The red curve describes the chi-squared distance S. The mean value of S are denoted by a green line. The collection Ψ and Φ are described by black dots and blue star points, respectively. **(b)** illustrates the curve for the chi-squared distance S after updating the RF from the collection Φ. **(c)** shows the three RF of Φ at 1, 62, 117.

example in Fig. 4a, where we plot the curve (red solid curve) for the chi-squared distance S in Eq. (5) when the first frame is selected as the RF. We can see that local maximums of the values of S get larger along the sequence, which may introduce bias estimation to the threshold value used for the micro-expression detection in Section. 3.6.

To cope with this possible bias, we define Φ as a collection of the reference frame indexes which can be expressed as

$$\Phi = \{Rf_i\}_{1 \le i \le m}, \quad m \in [1, L-1],$$

where m is the total number of the reference frames and Rf_i is the index of the i-th RF in the sequence. L is the total number of frames in the sequence. Let $Rf_1 = 1$ be the first frame of the sequence then the remaining elements of the collection Φ can be detected in the following two steps.

Firstly, apply the peak detection procedure to the chi-squared distance S in Eq. (5) to search for a collection Ψ of frame indexes

$$\Psi = \{\zeta_j\}_{1 \le j \le \tau}, \quad \tau \in [1, L-1],$$

Each element $\zeta_j \in \Psi$ is a local maximizer of the chi-squared distance S. In other words, ζ_j indicates an admissible peak of S such that $S(\zeta)$ is a local maximum value which is larger than the mean of S. We further assume that the elements ζ_j of the collection Ψ admits that $\zeta_i < \zeta_j$, if $i < j$. Secondly, search for the nearest local minimum value from the maximum ζ_j along the positive direction. Each pair of adjacent elements ζ_i, $\zeta_{i+1} \in \Psi$ determines a subsequence of frames, among which a local *minimizer* Rf_i of the computed chi-squared distance S can be obtained. This minimizer is taken as the index of the i-th RF for its notation and it is called Rf_i and $Rf_i \in \Phi$. If there are more than one local minimizer in the subsequence between ζ_i and ζ_{i+1}, we choose the closest one to the frame ζ_{i+1} (in the sense of Euclidean distance of indexes) as the RF.

Starting from the reference frames collection Φ, a new distance sequence is generated by updating the RF. This is done by computing the chi-squared distance between Rf_i and Rf_{i+1} using Rf_i as RF. Fig. 4b is an example of the new chi-squared distance sequence. One can claim that after updating the RF, it is easier to obtain the location of the micro-expression, where the ground truth of the micro-expression given in this example is 39-59 frames. Fig. 4c is an illustration of the detected reference frames which are neutral faces or nearly ones.

In this section, we perform the reference frames selection and the Gaussian smoothing operation on the sequences S to obtain a new adaptive chi-squared distance sequence $S^{'}$ based on the RF collection Φ. The smoothing step aims to remove the noises from the sequences.

3.6 Thresholding and Peak Detection

Once computed the chi-squared distance, it is necessary to use a thresholding method to obtain the location of the micro-expression.

The following steps are applied for the process of the distance sequence $S^{'}$:

1. **Polynomial Fitting.** Apply a second order polynomial fitting operation to the sequence $S^{'}$ by the least square method [Shr11] and generate a fitting function ρ. In Fig. 5a, we demonstrate the plot curve of the function ρ.

2. **Micro-expression Appearance Computation.** Compute a sequence β with the same length of S by subtracting the fitting sequence ρ from the sequence $S^{'}$. All of the negative values of β are set to 0. The expression of β can be found in Eq. (6) and it is illustrated on Fig. 5b.

3. **Peaks Detection.** Apply a peak detection procedure to search for a collection of peaks of the sequence β as the indicators of the appearance of micro-expressions. This peaks detection procedure relies on two threshold values as described in the following.

The polynomial fitting step is able to suppress the cropping errors accumulated over the whole sequence. In step 2, the thresholded sequence β can be computed by thresholding $S^{'}$ with ρ as follows:

$$\beta(k) = \max\left\{S^{'}(k) - \rho(k), 0\right\}, \quad \forall k \in [1, L-1], \quad (6)$$

where ρ is the fitting function and L is the total number of frames in a sequence. The sequence of β involves the information of the existence and location information of the potential micro-expressions.

The peaks detection procedure is carried out dependently of a threshold value T that can be computed by

$$T = \beta_{\text{mean}} + p\,(\beta_{\max} - \beta_{\text{mean}}), \quad (7)$$

where β_{mean} and $\beta_{\max}$ are the corresponding mean and maximum values of the thresholded sequence β. The scalar value $p \in [0,1]$ is a tuning parameter [Moi14]. This procedure plays the crucial role in the entire course of the micro-expression detection. Thus we give a detailed introduction in the following.

We first detect a collection K^* of M admissible peaks points k_i^* from the sequence β in Eq. (6). Each peak point survives in a subsequence $\Gamma_i \subset [1, L-1]$, where L is the length of the processed frames including the reference frame. These subsequences Γ_i can be considered as the neighborhoods of the corresponding peak point. We supposed that each subsequence Γ_i has only one peak point and is disjoint to another, i.e.,

$$\Gamma_i \cap \Gamma_j = \emptyset, \quad \forall i \neq j.$$

The detection of the collection K^* and the subsequences Γ_i can be done in two sub-steps. First of all, a candidate peak point is a local maximizer of the sequence β within the subsequence Γ_i

$$\beta(k_i^*) \geq \beta(k), \quad \forall k \in \Gamma_i.$$

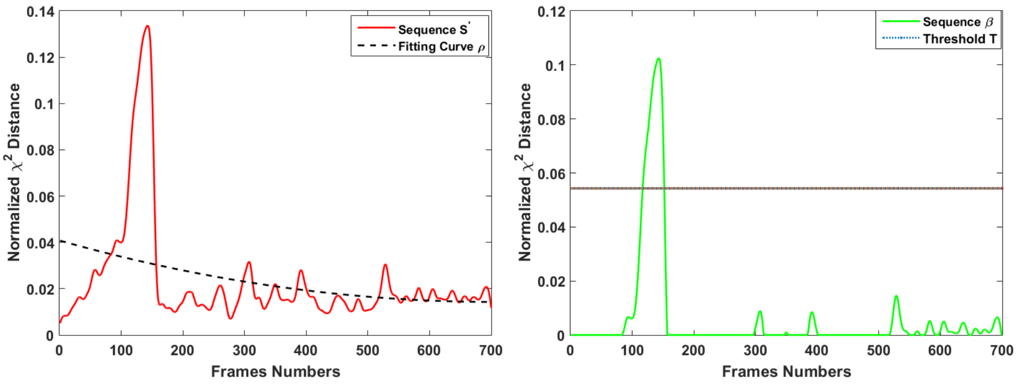
and has a value of β larger than the threshold T. Secondly, we detect the neighborhood Γ_i of this candidate peak point. A subsequence Γ_i can be characterized by the position k_i^* of the candidate peak point and two boundary points k_i^+ and k_i^- such that $\Gamma_i = [k_i^-, k_i^+]$. We search for the position k^+ from the candidate peak point β_i^* along the positive direction till we pass by a point k_* such that $\beta(k_*) < \beta(k_i^*)$, or $\beta(k_*)$ is a local minimum of β, i.e., $\beta(k_*) > \max(\beta(k_* - 1), \beta(k_* + 1))$, where $\alpha > 0$ is a constant value. Similarly, the position k_i^- is determined along the negative direction. In practice, the value of $\beta(k_*)$ is thought as a local minimum if $|\beta(k_*) - \beta(k_* + 1)|$ is small enough. Based on the two sub-steps described above, a candidate peak point k_i^* is admissible if

$$|k_i^+ - k_i^-| > k_T,$$

where k_T is a given threshold value dependent of datasets. Note that the subsequence Γ_i is actually the i-th duration of a micro-expression. The value of $\beta(k_i^*)$ is the i-th value of the peak of the thresholded sequence β. In this step, the values of β at the boundary points k_i^+ and k_i^- are approximately equal to a fraction $\beta(k_i^*)$

$$\beta(k_i^+) \approx \beta(k_i^-) \approx \alpha\beta(k_i^*). \quad (8)$$

In this paper, the constant α determines the detected length of the micro-expression. Fig. 5a illustrates for a video of 700 frames the fitting curve ρ (black color) for $S^{'}$ (red color). The threshold T in Eq. (7) and the sequence β in Eq. (6) used for spotting the micro-expression are shown in Fig. 5b by a horizontal dash line and a green solid curve, respectively. In Fig. 5b, it can be seen from the green curve that a micro-expression is spotted around the frame 143. A duration of $128 - 150$ frames is detected with $\alpha = 0.8$ which will be kept inside the algorithm. Compared with the referenced ground truth of frames $131 - 160$ with the peak frame 142, one can see that the detected starting and ending frames are not exactly the same as those of the ground truth but are very close with long overlapping between both. Based on this observation, it is reasonable to claim that obtained results agree with the ground truth.



(a) (b)

Figure 5: An example of the process of the micro-expression detection. **(a)** illustrates the curve of S is fitted by the polynomial fitting and **(b)** provides the step of locating the micro-expression for thresholding the curve of β by T.

4 EXPERIMENTS

For the evaluation, experiments are conducted on two well-known datasets in micro-expression analysis namely CASME [Yan13] and CASME II [Yan14]. Micro-expressions in these two datasets are elicited spontaneously and labeled with reliable ground truth corresponding to the onset, apex and offset frames which can be used for comparisons in the experiments.

4.1 Datasets and experiment sets

4.1.1 Datasets

The dataset of CASME consists of 195 spontaneous micro-expressions which were selected from more than 1500 elicited facial movements and filmed with two cameras corresponding to two classes (class A and B) at a frame-rate of 60fps. As the lighting condition and resolution of pixels differ between two classes, experiments were carried separately. The videos were recorded in natural light in Class A and in a room with two LED lights in Class B. The resolutions of pixels in Class A and B are 1280×720 and 640×480 pixels, respectively. All samples in CASME database were used in the experiment.

CASME II contains 255 micro-expressions sequences selected among 3000 facial movements. Thirty five participants were recruited with a mean age of 22.03 years in the database. The resolution is 640×480 pixels with a frame rate of 200fps such that they could be recorded more detailed information on the facial muscle movements than CASME. All samples from CASME II dataset were used for the evaluation.

The micro-expression is featured by its rapid movement with respect to the short duration. Thus, the total number of frames of the micro-expression duration is limited. In CASME database, micro-expressions with the duration no more than 500ms or facial expressions lasted more than 500ms but their onset duration less than 250ms were selected. The total duration less than 500ms or onset duration less than 250ms were chosen as the final samples in CSAME II.

4.1.2 Parameters setup

In our experiments, a comparison with methods of the optical flow (OF), local binary patterns (LBP) and histogram of oriented gradients (HOG) is provided. Parameters are set up in the following.

For the OF, optical flow is computed using the MATLAB implementation of Black [Sun10], [Bla96].

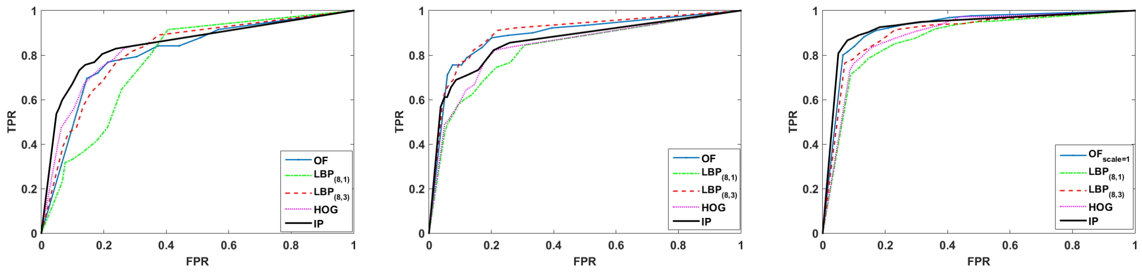
For the LBP, two uniform patterns [Oja96] of $((P,r) = (8,1),(P,r) = (8,3))$ are considered. P corresponds to the number of pixels on the local neighborhood of a circle defined by its radius r.

For the HOG [Dal05a], the histogram angle varies from 0 to π or from 0 to 2π, which corresponds to the 'unsigned' or 'signed' gradient. The number of orientation bins is a segmentation value of histogram angles. Here, 8 orientation bins on 2π angle corresponding to signed gradient are chosen as in [Dav15].

The variable parameter N defines the number of blocks mentioned in Section 3.2 and is set to 5 and α in Eq. (8) is set to 0.8. k_T mentioned in Section 3.6 is set to 2 in CASME dataset and set to 7 in CASME II dataset which corresponds to the minimum duration of the micro-expression.

We give a time window tolerance l to detect positively the appearance of the micro-expression peak. The locations of spotted peaks k_i^* are compared with the provided ground truth, and considered to be true positive if they fall within the frame span of $(onset+l, offset-l)$.

We set parameter $l = 5$ for CASME as discussed in [Moi14], and to $l = 16$ for CASME II same as in [Li15]. As eyes are masked in our experiment, spotted eye blinks are counted as false positives not true positives.



(a) (b) (c)

Figure 6: **a-c**: ROC curves for the datasets of CASME-A, CASME-B and Casme II, respectively.

5 RESULTS

Three indicators are adopted for assessing the performance of the algorithm: The receiver operating characteristic (ROC) curve, the area under the ROC curve (AUC) and processing time. The implementation was tested on an Intel Core i7 computer with 16GB of RAM which was equipped with Matlab 2015a.

The ROC curve is used for spotting performance comparison which is illustrated by plotting the true positive rate (TPR) in y-axis against the false positive rate (FPR) in x-axis. The TPR is defined as the number of frames of correctly spotted micro-expression divided by the total number of the ground truth micro-expression frames in the dataset. The FPR is computed as the number of incorrectly spotted frames divided by the total number of non-micro-expression frames in the database. The TPR is in the vertical axis, and the FPR is in the horizontal axis, and p in Eq. (7) is used as the varying threshold parameter with step size of 0.1.

Figs. 6a to 6c show the ROC curves obtained from CASME-A, CASME-B, CASME II for the 4 methods, respectively. Overall, we can observe that the proposed method achieves better performance than other methods (OF, LBP, HOG) in CASME-A and CASME II datasets. Some points with low FPR in ROC curves are meaningful. For example, our proposed method is able to detect 80% of the micro-expression with 4% FPR in CASME II dataset. $LBP_{(8,3)}$ outperforms $LBP_{(8,1)}$ on three database and provides the best results in CASME-B.

The area under the ROC curve (AUC) summarizes the spotting performance as shown in Table 1. The high values of the AUC means good performance of the method. The AUC results are positive overall and demonstrate that all 4 methods are efficient for spotting micro-expressions. Among two datasets, a better overall performance can be observed that in CASME II. Two reasons can explain this: one is that subjects in CASME dataset often move their head, and another one is that videos are recorded in a different lighting environment in CASME-B leading to the uneven distribution of the lighting in face. In contrast, CASME II contains short video clips at a frame rate of 200fps and no face moving rapidly leading to better detection results.

Dataset	CASME-A	CASME-B	CASME II
OF	0.8092	0.8888	0.9243
HOG	0.8268	0.8378	0.8939
$LBP_{(8,1)}$	0.7716	0.8244	0.8751
$LBP_{(8,3)}$	0.8177	**0.8987**	0.9014
Proposed IP	**0.8480**	0.8617	**0.9289**

Table 1: AUC performance for all datasets

Method	Time per frame(ms)	γ
OF	480	631.58
HOG	121.11	159.35
LBP	35.13	46.22
Proposed IP	**0.76**	**1**

Table 2: Computation time comparison (image size 320 × 260)

Among these methods, the proposed method can perform best except in CASME-B dataset because the IP is sensitive to illumination variance while the LBP is robust to illumination. However, the better performance of our method in CASME-A and CASME II shows that the IP is an efficient feature which can describe the temporal dynamic of the micro-expression. The processing time for different methods is presented in the Table 2. Here a ratio for computational time comparison is defined as:

$$\gamma = \frac{T_{method}}{T_{IP}}, \tag{9}$$

where T_{method} represents the processing time of the OF, HOG and LBP. T_{IP} indicates the processing time of the IP features.

As we can observe in Table 2, the algorithm of the optical flow is extremely slow taking 480ms for one image of 320 × 260 feature extracting. While the proposed method takes only 0.76ms and thus is promising for implementation in real-time process. It is also clear from Table 2 that the integral projections provide a huge reduction in computational time. Compared to the LBP and HOG, our method still globally outperforms them with an advantage in the lower computational complexity.

6 CONCLUSION

The analysis on differences of the integral projection allows detecting micro-movements automatically with a low computation complexity. Experimental results are positive on the datasets CASME-A, CASME-B and CASME II, indicating that this method is capable of catching micro-expressions from videos. To the best knowledge of the authors, this is the fastest method for automatic micro-expression detection and it could be implemented for future real-time detection. During the experiment, it was noticed that large head motions and the illumination variation can cause the miss-detections. In the future, more robust algorithms will be studied for addressing these problems. The motion-based method is accurate at the cost of computation complexity, while appearance-based methods are a bit less efficient in spotting micro-movements but with a fast computation. Thus, a combination of motion- and appearance-based method could bring a solution for improving performance of micro-expression detection.

7 REFERENCES

[Hag66] Haggard, E.A., Isaacs, K.S.: Micromomentary facial expressions as indicators of ego mechanisms in psychotherapy. Methods of research in psychotherapy. pp.154-165, 1966.

[Ekm69] Ekman, P., Friesen, W.V. Nonverbal leakage and clues to deception. Psychiatry. pp.88-106, 1969.

[Hua15] Huang, X., Wang, S.-J., Zhao, G., Piteikainen, M. Facial micro-expression recognition using spatiotemporal local binary pattern with integral projection. Proceedings of the IEEE International Conference on Computer Vision Workshops, pp. 1-9, 2015.

[Liu15] Liu, Y.-J., Zhang, J.-K., Yan, W.-J., Wang, S.-J., Zhao, G., Fu, X. A main directional mean optical flow feature for spontaneous micro-expression recognition. IEEE Transactions on Affective Computing. pp. 299-310, 2015.

[Pol09] POLIKOVSKY, S., KAMEDA, Y., OHTA, Y. Facial micro-expressions recognition using high speed camera and 3D-gradient descriptor. Crime Detection and Prevention (ICDP 2009), pp. 1-6, 2009.

[Dav15] Davison, A.K., Yap, M.H., Lansley, C. Micro-Facial Movement Detection Using Individualised Baselines and Histogram-Based Descriptors. Systems, Man, and Cybernetics (SMC), pp. 1864-1869, 2015.

[Moi14] Moilanen, A., Zhao, G., Pietikainen, M. Spotting rapid facial movements from videos using appearance-based feature difference analysis. 2014 22nd International Conference on Pattern Recognition (ICPR), pp. 1722-1727, 2014.

[Shr14] Shreve, M., Brizzi, J., Fefilatyev, S., Luguev, T., Goldgof, D., Sarkar, S. Automatic expression spotting in videos. Image and Vision Computing. pp.476-486, 2014.

[Bet12] Bettadapura, V. Face expression recognition and analysis: the state of the art. arXiv preprint arXiv:1203.6722. 2012.

[Wei09] Weiss, J.: Ekman, P. Telling Lies : Clues to Deceit in the Marketplace, Politics, and Marriage. New York: Norton. American Journal of Clinical Hypnosis. pp.287-288, 2011.

[Mat00] Matsumoto, D., LeRoux, J., Wilson-Cohn, C., Raroque, J., Kooken, K., Ekman, P., Yrizarry, N., Loewinger, S., Uchida, H., Yee, A. and Amo, L. A new test to measure emotion recognition ability: Matsumoto and Ekman's Japanese and Caucasian Brief Affect Recognition Test (JACBART). Journal of Nonverbal Behavior. pp. 179-209, 2000.

[Mat11] Matsumoto, D., Hwang, H.S. Evidence for training the ability to read microexpressions of emotion. Motivation and Emotion. pp. 181-191, 2011.

[Bru93] Brunelli, R., Poggio, T. Face recognition: Features versus templates. IEEE transactions on pattern analysis and machine intelligence. pp. 1042-1052, 1993.

[Xio13] Xiong, X., Torre, F. Supervised descent method and its applications to face alignment. Proceedings of the IEEE conference on Computer Vision and Pattern Recognition (CVPR 2013), pp. 532-539, 2013.

[Shr09] Shreve, M., Godavarthy, S., Goldgof, D., Sarkar, S.: Macro-and micro-expression spotting in long videos using spatio-temporal strain. 2011 IEEE International Conference on Automatic Face Gesture Recognition and Workshops (FG 2011), pp. 51-56, 2011.

[Yan13] Yan, W.-J., Wu, Q., Liu, Y.-J., Wang, S.-J., Fu, X. CASME database: A dataset of spontaneous micro-expressions collected from neutralized faces. Automatic Face and Gesture Recognition (FG), 2013 10th IEEE International Conference and Workshops, pp.1-7, 2013.

[Yan14] Yan, W.J., Li, X., Wang, S.J., Zhao, G., Liu, Y.J., Chen, Y.H. and Fu, X.. CASME II: An improved spontaneous micro-expression database and the baseline evaluation. PloS one, 9(1), 2014.

[Sun10] Sun, D., Roth, S., Black, M.J. Secrets of optical flow estimation and their principles. Presented at the Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference, pp. 2432-2439, 2010.

[Oja96] oja96 Ojala, T., Pietikäinen, M., Harwood, D. A comparative study of texture measures with classification based on featured distributions. Pattern recognition. pp. 51-59, 1996.

[Dal05] Dalal, N., Triggs, B. Histograms of oriented gradients for human detection. 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2005), pp. 886-893, 2005.

[Li15] Li, X., Hong, X., Moilanen, A., Huang, X., Pfister, T., Zhao, G., Pietikäinen, M. Reading hidden emotions: spontaneous micro-expression spotting and recognition. arXiv preprint arXiv:1511.00423. 2015.

[Bla96] Black, M.J., Anandan, P. The robust estimation of multiple motions: Parametric and piecewise-smooth flow fields. Computer vision and image understanding. pp. 75-104, 1996.

[Del15] De la Torre, F., Chu, W. S., Xiong, X., Vicente, F., Ding, X., & Cohn, J. Intraface. In Automatic Face and Gesture Recognition (FG), 2015 11th IEEE International Conference and Workshops on, pp. 1-8, 2015

[Xia16] Xia, Z., Feng, X., Peng, J., Peng, X., & Zhao, G. Spontaneous micro-expression spotting via geometric deformation modeling. Computer Vision and Image Understanding. pp. 87-94, 2016.

[Pfi11] Pfister, T., Li, X., Zhao, G., & Pietikäinen, M.. Recognising spontaneous facial micro-expressions. Computer Vision (ICCV), 2011 IEEE International Conference. pp. 1449-1456, 2011.

Soft Shadow Computation using Precomputed Line Space Visibility Information

Kevin Keul
Department of Computer Graphics, Institute for Computational Visualistics, University of Koblenz-Landau, Koblenz, Germany
keul@uni-koblenz.de

Nicolas Klee
Department of Computer Graphics, Institute for Computational Visualistics, University of Koblenz-Landau, Koblenz, Germany
nklee@uni-koblenz.de

Stefan Müller
Department of Computer Graphics, Institute for Computational Visualistics, University of Koblenz-Landau, Koblenz, Germany
stefanm@uni-koblenz.de

ABSTRACT

Shadows are one of the most important effects to create realism in rendering. Most real-time applications use some sort of image based technique like shadow mapping. While these techniques are quite fast, they often struggle at rendering realistic and accurate shadows of area lights. To produce correct shadows it is therefore often necessary to use ray tracing with some sort of acceleration method, nowadays mostly GPU based BVH which have their downsides in real-time rendering. We present a novel approach in calculating approximated but fast shadows using the line space as precomputed data structure for visibility information. With that it is possible to skip intersection tests with scene geometry and completely rely on the line space data structure for the shadow computations of area lights. Our approach is therefore almost scene-independent and is able to produce accurate shadows with better performance in comparison to typical ray tracing data structures.

Keywords
Visualization, Computer Graphics, Ray Tracing, Data Structures, Visibility Algorithms

1 INTRODUCTION

Computing shadows is one of the most important ways to enhance realism in a scene. Shadows increase the spatial perception and with that the overall appearance of realistic rendering results. A simple way to compute shadows is to compute the distance of the foremost objects to a point light source first and store those in a shadow map. This map can then be used to differentiate occluded from lighted objects. This approach can be applied in combination with typical rendering and is therefore useful for exploitation of the parallel nature of the graphics processing unit (GPU). With that it is fast and gives realistic results for point light sources. But in most cases area lights are favoured because of better quality in realistic scenes and shadow mapping techniques fail to produce fast shadows of those with good quality. Therefore in most approaches some sort of ray tracing method is used to approximate the surface of the area light source with multiple samples. While the

Figure 1: Soft Shadow computation by using 49 Shadow rays for a static scene. The scene is rendered with typical forward rendering while the shadows are computed with our ray tracer using early line space termination. The usage of the line space grants better performance compared to an equivalent BVH-based ray tracer with similar quality.

results of these techniques have a good and physically plausible quality, the computation needs quite a lot of time even with good acceleration data structures.

We propose a novel approach to compute approximate shadows using the line space as acceleration data structure. By using the line space it is possible to precompute visibilities which are used in our method for blocker calculation. With this we do not need to test the actual scene geometry for intersection, but we use approximate occlusions based on the shaft informations of the line space. This increases the rendering performance and allows us to only store the line space data structure in GPU memory with no need of storing any geometry information at all. One downside of our method is that the produced shadows are not precise because of the approximations with shafts. By using the line space for area lights we are able to show that these inaccuracies are negligible. Our results demonstrate that the use of the line space leads to a faster method compared to other ray tracing data structures and better quality compared to typical image based techniques.

Our main contributions are:

- An approximate technique for shadow computation using the line space as termination criterion.
- An acceleration for rendering approximate soft shadows of static scenes in real-time on modern GPUs.
- An analysis and comparison of the benefits of our technique.

2 RELATED WORK

Shadow Methods Rendering of shadows is a well researched topic and we will only give a brief overview of recent and relevant work. For further information we refer the reader to [Eis11] and [Has03].

Many methods exist for the task of rendering shadows. Starting with the work by [Wil78] there have been many approaches to image based methods. There, the occluding objects are stored in a so called shadow map first. In a second pass it is possible to determine with only one texture lookup of the shadow map which objects are visible from the light source. Lighting therefore has to be computed for exactly these objects, while all nonvisible objects from the light source have to be shaded. This standard process of shadow mapping is fast but tends to have visible aliasing artifacts if the resolution of the shadow map is not big enough. In this form, it is only possible to produce hard shadows, where the light source has no volume at all but is only represented by a single point in space.

Percentage closer filtering [Ree87] is one possibility to reduce the problem with aliasing through blurring of the shadow edges. It works by taking not only one but multiple nearby texture lookups of the shadow map and using this to calculate the percentage of visibility from the light source. With adjustments to this it is possible to approximate soft shadows from area light sources [Fer05]. There, the size of the filter kernel is adjusted according to the distance of the occluder. With this approximation the shadow is not physically accurate but the results are sufficient in many cases.

Other concepts to create shadows are geometry based methods based on the generation of shadow volumes that enclose the shadowed space [Cro77]. It is possible to create correct shadows for point lights with hard shadow edges but it also needs some adjustments to create soft shadows with this idea [Ass03] [Lai05]. In general, image based methods using some kind of shadow mapping algorithm are more popular in comparison to geometry based methods. This is due to performance reasons and a greater versatility and applicability of shadow mapping algorithms, but both approaches can benefit from rasterization and are therefore fast.

Ray Tracing Methods A different approach to compute shadows is usually done with some kind of ray tracing algorithm. For each point that has to be tested for lighting a ray is constructed starting in that point and ending in the light source. If the ray is not intersecting scene geometry on this path then the point is lit, otherwise it is shadowed. This approach is more versatile in comparison to the previous ideas but the calculation of the intersection between rays and scene geometry is rather slow. Among the most popular and effective acceleration data structures for this are bounding volume hierarchies (BVHs) because of good performance [Sti09][Ail09]. Recently, there have been approaches to efficiently build good BVHs on the GPU [Kar12][Kar13]. Construction can be parallelized for example with SAH binned methods [Wal07][Wal12] or by using linear BVHs [Lau09]. This way it is possible to produce interactive results for construction and traversal of the data structure on GPUs. Extensions like multi bounding volume hierarchies are further exploiting the parallelization to get better performance by storing more subnodes per node than usual [Ern08][Wal08][Áfr14].

While all previous approaches use the scene geometry to test for intersections, there are algorithms that avoid that. Among the most popular are sparse voxel octrees (SVOs) where far away nodes are used to compute shadow occlusions [Gob05]. Efficient SVOs where proposed that use contours in order to decide whether the subdivision of a node can stop [Lai11]. With this it is possible save memory. Compression methods for SVOs were introduced in [Käm13], which can be precomputed [Sin14]. It was shown that the construction and the traversal speed are fast enough to be used in real time applications [Cra11].

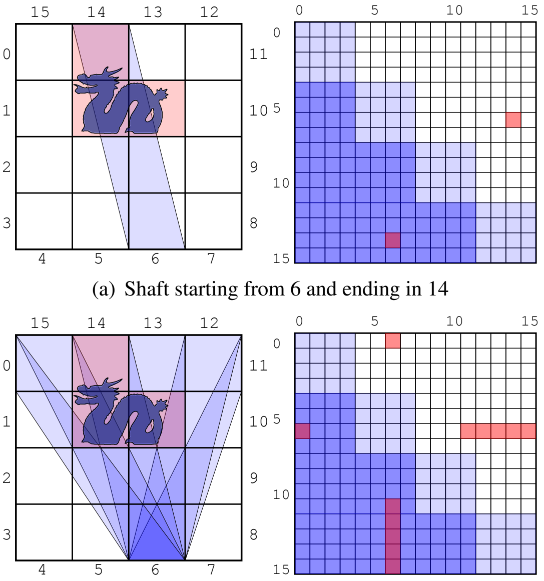
Visibility Precomputation Moreover, there have been approaches that precompute visibilities for example in radiosity calculations. Line space computations were used by [Dre97], where they compute shafts between two arbitrary surface elements. Those shafts represent all possible rays between the corresponding surface elements and thereby visibilities can be precomputed. This approach was recently applied to the N-Tree [Keu16], which is a variation of recursive grids [Jev88]. In this attempt every subdivided node has the line space information as well, which summarizes every possible shaft in this node and shows which shafts are empty or non-empty. This information is used during traversal. Precomputed visibilities for urban scenes were proposed by [Bit01] and [Ley03]. They also use the term ”line space” but in a different meaning compared to approaches mentioned above.

3 LINE SPACE INFORMATION

Our goal is to compute shadows without testing the scene geometry for intersection. For this, we create a data structure with the help of the scene geometry which does not need the geometry afterwards. In that our data structure is similar to sparse voxel octrees as they are for example used in voxel cone tracing [Cra11]. In contrast to octress we do not store leaf-nodes containing scene geometry at all. Instead we store a line space in every subdivided (= non-leaf) node and ignore the deepest level of the tree as proposed by [Keu16]. We therefore have approximated shadows comparable to those of sparse voxel octrees. By using the direction based data of the line space and its early termination criterion (which is explained later on) we are able to accelerate the shadow computation even further.

A line space contains the visibility information for every possible shaft within the corresponding node. A shaft is expressed through a given start and end side from the nodes surface. In the N-Tree every node is axis aligned and can be represented with an axis aligned bounding box (AABB). Every subdivided node has precisely $N \times N \times N$ subnodes, so the surface of each side of the AABB is subdivided in $N \times N$ subsides. Those subsides serve as the start and end sides for the shafts. A shaft therefore is able to group all rays starting and ending at specific subsides of the AABB of the node.

It can be precomputed which subnodes of the current node are intersected by a certain shaft. If all intersected subnodes are empty, so they do not contain any geometry of the scene geometry at all, it is possible to conclude that all possible rays within this shaft are unable to intersect scene geometry within this node and all potentially intersected subnodes. For this, a shaft only needs the information if all intersected subnodes are empty or if there is at least one non-empty subnode intersecting the shaft. This can be expressed in one bit of information. The line space contains this information for all possible shafts of one node and can be represented as a 2D array or texture where the first dimension stands for the start side and the second dimension for the end side of a shaft.



(a) Shaft starting from 6 and ending in 14

(b) All non-empty shafts starting from start side 6

Figure 2: Illustration of the the N-Tree and part of the according line space in 2D. In the upper images one non-empty shaft and the associated entry in the line space are shown. In the lower images all non-empty shafts starting from one specific start side are shown. The light blue entries in the line space represent shafts that start and end on the same sides of the bounding box, while the dark blue entries are symmetric information and therefore unnecessary. Note that every red node in the N-Tree may be subdivided as well and would therefore contain line spaces on their own.

Figure 2 shows an example of the line space information, where it is shown that the entries of this array are symmetrical around the diagonal because of the inversion of start and end sides of the shafts. Shafts that start and end on the same side of the AABB do not contain any volume at all and are therefore always empty which is observable in the empty squares around the diagonal. Keeping this in mind, the necessary size of the line space can be reduced to less than a half.

As with other data structures finding the correct settings is important. The line space has two essential parameters: the maximum depth and the branching factor of the underlying N-Tree. The maximum depth limits the maximum number of nodes. Only subdivided nodes contain a line space. So in our case the deepest level in the tree is not needed after the initialization anymore. According to [Keu16] the branching factor

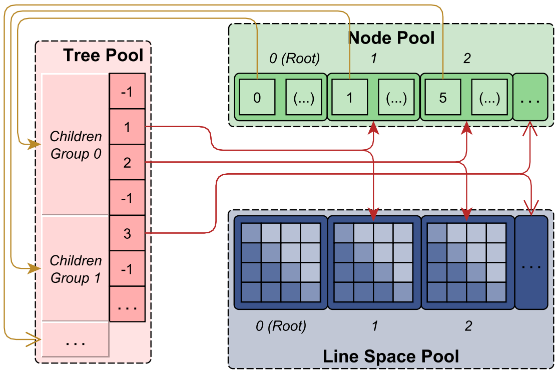


Figure 3: Management of the GPU data structure. The entries of the tree pool are clustered in children groups of nodes. Each entry of this pool is either set to a default value or refers to a subdivided node (stored in the node pool) and its corresponding line space (stored in the line space pool). Each node in the node pool contains the necessary data for its traversal and therefore a reference to its children group which can be traversed recursively.

N has a significant impact on the shape of the shafts and therefore on the traversal speed. A high branching factor leads to long and slim shafts which are good for skipping many subnodes during traversal at once. On the other hand the number of entries within the 3D line space is $15 \times N^4$, so a high value for the branching factor results in a huge memory consumption. For good traversal results it was stated that the optimal value for the branching factor is between 6 and 10 and for the depth is either 3 or 4.

3.1 GPU Data Structure

Adapting the idea of [Cra11] we implement our data structure in data pools. We use three different pools, which are stored in linear buffers on the GPU. In the first data pool (the tree pool) we store the tree information of the N-Tree where all node relations are ordered in groups of subnodes. In the second data pool (the node pool) the information for all subdivided nodes is stored. This information is used to compute the traversal of the subnodes of one node. The third data pool (the line space pool) is used for all line space information of the subdivided nodes. This concept is illustrated in figure 3. We implemented our approach with OpenGL Compute Shaders and therefore optimized all storage units for this.

The data structure is used in a way to only rely on subdivided nodes for traversal. Leaf nodes containing scene geometry are not needed and therefore not stored within the data pools. The tree pool consists of all possible pointers that are needed to represent the hierarchy. The order of the pointers is based on groups of children of subdivided nodes. All children of one node are clustered to one children group. They have a specific internal order, which is the same for all children groups and dependant on the local position within the parent node. If a children node is subdivided then the pointer is set to the index of the children within the node pool. If the children is not subdivided then the pointer is set to a default value, indicating that the traversal can skip this node. Instead of using a default value it is also possible to use negative values with a special meaning. With this it is possible to also store references to geometry information if needed.

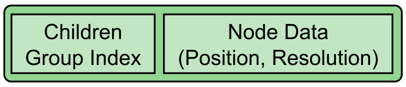


Figure 4: Illustration of the structure of one N-Tree node. It consists of different attributes which are used for the traversal of the tree. All subdivided nodes have a reference to the children group index of this node. The reference is used in combination with this nodes position and additional information like its resolution for the traversal of its subnodes. If it has no subdivided children then the reference is set to a default value, indicating that this node can be skipped during traversal.

The node pool is also used for the traversal. During traversal of the line space leaf nodes are skipped completely. For efficient memory usage therefore only subdivided nodes are stored within the node pool. A node within the node pool consists of different attributes as illustrated in figure 4 which are used for the traversal. The main attribute of a node is a reference pointer to its children group within the tree pool. Other information needed for the traversal are the position of this node in world space and its size. It is possible to store additional information of a node like its resolution for the case that nodes can have a variable number of children nodes.

The data of the line space pool is used as termination criterion in the traversal. The single bit information whether a shaft is empty is stored in this pool. For better incorporation with GPU-memory, the information of multiple shafts is combined to an integer value. The partition of the pool correlates with the node pool, so the n-th node in the node pool is related to the n-th line space in the line space pool. With this a pointer of the tree pool simultaneously refers to the corresponding node and its line space as shown in figure 3. While a line space, as explained above and shown in the figure, is illustrated as a two-dimensional data set, it is in fact implemented as one-dimensional data within the buffer. It consists of a sequence of combined integer values and is therefore stored in an efficient way.

4 SHADOW COMPUTATION

Using the line space we have a data structure that is able to decide whether a ray is probably going to intersect scene geometry through a given shaft or if it is

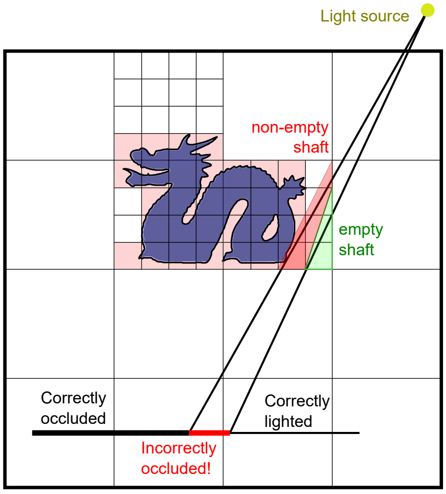


Figure 5: Illustration of the line space used for shadow computation. The object on the bottom is partially occluded by the dragon. While the occlusion on the left of the object is correct, the usage of the line space also results in incorrect occlusion. This inaccuracy is based on non-empty shafts in which the shadow rays would normally pass by the scene geometry without intersecting it, but are wrongly classified as occluded. Note that this inaccuracy can be greatly reduced by using a higher branching factor for the underlying N-Tree.

definitely not going to intersect anything. Visibility of a light source in this context is therefore merely an approximation for the pure possibility of visibility. If a shaft only intersects empty subnodes, it is called empty itself. An empty shaft does not change the possibility of visibility and therefore the light source counts as ”visible”. If a shaft intersects at least one non-empty subnode, then the shaft is called non-empty. A non-empty shaft may be able to block the visibility of a light source and as a result we classify this shaft as ”occluding”. Note that this is a rather conservative estimation of occlusion. A ray to a light source within a shaft may be declared as occluded even though it may pass by the scene geometry contained in the subnodes. An example to this is shown in figure 5. Although this approximation may result in places that are wrongly occluded, this technique allows for a quick shadow traversal without the need to test the actual scene geometry for intersection. The benefit is not only that the test for occlusion is faster than the typical occlusion test in ray tracing, but furthermore it is not necessary to store the scene geometry in the data structure at all. With this our data structure is mostly scene-independent.

4.1 Traversal

Normally the traversal for shadow computation traverses through the data structure until a node with scene geometry is found. This geometry is then tested for intersection. If an intersection is found within the node, then the traversal can end with a positive result. Otherwise it has to continue until an intersection is found or the data structures boundary is reached. Depending on the data structure the tests for intersection with scene geometry may be more costly than the traversal of the data structure itself.

In our case we try to use this fact in two ways. On the one hand we have an implicit intersection test of the scene geometry which is done with the help of the shaft information in the line space. This shaft information is precomputed during initialization of the line space, so no intersection tests with scene geometry have to be done during rendering. On the other hand we try to accelerate the traversal of the data structure by skipping the deepest level of the hierarchy by using the shafts as a summary of the underlying deepest level.

Algorithm 1 The recursive line space traversal algorithm for shadow computation

```
 1: procedure TRAVERSE(Ray r, Node n)
 2:     if n has subnodes and LS(r, n) is true then
 3:         if n is deep enough in hierarchy then
 4:             return true
 5:         else
 6:             while subnodes left do
 7:                 s ← next subnode in direction of r
 8:                 if s is non-empty then
 9:                     if TRAVERSE(r, s) then
10:                         return true
11:                     end if
12:                 end if
13:             end while
14:         end if
15:     end if
16:     return false
17: end procedure
```

The final traversal algorithm is shown in algorithm 1. All nodes in the data structure are either subdivided and contain a line space or are leaf nodes and contain scene geometry. The latter are not needed during traversal as explained above, so this is checked first. If the node is subdivided then the entry for the ray within the line space of this node is checked (line 2). If the entry is not set, it means that the according shaft of this ray is empty and therefore the ray is not able to hit anything within this node. Further inspection of this node can be skipped. If the entry is set, so the according shaft is non-empty, then the traversal of this node continues. Next it is tested whether the current node is deep enough in the tree hierarchy (line 3). A node is called deep enough, if it is on the second to last level in the hierarchy, so all subnodes of this node are leaf nodes. With this it may be possible that scene geometry is intersected by the ray and the ray gets accepted as occluded. Note that this is the part where the shadow is approximated. If the current recursion depth is not yet deep enough then the subnodes intersecting the ray are being recursively tested for intersection (line 9).

A drawback using the line space without scene geometry is that it does not work if the ray starts within a shaft. The information stored within this shaft can not be applied to the ray because of the uncertainty how the objects within the shaft may be positioned in relation to the ray. Therefore the node where the ray starts has to be skipped in the process and the traversal needs to start with the next node. This leads to a loss in quality in detailed areas.

4.2 Soft Shadow Computation

Using the line space as termination criterion for the traversal has the benefits of a faster occlusion test and it may need less memory space. The downside to this is the loss in accuracy which comes from the approximation of the scene geometry with shafts as explained above. This effect is observable at the edges of the shadow regions where in general the line space produces more occluded areas as other approaches (see figure 5). Then again in most cases there are no point lights required but area lights, which do normally not produce hard shadow edges but soft transitions between occluded and non-occluded areas. Most approaches try to generate this effect by using multiple samples of the area light and calculating the percentage of non-occluded samples for lighting. By using this technique combined with the line space, the approximative nature of line space generated shadows become negligible (see figure 7).

Though the shadows generated by this are overly shadowed, the difference is almost not visible, especially when many samples are used. In addition the inaccuracy caused by the line space is especially less significant the bigger the area light sources become. This is due to the fact that less samples near the geometry edge are falsely occluded by the line space. An example of soft shadows generated by the line space is shown in figure 6. By only testing the visibility based on shafts and not on the actual scene geometry the traversal of the shadow rays is also more coherent and therefore better suitable for parallelization with the GPU.

5 RESULTS

We implemented our approach on a NVidia GeForce GTX 1080 system with an Intel i7-6800k 3.6 GHz CPU and used GLSL Compute Shaders for GPU computing. As test scenes we used typical test models with different numbers of triangles and different characteristics (Bunny 69k triangles, Dragon 871k triangles, Buddha 1087k triangles and Dragon and Buddha combined) on top of a simple plane. All scenes were rendered with a resolution of 1024 × 1024. While the geometry is rendered using typical forward rendering first, the shadows are applied using one of the techniques mentioned. We use different camera angles and take the average run time as result. We compared our method with the state of the art in BVH accelerated ray tracing using the GPU as proposed by Aila and Laine [Ail09].

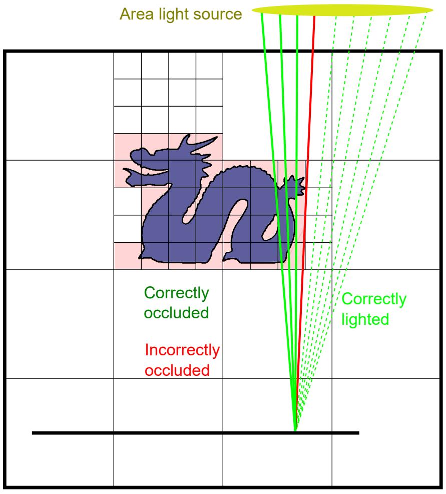


Figure 6: Illustration of the line space used for soft shadow computation. In this example 10 samples of the light source are used to calculate occlusion, while one of those samples is wrongly occluded. Note that as with normal shadows the accuracy of soft shadows based on the line space can be greatly reduced by using a higher branching factor for the underlying N-Tree.

The size of the line space and the build time varies significantly with the branching factor and the maximum tree depth used for the underlying N-Tree. As supposed by [Keu16] we use a branching factor between 4 and 8 with a maximum recursion depth of either 3 or 4. The line space is constructed on the GPU on top of the previously build N-Tree. As with most other complex ray tracing data structures, we do not achieve interactive initialization timings. The number of nodes containing a line space and the resulting size of the data structure with different parameter sets are given in table 1. There, the build timings for the line space on top of the N-Tree and the rendering timings are shown as well.

In comparison to BVH based ray tracing, the line space does not need any intersection test with scene geometry. With this, it has a substantially better performance in computing soft shadows. The results may differ significantly in scenes that do not fit in the GPU memory, but this needs to be further investigated and is beyond the scope of our work. It is visible that the rendering performance of the line space in comparison to the BVH is better in medium and big scenes, whereas the BVH achieves faster results in the small scene. Overall it is visible that the quality of the BVH in terms of performance is mainly influenced by the number of triangles

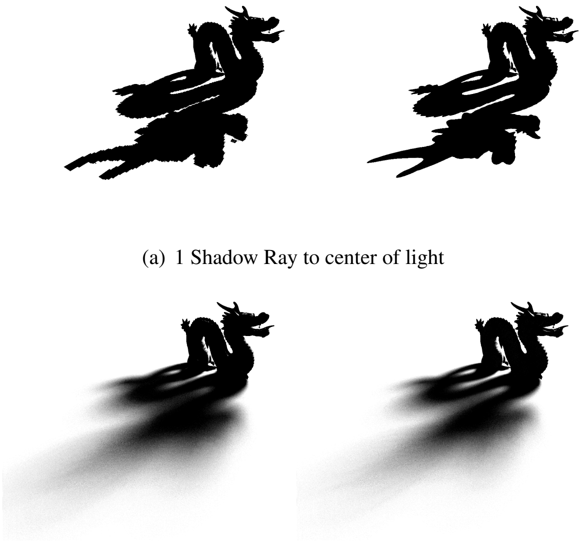
Scene		BVH	LS (4, 4)	LS (5, 4)	LS (6, 3)	LS (7, 3)	LS (8, 3)
Bunny	size (MB)	4,3	9,4	92	12,3	45,6	115,2
(69k tris)	nodes (in 1000)	31,4	13	57,2	3,9	8,1	12,4
	init time (ms)	-	28	489	100	547	1779
	render (FPS)	95,3	58,7	52,5	69,9	62,2	59,6
Dragon	size (MB)	35	9,9	97,7	13,1	48,7	122,2
(871k tris)	nodes (in 1000)	82,9	13,8	60,7	4,1	8,7	13,2
	init time (ms)	-	34	533	105	558	1788
	render (FPS)	22,4	23,7	21,1	33,6	32,2	29,8
Buddha	size (MB)	41,6	9,4	67,1	12,1	35,4	95,7
(1087k tris)	nodes (in 1000)	69,9	13,1	41,7	3,8	6,3	10,3
	init time (ms)	-	30	355	98	397	1432
	render (FPS)	16,4	47,5	42,6	60,8	57,3	19,7
Buddha &	size (MB)	76,6	12,7	89,4	15,8	46,2	127,6
Dragon	nodes (in 1000)	152,7	17,7	55,6	5	8,2	13,7
(1959k tris)	init time (ms)	-	42	475	131	523	1928
	render (FPS)	12,7	23,6	22,4	26,6	24,9	24,3

Table 1: Comparison of the size in MB, number of subdivided nodes, build time in ms and rendering times in frames per second for different parameter sets of the line space and BVH based on the work by Aila and Laine [Ail09]. Every parameter set of the line space is given as (N, d), where N stands for the branching factor and d for the maximum depth of the used N-Tree. For the rendering we measured the time to compute soft shadows of one area light source with 25 shadow rays with an image resolution of 1024×1024. It is visible that the line space has better rendering performance in medium and big scenes, but not in scenes with only a small number of triangles.

used in the scene. In contrast, the performance results of the line space are as expected more stable with varying numbers of scene triangles, but are more influenced by the spatial structure of the scene. This is due to the fact that the line space does not store the scene geometry in any kind, but an approximation of the scene within the abstraction of the shafts.

In terms of memory consumption the BVH is mainly affected by the number of scene triangles which have to be stored in addition to the node information of the hierarchy. In contrast the line space does not need to store any triangle information at all and it can only rely on the node and line space data. Figure 8 shows the memory consumption for the mentioned scenes for the BVH and the line space with the tested parameter sets. The size of the line space significantly depends on the used parameter set, where a higher value of the branching factor N or the maximum tree depth d leads to a much higher memory consumption. As with the rendering performance, it is visible that the memory size of the line space is nearly scene independent and quite stable over all used scenes. With this it is possible to give an early estimation of the memory size for a given parameter set before actually building it.

As explained before, the usage of the line space for shadow generation is flawed with approximations due to the abstraction with the shafts. This is especially visible when only one shadow ray is used per pixel. In soft shadow computation, this problem is only barely noticeable and can be reduced by increasing the parameter set values used. Figure 7 illustrates this effect.



(a) 1 Shadow Ray to center of light

(b) 25 Shadow Samples to area light

Figure 7: Rendering results for the Dragon Scene using a single shadow ray (top) and 25 shadow samples (bottom). The shadows in the left images are computed with the line space with a low valued parameter set for illustration (a branching factor N of 5 and tree depth d of 3) while the shadows in the right images are computed with a BVH as ground truth data. Using the low values in the line space parameter set leads to a bad shadow silhouette in the case that 1 shadow ray is used. In soft shadow computation the difference to the ground truth data is nearly not visible.

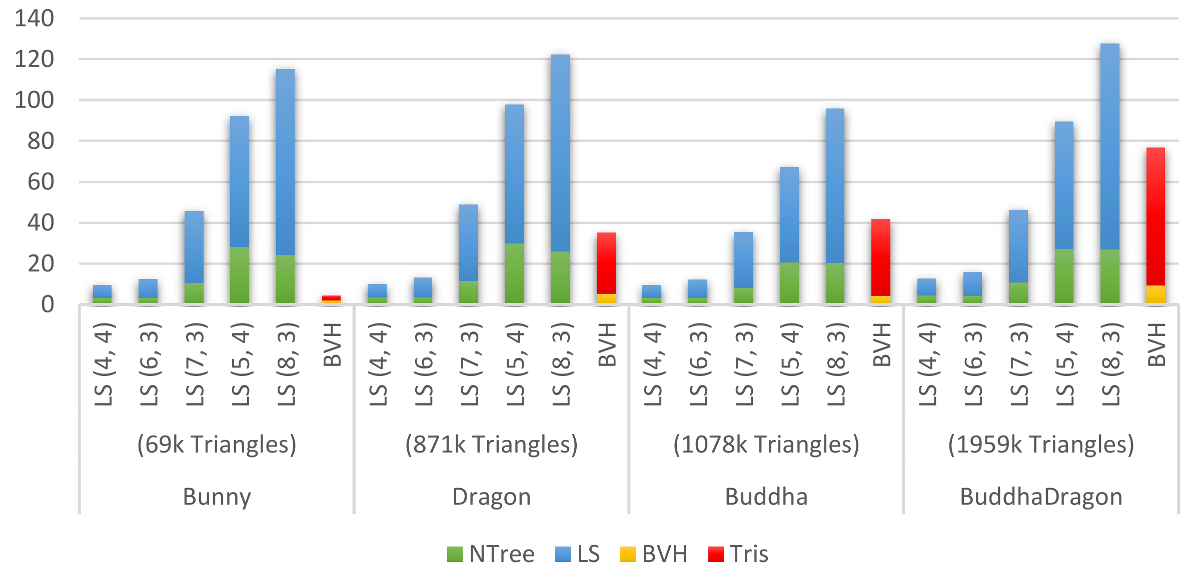


Figure 8: Overview of the memory usage (in MB) of the line space using different parameter sets and BVH for the test scenes. Although it is not necessary to store triangle information within the line space data structure, it is important in terms of memory usage to choose the correct parameter set. Using a parameter set with a big value of *N* or *d* leads to a high memory consumption, as shown by the parameter sets of LS (5, 4) and LS (8, 3). It is visible that the size of the used memory is nearly scene independent and mainly depends on the used parameters. The size of the BVH is rather small but needs the triangle information of the scene in addition.

The rendering results for soft shadow computation are shown in figure 9 on the last page. The usage of the line space leads to faster results in bigger scenes with only a minimal loss in accuracy. Nevertheless, the shadows are only approximated as explained above and so the quality of the line space results is not as precise compared to accurate computations using BVH accelerated ray tracing. The results show the mentioned difficulties of our approach. It is observable that different parameter sets of the line space lead to different results in the shadow generation, which can be explained with the varying orientation and size of the shafts within nodes with different resolutions. Furthermore it is visible, that shadows in detailed areas get lost with the line space and the shadows are in total slightly darker. But overall, these inaccuracies are only barely noticeable. The results are nearly similar to the precise results of the BVH traced method and are therefore quite acceptable.

6 CONCLUSION AND FUTURE WORK

We present a novel approach in calculating fast shadows with the GPU. Using the line space as a data structure for precomputed approximated occlusion values leads to a fast traversal of shadow rays. Though the produced shadows are not absolutely accurate, they are precise enough for soft shadows. Moreover, we showed that the results are faster in production than typical ray tracing methods. Furthermore, the data structure does not need information about the scene geometry for shadow calculation and it is therefore able to have a smaller memory consumption.

As future work we want to accelerate the initialization process by computing it in parallel on the GPU. With that it may be possible to work with dynamic scenes and the line space may then become an alternative for typical soft shadow techniques for dynamic scenes.

Apart from this we want to investigate the impact of storing not only binary information for shafts. For example it may be possible to precompute ambient occlusion values per shaft and store them within the data structure. This would accelerate the traversal step further and may therefore lead to better results. Another option would be to not store binary information in a shaft, whether it is intersected by scene geometry, but to count the number of objects intersecting the shaft. This may give the possibility for dynamic updates, so that it is not necessary to recompute the full line space once an object is moving.

Furthermore we want to identify the extents of the usefulness of the line space. In previous work it was shown that it is possible to speed up the general traversal in ray tracing. Additionally we showed that it is possible to use the line space visibility information for shadow computations. In the future we want to examine if other information for shafts may also grant the possibility to compute indirect or global illumination.

7 REFERENCES

[Áfr14] Áfra, A. T. and Szirmay-Kalos, L. Stackless multi-bvh traversal for cpu, mic and gpu ray tracing. In *Computer Graphics Forum*, volume 33, pp. 129–140. Wiley Online Library, 2014.

[Ail09] Aila, T. and Laine, S. Understanding the efficiency of ray traversal on gpus. In *Proceedings of the conference on high performance graphics 2009*, pp. 145–149. ACM, 2009.

[Ass03] Assarsson, U. and Akenine-Möller, T. A geometry-based soft shadow volume algorithm using graphics hardware. In *ACM Transactions on Graphics (TOG)*, volume 22, pp. 511–520. ACM, 2003.

[Bit01] Bittner, J., Wonka, P., and Wimmer, M. Visibility preprocessing for urban scenes using line space subdivision. In *Computer Graphics and Applications, 2001. Proceedings. Ninth Pacific Conference on*, pp. 276–284. IEEE, 2001.

[Cra11] Crassin, C., Neyret, F., Sainz, M., Green, S., and Eisemann, E. Interactive indirect illumination using voxel cone tracing. In *Computer Graphics Forum*, volume 30, pp. 1921–1930. Wiley Online Library, 2011.

[Cro77] Crow, F. C. Shadow algorithms for computer graphics. In *Acm siggraph computer graphics*, volume 11, pp. 242–248. ACM, 1977.

[Dre97] Drettakis, G. and Sillion, F. X. Interactive update of global illumination using a line-space hierarchy. In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, pp. 57–64. ACM Press/Addison-Wesley Publishing Co., 1997.

[Eis11] Eisemann, E., Schwarz, M., Assarsson, U., and Wimmer, M. *Real-time shadows*. CRC Press, 2011.

[Ern08] Ernst, M. and Greiner, G. Multi bounding volume hierarchies. In *Interactive Ray Tracing, 2008. RT 2008. IEEE Symposium on*, pp. 35–40. IEEE, 2008.

[Fer05] Fernando, R. Percentage-closer soft shadows. In *ACM SIGGRAPH 2005 Sketches*, p. 35. ACM, 2005.

[Gob05] Gobbetti, E. and Marton, F. Far voxels: a multiresolution framework for interactive rendering of huge complex 3d models on commodity graphics platforms. *ACM Transactions on Graphics (TOG)*, 24(3):pp. 878–885, 2005.

[Has03] Hasenfratz, J.-M., Lapierre, M., Holzschuch, N., and Sillion, F. A survey of real-time soft shadows algorithms. In *Computer Graphics Forum*, volume 22, pp. 753–774. Wiley Online Library, 2003.

[Jev88] Jevans, D. and Wyvill, B. Adaptive voxel subdivision for ray tracing. 1988.

[Käm13] Kämpe, V., Sintorn, E., and Assarsson, U. High resolution sparse voxel dags. *ACM Transactions on Graphics (TOG)*, 32(4):p. 101, 2013.

[Kar12] Karras, T. Maximizing parallelism in the construction of bvhs, octrees, and k-d trees. In *Proceedings of the Fourth ACM SIGGRAPH/Eurographics conference on High-Performance Graphics*, pp. 33–37. Eurographics Association, 2012.

[Kar13] Karras, T. and Aila, T. Fast parallel construction of high-quality bounding volume hierarchies. In *Proceedings of the 5th High-Performance Graphics Conference*, pp. 89–99. ACM, 2013.

[Keu16] Keul, K., Lemke, P., and Müller, S. Accelerating spatial data structures in ray tracing through precomputed line space visibility. In *Proceedings of the 24th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision*, pp. 17–26. WSCG, 2016.

[Lai05] Laine, S., Aila, T., Assarsson, U., Lehtinen, J., and Akenine-Möller, T. Soft shadow volumes for ray tracing. *ACM Transactions on Graphics (TOG)*, 24(3):pp. 1156–1165, 2005.

[Lai11] Laine, S. and Karras, T. Efficient sparse voxel octrees. *IEEE Transactions on Visualization and Computer Graphics*, 17(8):pp. 1048–1059, 2011.

[Lau09] Lauterbach, C., Garland, M., Sengupta, S., Luebke, D., and Manocha, D. Fast bvh construction on gpus. In *Computer Graphics Forum*, volume 28, pp. 375–384. Wiley Online Library, 2009.

[Ley03] Leyvand, T., Sorkine, O., and Cohen-Or, D. *Ray space factorization for from-region visibility*, volume 22. ACM, 2003.

[Ree87] Reeves, W. T., Salesin, D. H., and Cook, R. L. Rendering antialiased shadows with depth maps. In *ACM Siggraph Computer Graphics*, volume 21, pp. 283–291. ACM, 1987.

[Sin14] Sintorn, E., Kämpe, V., Olsson, O., and Assarsson, U. Compact precomputed voxelized shadows. *ACM Transactions on Graphics (TOG)*, 33(4):p. 150, 2014.

[Sti09] Stich, M., Friedrich, H., and Dietrich, A. Spatial splits in bounding volume hierarchies. In *Proceedings of the Conference on High Performance Graphics 2009*, pp. 7–13. ACM, 2009.

[Wal07] Wald, I. On fast construction of sah-based bounding volume hierarchies. In *2007 IEEE Symposium on Interactive Ray Tracing*, pp. 33–40. IEEE, 2007.

[Wal08] Wald, I., Benthin, C., and Boulos, S. Getting rid of packets-efficient simd single-ray traversal using multi-branching bvhs. In *Interactive Ray Tracing, 2008. RT 2008. IEEE Symposium on*, pp. 49–57. IEEE, 2008.

[Wal12] Wald, I. Fast construction of sah bvhs on the intel many integrated core (mic) architecture. *IEEE Transactions on Visualization and Computer Graphics*, 18(1):pp. 47–57, 2012.

[Wil78] Williams, L. Casting curved shadows on curved surfaces. In *ACM Siggraph Computer Graphics*, volume 12, pp. 270–274. ACM, 1978.

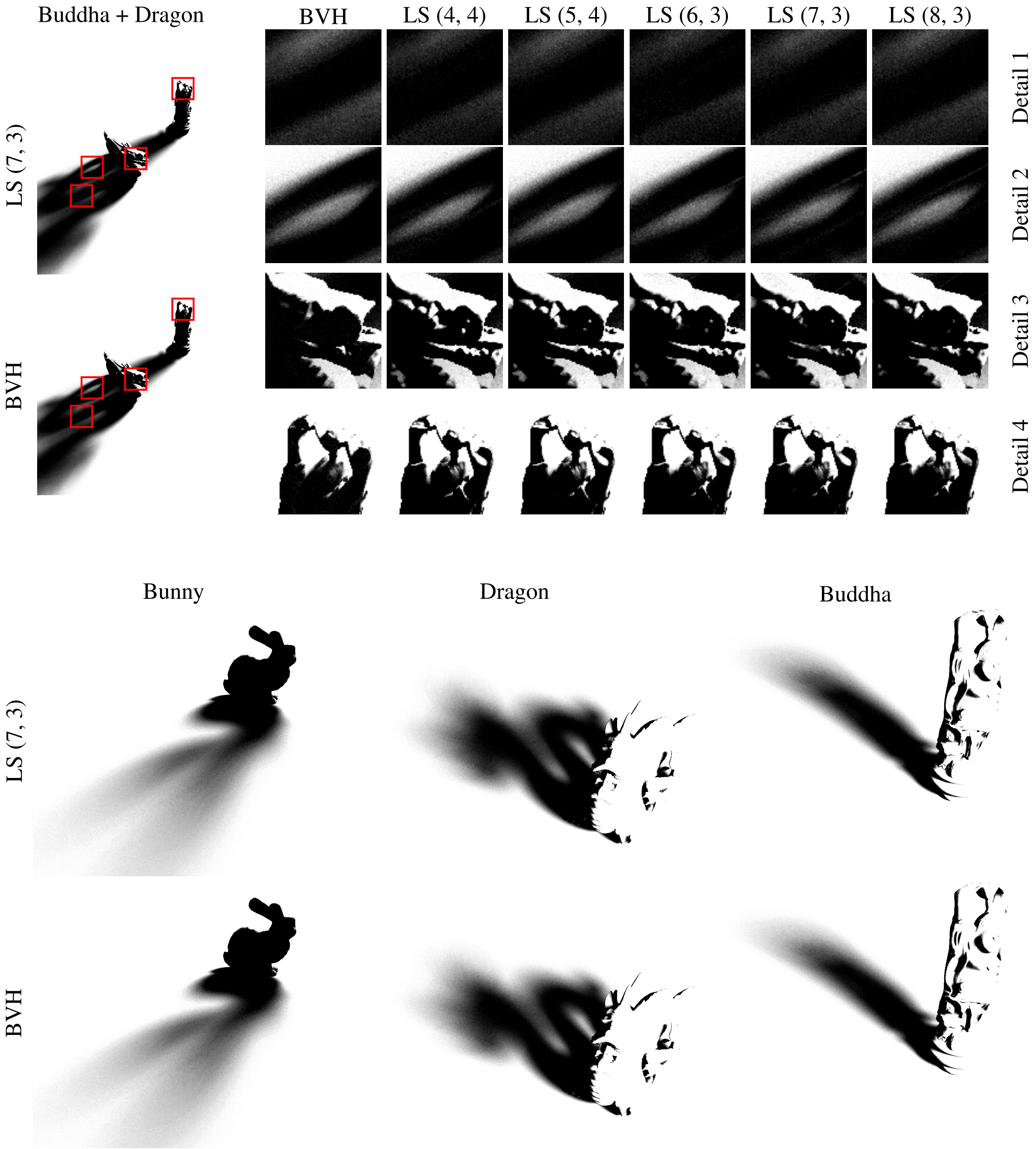


Figure 9: The test results with different test scenes. The geometry is rendered with typical forward rendering. Soft shadows are then computed with our technique using a medium sized parameter set of $(N,d) = (7,3)$. The results are compared to a BVH accelerated ray tracer based on the work of Aila and Laine [Ail09]. All images have resolutions of 1024 × 1024 and soft shadows were generated using one area light source and 25 samples. The size of the test scenes range from small (Bunny, 69k triangles) to rather big (Buddha + Dragon, 1959k triangles). Additionally two medium sized test scenes (Dragon, 871k triangles and Buddha, 1087k triangles) were used for evaluation. In the upper part the results for the big scene are shown. For the line space magnified images with varying parameter sets are illustrated to demonstrate the differences caused by the early ray termination of the line space. Using parameter sets with bigger parameter values lead to better results but have a much higher memory consumption as demonstrated in table 1. In the lower part a comparison to a BVH based ray tracer as ground truth is presented. The inaccuracies of the line space mentioned above are only barely noticeable, even with smaller parameter sets. The produced shadows are slightly darker and the skipping of the first node in the line space traversal leads to less occlusion in detailed areas. Overall, our results are similar to ground truth renderings, but have better performance and do not need to store the scene geometry, which grants less memory usage and different opportunities in the future.

Low Cost Rapid Acquisition of Bidirectional Texture Functions for Fabrics

Banafsheh Azari
CogVis/MMC,
Faculty of Media,
Bauhaus-University Weimar
Bauhausstrasse 11
99423 Weimar, Germany
banafsheh.azari@uni-weimar.de

Sven Bertel
Usability,
Faculty of Media,
Bauhaus-University Weimar
Bauhausstrasse 11
99423 Weimar, Germany
sven.bertel@uni-weimar.de

Charles A. Wüthrich
CogVis/MMC,
Faculty of Media,
Bauhaus-University Weimar
Bauhausstrasse 11
99423 Weimar, Germany
charles.wuethrich@uni-weimar.de

ABSTRACT

Creating photo realistic images from real word complex materials is a great challenge: the reflectance function of materials, especially fabrics, has glossy or specular highlights, reflectance anisotropy and retroreflections. This increases greatly the complexity of rendering. Bidirectional Texture Functions (BTFs), i.e. 2D textures acquired under varying illumination and viewing directions, have been used to render complex materials. However, the acquisition of textures for BTF requires up to now expensive setups and the measurement process is very time-consuming as the directional dependent parameters (lighting and viewing directions) have to be controlled accurately. This paper will present in detail a new low cost programmable device for the rapid acquisition of BTF datasets. The device allows to acquire BTF databases at a fraction of the cost of available setups, and allows to experiment when a texture resolution and sample density increase in the parameter space is not perceivable by an observer of the renderings. The paper proves that using smaller resolution textures and decreasing the samples in parameter space does not lead to a loss of picture quality.

Keywords

Bidirectional Texture Function Acquisition, Photorealistic Rendering, Image Quality, Hardware Devices.

1 INTRODUCTION

Rendering real world fabrics has been one of the big challenges in Computer Graphics. Fabrics exibit a complex reflection behaviour, which depends, among other factors on the meso- and micro-structures of the thread, on the type of weaving, which influences the position of the thread in the fabric, on the interreflections between the components of the fabric, and on surface and subsurface scattering of light. For highly realistic material rendering the reflectance of surface must be simulated accurately. The amount of light reflected by a material is not only dependent on the light's incident direction, but also on the angle from which a viewer looks at the fabric. Most woven materials have small three-dimensional details so that some part of surface can be occluded to the viewer depending on his position, a phenomenon which is called self occlusion. For materials exibiting small-scale bumps, depending on the illuminant position, self shadowing can occur: surface irregularities cast shadows onto other parts of the surface. Moreover, incoming light gets reflected, refracted and scattered by the fibers in the thread, leading to an even more complex reflection behaviour.

Therefore, the surface reflectance of a material is often modeled by a four dimensional function which is dependent on the viewing and lighting vector. Two additional dimensions are required for an exact rendering function due to thread color patterns and visible geometry variances due to the weaving or knitting pattern. A material function capturing all these effects has six dimensions and is therefore neither easy to design nor to evaluate efficiently [1]. In theory, all these components need to be modeled individually to simulate the reflection behaviour of the fabric.

Bidirectional Texture Functions (BTFs), introduced by Dana et al. [2], represent an alternative solution to exact rendering: instead of doing the complex modeling, pictures of the fabric taken at different illumination and viewing directions are used as textures for rendering, implicitly integrating into the rendering step the reflectance properties of the surface. A BTF contains all information on reflectance of a set of points of a surface

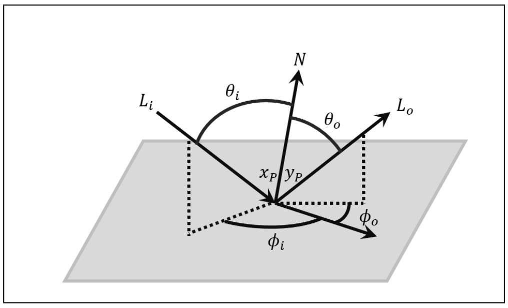


Figure 1: General light-material interaction.

under a particular lighting and viewing condition, and is described as follows:

$$S_{\mathbf{BTF}} = \int_{p \in \mathbf{P}} (\theta_\mathrm{i}, \phi_\mathrm{i}, x_\mathrm{p}, y_\mathrm{p}, \theta_\mathrm{o}, \phi_\mathrm{o})\ \delta p, \qquad (1)$$

where p is a discrete point of the sampled surface $\mathbf{P}$, the variables θ_i, ϕ_i, θ_o and ϕ_o are the angles describing the direction of the incoming and the reflected light ray $(L_\mathrm{i}, L_\mathrm{o})$ and N is the surface normal. Furthermore the parameter x_P and y_P describe the point of incidence and reflectance of the light ray. Given a BTF with varying θ_i, ϕ_i, θ_o and ϕ_o, one can accurately describe the reflection behavior of a sampled surface area.

In practice, BTFs use large collections of digitally acquired pictures of a material taken at discretely varying illumination and viewing angles. When a simulation of the material needs to be computed for rendering, the viewing and illumination vectors are used to pick matching textures from the collection of scanned textures, and, if the angles do not match with angles of the corresponding textures, neighbouring textures are interpolated at the point to be rendered.

A big disadvantage of BTFs is that state of the art measurement devices require expensive robotics setups and that the measurement process is very time consuming since direction dependent parameters (light- and view-direction) have to be controlled accurately. Otherwise the resulting data will be poor. Moreover, the size of BTF data can range from hundreds of megabytes to several gigabytes, since in the ideal case a large number of high resolution pictures have to be used. For real time rendering this is a big disadvantage, since either the entire collection of pictures needs to be kept in the computer memory, or computationally expensive methods have to be used to intelligently load/unload the textures. In this context, several authors focused on efficient compression methods for BTF data (including reflectance models based on linear factorization and pixelwise bidirectional reflection distribution functions, in short BRDFs, which are the general reflection model from which BTFs are derived [3]).

The focus was rarely set on the perceived quality of the results of compression or loading/unloading mipped-mapped textures [4–12], and while the existing approaches are technically well motivated, we believe that before starting to choose how and how strongly BTF data should be compressed, it makes sense to first take a step back and see how many measured samples at which resolution are needed to have the same perceived quality when rendered instead of using a complete database at the highest possible resolution, or automatically degraded texture downscalings not taking into account the final user. If the texture database is perceptually sound, then it can be reduced in its number of texture samples.

In this paper we present two basic improvements to the use of BTFs for rendering: firstly we want to address the cost of BTF acquisition by introducing a flexible low cost step motor setup for BTF acquisition allowing to generate a high quality BTF database taken at user defined arbitrary angles. Secondly, we want to adapt the number of acquired textures to the perceptual quality of the renderings, so that the database size is not overbloated and can fit better in memory when rendered. In oder to do this, we will use Daly's Visual Difference Predictor (VDP; [13, 14]) to prove that the reduced dataset acquired through our device does not lead to perceivable differences for the rendered images for a viewer.

In the next section, we will introduce how we plan to reduce the BTF database. Next the BTF measurement setup will be described in detail. Then an experimental evaluation of the BTF acquisition setup results will be presented, proving that no perceivable differences in the renderings are made by reducing the BTF database angle steps. Finally we will present some conclusions, and an outlook.

2 REDUCING THE SAMPLE DENSITY

In [10–12] the authors propose to reduce the BTF dataset size by down-sampling resolution and view/illumination angles and proved that perceived quality did not decrease. The outcomes could help prevent capturing redundant images with high resolution from a sample and this will reduce the acquisition time significantly. We propose a preprocessing step before starting to acquire the complete database to determine the down-sampling threshold. As this threshold depends on the surface characteristics of a material, each sample should be tested individually.

The first step in the proposed preprocessing method is to generate solely samples required to texturing a section of a sphere, as shown in Table 1. The produced database therefore covers a fourth of the BTF database (22*22=484 samples) as opposed to a complete BTF database (81*81=6561 samples), which we will refer to as *'BTF'*. In the next step this database is down-sampled using reduced densities and resolution

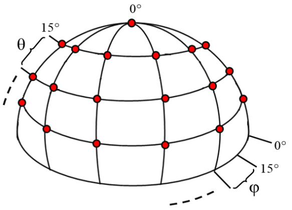
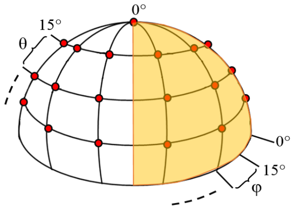
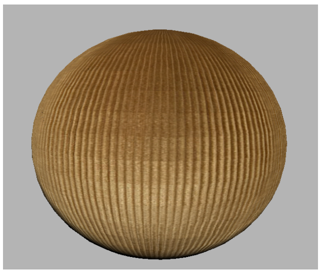
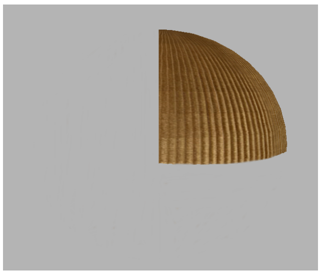
Conventional BTF Database	Proposed BTF Database	Rendered BTF Data
 81 Samples $\theta = 0°, \# \phi = 1$ $\theta = 15°, \# \phi = 6$ $\theta = 30°, \# \phi = 12$ $\theta = 45°, \# \phi = 18$ $\theta = 60°, \# \phi = 20$ $\theta = 75°, \# \phi = 24$	 22 Samples $\theta = 0°, \# \phi = 1$ $\theta = 15°, \# \phi = 2$ $\theta = 30°, \# \phi = 4$ $\theta = 45°, \# \phi = 4$ $\theta = 60°, \# \phi = 5$ $\theta = 75°, \# \phi = 6$	 

Table 1: Sampling of the Conventional BTF Database (left) compared with the Proposed BTF Database (center) and the Rendered Full and partial sphere (right).

Scheme A (11 Samples)	Scheme B (11 Samples)
$\theta = 0°, \# \phi = 1$	$\theta = 0°, \# \phi = 1$
$\theta = 15°, \# \phi = 1$	$\theta = 18.75°, \# \phi = 1$
$\theta = 30°, \# \phi = 2$	$\theta = 37.5°, \# \phi = 2$
$\theta = 45°, \# \phi = 2$	$\theta = 56.25°, \# \phi = 3$
$\theta = 60°, \# \phi = 2$	$\theta = 75°, \# \phi = 4$
$\theta = 75°, \# \phi = 3$	

Table 2: The down-sampled schemes: A along azimuth θ, B along azimuth θ and elevation ϕ angles.

Database	Number of Samples	Resolution
BTF	(22 x 22)	256 x 256
BTF-R	(22 x 22)	128 x 128
BTF-A	(A x A)	256 x 256
BTF-B	(B x B)	256 x 256
BTF-C	(22 x B)	256 x 256

Table 3: The Proposed BTF Database (*BTF*) compared with the down-sampled BTF Database using reduced resolution (*BTF-R*) and densities (*BTF-A*, *BTF-B* and *BTF-C*).

according to [10–12]. Four down-sampling schemes are adopted. In the first scheme we reduced the resolution of the each texture from 265 x 256 to 128 x128: it will be referred in this paper to as *'BTF-R'*. In order to obtain considerable reduction of BTF dataset size we adopted two different BTF sampling schemes denoted as A, B from [11]. While scheme 'A' preserves the original sampling of elevation angle θ but reduces the number of azimuthal samples along angle ϕ, scheme 'B' reduce sampling for both angles (see Table 2).

It should be noticed that BTFs require directional sampling of both illumination (θ_i, ϕ_i) and view directions (θ_o, ϕ_o) and in these two directions different sampling schemes can be adopted without limiting practical usage of the data. We choose three down-sampled BTF datasets. The first two are straightforward and down-sampled both illumination and viewing directions in the same way, using a combination of the schemes AxA and BxB, which we will refer to as *'BTF-A'* and *'BTF-B'*. The third one used scheme B on just view directions (*'BTF-C'*). Consequently, four down-sampled datasets are generated (see Table 3).

The samples in each databases are then used to render the sphere in order to check the influence of down-sampling on the perceived quality. This can be done either by using a ***Subjective Quality Metrics*** or an ***Objective Quality Metrics*** [15].

Since human beings are the users in most image-processing applications, the most reliable way of assessing the quality of an image is by using Subjective Quality Metrics. Indeed, the mean opinion score (MOS), a subjective quality measure requiring human observers, has been long regarded as the best method of image quality measurement. However, the MOS method is expensive, and it is usually too slow to be useful in real-world applications.

To solve the problem Objective Quality Metrics have been proposed. The goal of these metrics is to design mathematical models that are able to predict the quality of an image accurately and automatically. An ideal method should be able to mimic the quality predictions of an average human observer.

One of the most popular and widely used objective quality metric based on models of the human vision system is Daly's Visual Difference Predictor (VDP; [13, 14]). We used VDP to assess visual differences between the rendered spheres by different down-sampling schemes and to find a compression threshold. Based on this threshold a compressed BTF database could be acquired without capturing redundant images which reduce strongly the acquiring time. To test the method introduced above a measurement setup for the acquisition of BTF data has been built, which will be explained in detail in the next section.

3 ACQUISITION SETUP

The acquisition of 2D textures is a very simple process which can be performed using a standard 2D scanner or an off-the-shelf digital camera and image-processing software. On the contrary, the acquisition of BTFs requires a complex and controlled measurement environment. Since BTF acquisition can be seen as physical measurement of real-world reflection, special attention has to be paid to device calibration and image registration. Otherwise the measurements will contain inaccuracies which may generate visible rendering artifacts.

3.1 Prior Works

Dana et al. [2] built the first BTF measurement device. A robot arm is used in this device to orient the texture sample at arbitrary orientations and the camera and light orbit around the sample. 205 combinations of light and view directions are sampled for each material, and more than 60 materials have been measured and published[1]. Due to the sparse sampling, it is not practical to use the measured data for rendering directly.

More recently, researchers have built similar setups and provided measurements at higher angular resolutions [16–18]. Significantly to the gonioreflectometer for BRDFs, only one sample is measured at a time for each lighting and viewing directions, however, unlike the in gonioreflectometer case where one value is measured per setting, in these newer methods each sample is a texture.

For a fast high quality acquisition of BTFs Müller et al. [19] propose an array of 151 digital still cameras mounted on a hemispherical gantry. The on-camera flashes serve as light source. By synchronizing the cameras, 151*151=22801 images can be captured in 151 time steps and the authors report a measurement time of about 40 minutes. In this setup no moving parts are needed. Hence, the region of interest is known for every camera and consequently, there is no need for a time-consuming detection of the region of interest. While this is a big improvement in terms of measurement time, the setup is large and expensive.

Han and Perlin [20] introduced a measurement setup based on a kaleidoscope which allows viewing a sample from multiple angles at the same time through multiple reflections. Illumination is provided by a projector pointing into the kaleidoscope. By selectively illuminating a small group of pixels, the light direction can be controlled. Since there is no moving part in this setup, measurement is very fast. However, the equipment is difficult to build and calibrate. In addition, due to multiple reflections in the optical path, the resulting quality tends to be rather low.

Dana and Wang [21] proposed a setup based on a parabolic mirror. While their setup can provide higher quality measurements than the kaleidoscope setup, they can only capture a single spatial location at a time. As a result, it does not offer any acceleration compared to the gonioreflectometer-like approaches.

The standard gonioreflectometer-like approaches allow to capture high–quality BTFs reliably. Their drawback is the speed - several hours are needed and this makes measured BTFs an expensive resource. Using mirrors may be a promising approach in the future, but the quality of the measurements of current systems remains dubious. Using a camera array sensor greatly reduces measurement times at the expense of the costs for a large number of cameras. In the next section the proposed measurement device will be introduced in detail.

3.2 The Proposed Measurement Device

During the measurement the light source and the sensor are positioned at various angles covering the entire hemisphere above a flat sample of a homogeneous material. In other words, the system allows acquiring images from all possible angles of illumination and of camera perspective. The proposed device (Figure 2) is the result of our attempts to find a setup covering the 4 degree of freedom available in a BTF.

We set our reference coordinate as shown in Figure 2. The origin is placed on the center of the sample. The sample can rotate about the x, y and z axes. While the light can rotate about z-axis using a step motor, the camera is fixed. The camera and light are directed to the center of the sample. The system has 4 degree of freedom and is appropriate for anisotropic material. To rotate the sample we decided to use a combination of three step motors [1-3], (see Figure 2).

With these motions three degree of freedom are achieved (ϕ_i, θ_o, ϕ_o). To reach the additional degree

[1] http://www1.cs.columbia.edu/CAVE/software/curet/

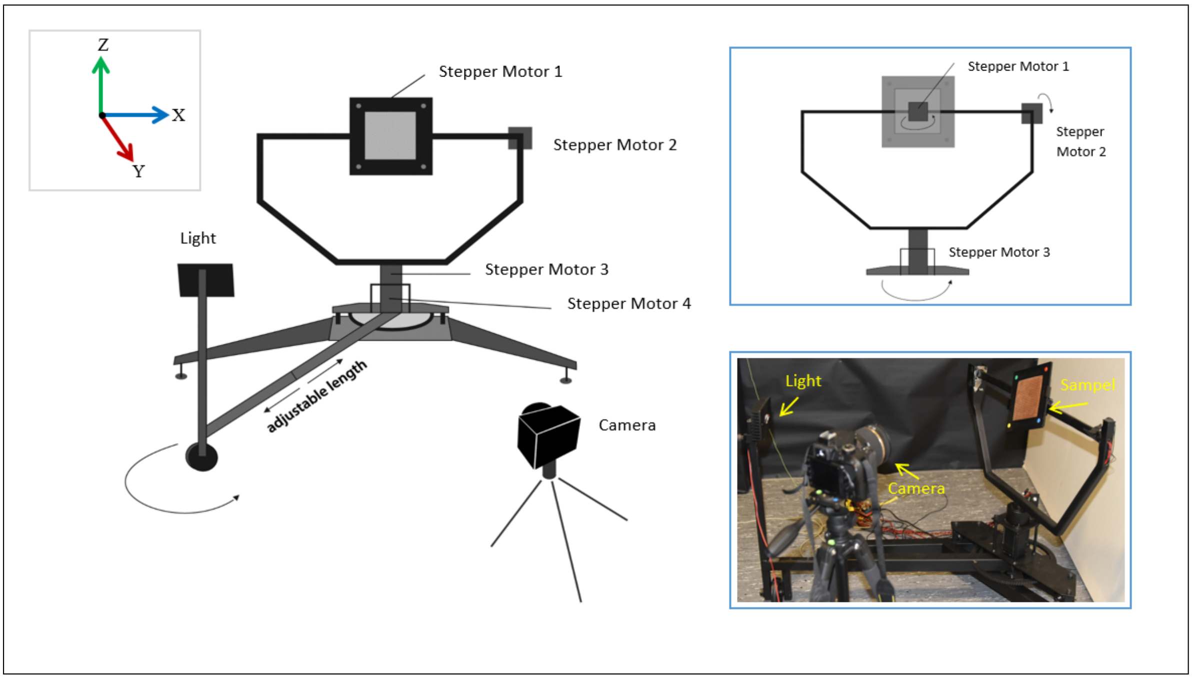


Figure 2: The proposed setup

of freedom θ_i, the light should rotate in the altitude direction. For this reason, we mounted our light source on an axes and rotate it with a wheel which can move with the help of the step motor number 4 (see Figure 2). The length of the light radius is adjustable.

The system is composed of different parts. The main component is an Arduino Mega 2560 which is equipped with a RAMPS 1.4 board and the Marlin operating system. The Arduino takes commands from a host PC and controls the motors and the remote control of the camera. This is done by using a serial connection between the Arduino and the host PC. The commands are transmitted as Gcodes[2].

Hardware: According to the hardware producer, the relationship between the voltage on the potentiometer and the motor current is given by,

$$A = \frac{V_{ref}}{8 \cdot R_s} \tag{2}$$

where A is the motor current and V_{ref} is the voltage on the potentiometer: in our case the drivers have a resistance R_s of 0.1Ω. This formula is driver specific: if another driver is used, the values have to be updated or another formula could be necessary. The system uses four step motors to move the sample in all directions.

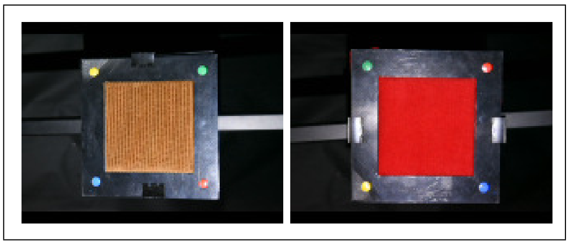


Figure 3: BTF samples

We decided to choose SM42051 and E7126-0140 step motors [3]. The SM42051 has 0.196 Nm torque with max rated current of 0.6 A and is used to rotate the sample around the x and y axes, which we will refer to as S1 and S2. Because of the higher friction force by rotating the light and sample about z axis two E7126-0140 step motors with 1.6 Nm torque has been chosen (S3, S4). The motors are connected to the according S1, S2, S3 and S4 axes of the RAMPS 1.4 board. S1, S2 and S3 move the sample while S4 moves the light source.

Each step motor has a 1.8° step resolution, Therefore in order to rotate the S1,S2 and S3 axis by 9°, 5 steps are needed. To achieve adequate leveraging, the S4 axis is equipped with a gear. The gear ratio is 9 : 120. Each tooth of the small gear wheel corresponds to a 3° movement of the light, which is the smallest possible movement of Z axis. To move the small gear wheel one tooth further, 22.22 motor steps are necessary. To have a reference, the S1,S2 and S4 axes have end-stops which are triggered whenever the corresponding axis reaches its maximum or minimum rotation.

Software: The system uses the 3D printer software Marlin as operating system. The Marlin firmware is customized for this purpose. Therefore, the file configuration header is changed at specific points, so that the

[2] Gcode is a control language for CNC (or Reprap) machines

[3] http://www.emisgmbh.de/schrittmotoren.html

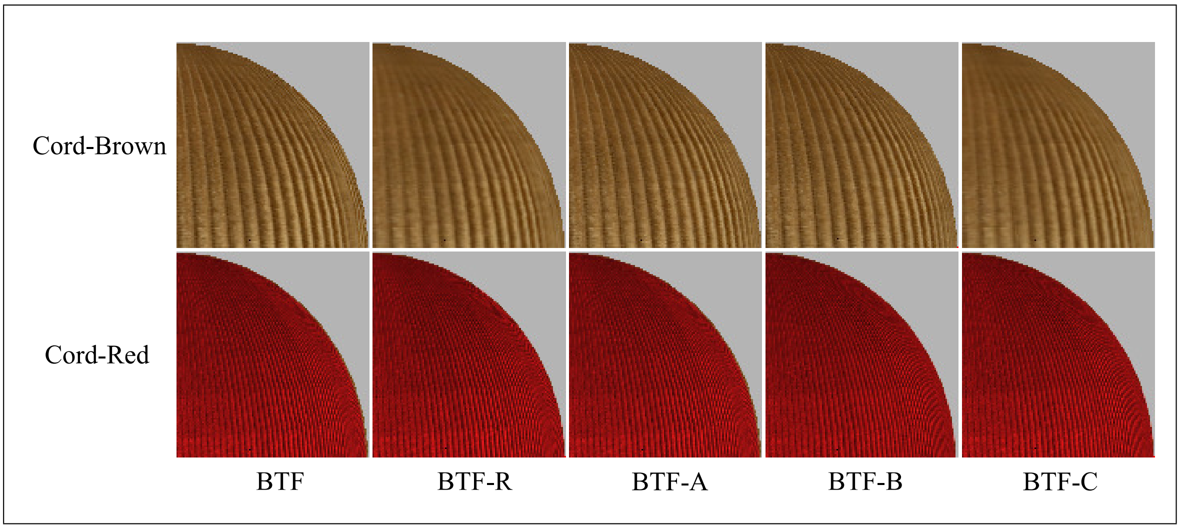


Figure 4: Objects rendered by the proposed BTF and down-sampled BTF.

system is connected to the host PC using a serial connection and receives commands from that PC. As host software for the PC Pronterface is used. After connection to the Arduino, various commands can be send to the Arduino.

The camera is the center piece of hardware in our measurement setup. Therefore special attention was paid to choose the camera. We selected a Nikon D750 DSLR camera, a high-end and full format digital camera intended for professional photography. The camera captures the material sample's appearance at different positions in raw format at a resolution of 6016 x 3375 pixels. A fixed length SIGMA camera lens (105 mm F/2.8) is mounted on the camera. Via an IR Remote Control the camera's shutter is released.

The other important piece of hardware in the measurement setup is the light source therefore it should be selected carefully as well. The decision for a specific light source was based on the emitter geometry and the lamp‘s photometric properties. An OSTAR-Lighting LED Light Source [Osram GmbH]. Besides a controlled environment and suitable equipment, we applied a number of standard algorithms to further increase the quality of the images that are used as input to our measurements. These algorithms are Geometric and Colorimetric Camera Calibration.

Geometric calibration involves the recovery of a camera's extrinsic and intrinsic parameters. While the intrinsic parameters relate the camera's coordinate system to the idealized coordinate system, the extrinsic parameters relate the camera's coordinate system to a fixed world coordinate system and specify its position and orientation in space. The actual transformation of the camera's lens system is described by its intrinsic parameters. To extract the feature points from the calibration images an implementation of the Harris detector [22] included in Bouguet's camera calibration toolbox [4] was used.

To achieve the best possible color reproduction, the camera has to be color calibrated as well. In order to relate the recorded color to well defined standards, color management systems have become a standard tool. Hereby, an image of a test target with well known properties (the Macbeth ColorChecker) was taken and processed in the same way as all later images.

After the measurement the raw image data has been converted into a set of rectified registered images. Registration is done by projecting all sample images onto the plane which is defined by the frontal view. In a final step, the textures are cut out of the raw reprojected images and resized appropriately. Finally the technique presented by Goesele et al. [23] was used to reduce the fixed pattern noise.

4 EXPERIMENT AND RESULTS

To generate high quality real world input data for appearance measurements a special purpose digital photo studio has been built. Special attention was paid to carefully control the illumination and image capturing conditions in order to be able to acquire exact data about the surface properties of samples using readily available digital camera technology. We set the distance of the light and camera to the sample to one meter and $\triangle\theta$ to 15°. We choose two planar samples with the size of 10 x 10 cm^2, Cord-Brown and Cord-Red (shown in Figure 3). 484 raw images with the resolution of 6061 x 3375 were captured for each sample. After the measurement the raw image data are projected onto the plane which is defined by the frontal view ($\phi = 0°$, $\theta = 0°$). To be able to conduct an automatic registration

[4] http://www.vision.caltech.edu/bouguetj

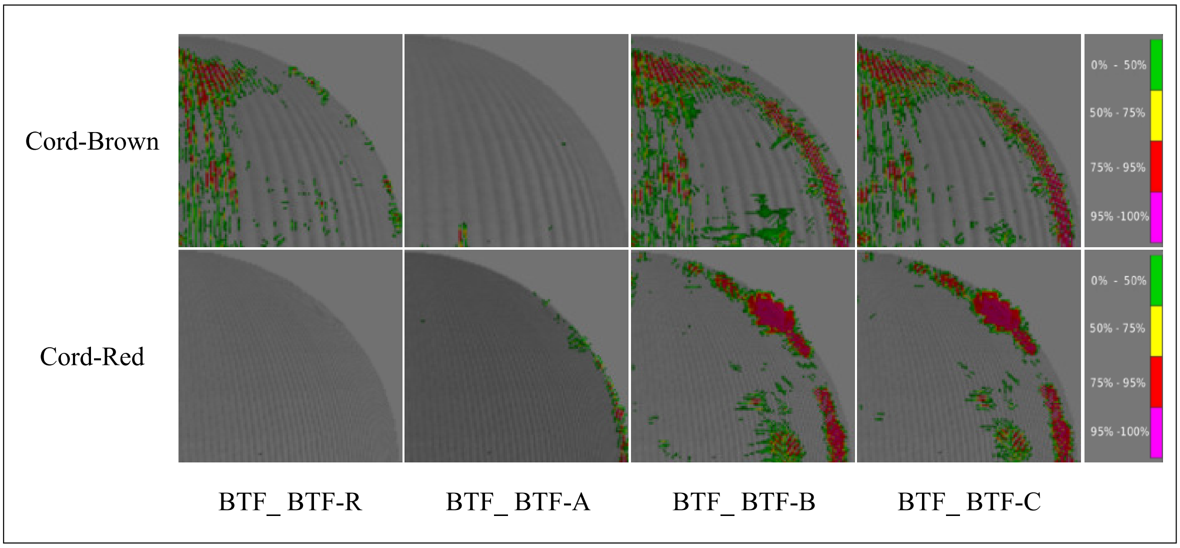


Figure 5: The responses of visual difference predictor for tested image pairs. Each pair consisted of a rendering using BTF dataset (BTF) and one of four downsampled BTF datasets (*BTF-A*, *BTF-B*, *BTF-C* and *BTF-R*). The colorscales indicate the probability values.

we have attached point markers visible at the corners of the sample holder in Figure 3. Consequently four down-sampled databases are generated out of each of these two databases: *BTF-A*, *BTF-B*, *BTF-C* and *BTF-R* (as explained in section 2). For each of the five texture data sets, a three dimensional textured model of a section of a sphere was rendered through the standard BTF rendering method [3] at a screen resolution of 1920x1084 pixels.

An objective quality metric introduced by Daly (VDP, [13, 14]) has been used to assess the perceived quality differences between objects rendered by complete and down-sampled databases. VDP simulates low level human perception for known viewing conditions (in our case: a resolution of 1920 x1080 pixels at an observer's distance of 0.7m). Figure 5 shows the visually perceivable differences per image pair as predicted by VDP. Each pair consisted of a rendering using BTF dataset and one of four downsampled BTF datasets.

It can be seen that there are not a significant perceivable quality differences between BTF and BTF_A in both of the samples while the Cord-Brown react more sensitive to the down-sampling by BTF_B and BTF_C than BTF-Red. The last row of the Figure 5 shows that the resolution reduction from 256*256 to 128*128 is not perceivable in BTF-Red. According to this information for both of the databases it is possible to reduce the number of generated samples as scheme A without losing quality: this decrease the acquisition time to 26% of the acquisition of complete database. Because less pictures need to be taken, the acquisition time becomes 4.5 hours instead of 6. In Cord-Red the captured images could have the half of the resolution, which reduce the database size 50%.

5 CONCLUSION

In this paper we presented a new low cost programmable BTF database acquisition device based on standard off the shelf components, step motors, a semiprofessional camera and a standard LED illumination source capable of capturing high quality databases. The device cuts the cost of existing database acquisition setup by a factor of hundreds.

Since the positions of the illumination source and the orientation of the sample to be acquired can be chosen at will and therefore cover all four degrees of freedom of the the parameter space, the device allows to investigate if smaller databases obtained through undersampling the parameter space allow perceptually sound renderings which show no perceptual difference with respect to a higher sampling of the parameters space.

Daly's VDP results show that in our case both the texture resolution as well as a reduction of the samples to 26% of the number of samples used in widespread databases do not deteriorate significantly the perceived quality. Furthermore, also the time spent in the acquisition of the database is also reduced to little more than one fourth.

The new device appears therefore to be an excellent compromise, cutting significantly the costs in the acquisition process (to approximately €400 plus the Camera and the lens). Moreover, its programmability allows to conceive new experiments aimed at understanding the limits at which increasing the number of samples in the database, as well as the resolution of the acquired textures makes little sense since the observer of the rendered objects does not perceive any differences.

In future work, we plan to use the device as a basis for new experiments aimed at sheding a light in the relationship between high quality rendering and the perception of observers of rendered images. This should be relevant for the Computer Graphics, Image Processing and Image Compression communities alike.

6 REFERENCES

[1] F. E. Nicodemus, J. C. Richmond, J. J. Hsia, I. W. Ginsberg, and T. Limperis, *Geometrical considerations and nomenclature for reflectance*, vol. 160. US Department of Commerce, National Bureau of Standards Washington, DC, USA, 1977.

[2] K. J. Dana, S. K. Nayar, B. van Ginneken, and J. J. Koenderink, "Reflectance and texture of real-world surfaces," *IEEE Comuter Society*, 1997.

[3] J. Filip and M. Haindl, "Bidirectional texture function modeling: A state of the art survey," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 31, no. 11, pp. 1921–1940, 2009.

[4] R. W. Fleming, R. O. Dror, and E. H. Adelson, "Real-world illumination and the perception of surface reflectance properties," *Journal of Vision*, vol. 3, no. 5, p. 3, 2003.

[5] R. Lawson, H. H. Bulthoff, and S. Dumbell, "Interactions between view changes and shape changes in picture-picture matching," *PERCEPTION-LONDON-*, vol. 32, no. 12, pp. 1465–1498, 2003.

[6] S. F. te Pas and S. C. Pont, "A comparison of material and illumination discrimination performance for real rough, real smooth and computer generated smooth spheres," in *Proceedings of the 2nd symposium on Applied perception in graphics and visualization*, pp. 75–81, ACM, 2005.

[7] F. Pellacini, J. A. Ferwerda, and D. P. Greenberg, "Toward a psychophysically-based light reflection model for image synthesis," in *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pp. 55–64, ACM Press/Addison-Wesley Publishing Co., 2000.

[8] J. Meseth, G. Müller, R. Klein, F. Röder, and M. Arnold, "Verification of rendering quality from measured btfs," in *Proceedings of the 3rd symposium on Applied perception in graphics and visualization*, pp. 127–134, ACM, 2006.

[9] L. Mcmillan, A. C. Smith, W. Matusik, and W. Matusik, "A data-driven reflectance model," in *in Proc. of SIGGRAPH*, 2003.

[10] J. Filip, M. J. Chantler, and M. Haindl, "On optimal resampling of view and illumination dependent textures," in *Proceedings of the 5th symposium on Applied perception in graphics and visualization*, pp. 131–134, ACM, 2008.

[11] J. Filip, M. J. Chantler, and M. Haindl, "On uniform resampling and gaze analysis of bidirectional texture functions," *ACM Transactions on Applied Perception (TAP)*, vol. 6, no. 3, p. 18, 2009.

[12] B. Azari, S. Bertel, and C. A. Wuethrich, "A perception-based threshold for bidirectional texture functions," *CogSci*, 2016.

[13] S. Daly, "Digital images and human vision," ch. The Visible Differences Predictor: An Algorithm for the Assessment of Image Fidelity, pp. 179–206, Cambridge, MA, USA: MIT Press, 1993.

[14] R. Mantiuk, K. J. Kim, A. G. Rempel, and W. Heidrich, "Hdr-vdp-2: a calibrated visual metric for visibility and quality predictions in all luminance conditions," in *ACM Transactions on Graphics (TOG)*, vol. 30, p. 40, ACM, 2011.

[15] Z. Wang and A. C. Bovik, "Modern image quality assessment," *Synthesis Lectures on Image, Video, and Multimedia Processing*, vol. 2, no. 1, pp. 1–156, 2006.

[16] M. Sattler, R. Sarlette, and R. Klein, "Efficient and realistic visualization of cloth," in *Rendering Techniques*, pp. 167–178, 2003.

[17] M. L. Koudelka, S. Magda, P. N. Belhumeur, and D. J. Kriegman, "Acquisition, compression, and synthesis of bidirectional texture functions," in *3rd International Workshop on Texture Analysis and Synthesis (Texture 2003)*, pp. 59–64, 2003.

[18] R. Furukawa, H. Kawasaki, K. Ikeuchi, and M. Sakauchi, "Appearance based object modeling using texture database: Acquisition compression and rendering.," in *Rendering Techniques*, pp. 257–266, Citeseer, 2002.

[19] G. Müller, J. Meseth, M. Sattler, R. Sarlette, and R. Klein, "Acquisition, synthesis, and rendering of bidirectional texture functions," in *Computer Graphics Forum*, vol. 24, pp. 83–109, Wiley Online Library, 2005.

[20] J. Y. Han and K. Perlin, "Measuring bidirectional texture reflectance with a kaleidoscope," *ACM Transactions on Graphics (TOG)*, vol. 22, no. 3, pp. 741–748, 2003.

[21] K. J. Dana and J. Wang, "Device for convenient measurement of spatially varying bidirectional reflectance," *JOSA A*, vol. 21, no. 1, pp. 1–12, 2004.

[22] C. Harris and M. Stephens, "A combined corner and edge detector.," in *Alvey vision conference*, vol. 15, pp. 10–5244, Citeseer, 1988.

[23] M. Goesele, W. Heidrich, and H.-P. Seidel, "Entropy-based dark frame subtraction," in *PICS*, pp. 293–298, 2001.

Multiview Layered Depth Image

Rafael dos Anjos
FCSH/IST
Av de Berna 26, 1.24
Portugal (1069-061),
Lisbon
rafael.kuffner@fcsh.unl.pt

João Madeiras Pereira
INESC-ID
R. Alves Redol 9
Portugal(1000-029),
Lisbon
jap@inesc-id.pt

José António Gaspar
ISR
Av. Rovisco Pais 1
Portugal (1049-001),
Lisbon
jag@isr.tecnico.ulisboa.pt

Carla Fernandes
FCSH
Av de Berna 26, 1.24
Portugal (1069-061),
Lisbon
fcar@fcsh.unl.pt

ABSTRACT

Layered Depth Images (LDI) compactly represent multiview images and videos and have widespread usage in image-based rendering applications. In its typical use case scenario of representing a scanned environment, it has proven to be a less costly alternative than separate viewpoint encoding. However, higher quality laser scanner hardware and different user interaction paradigms have emerged, creating scenarios where traditional LDIs have considerably lower efficacy. Wide-baseline setups create surfaces aligned to the viewing rays producing a greater amount of sparsely populated layers. Free viewpoint visualization suffers from the variant quantization of depths on the LDI algorithm, reducing resolution of the dataset in uneven directions. This paper presents an alternative representation to the LDI, in which each layer of data is positioned in different viewpoints that coincide with the original scanning viewpoints. A redundancy removal algorithm based on world-space distances as opposed to to image-space is discussed, ensuring points are evenly distributed and are not viewpoint dependent. We compared our proposed representation with traditional LDIs and viewpoint dependent encoding. Results showed the multiview LDI (MVLDI) creates a smaller number of layers and removes higher amounts of redundancy than traditional LDIs, ensuring no relevant portion of data is discarded in wider baseline setups.

Keywords

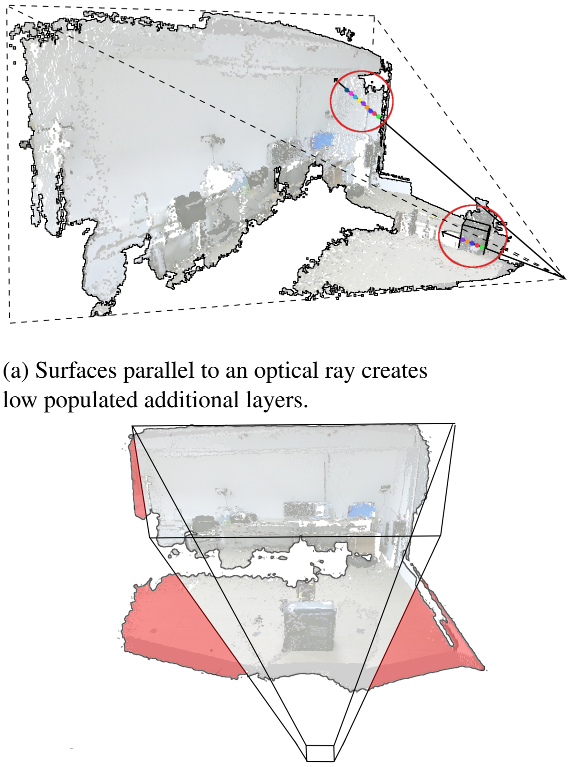
Video-based rendering, Image-based representation, Point clouds.

1 INTRODUCTION

Image-based representations for rendering were initially introduced as more efficient alternatives to render geometrically complex scenarios. More recently, they have been tightly connected to Video-based rendering (VBR), a growing field that has applications with high rendering and storing requirements. Multiview+depth (MVD) encoding and Layered Depth Images (LDI) [11] have been the most popular approaches in these scenarios, with LDI being a reportedly less costly representation [13]. They are popular in this scenario due to the fact that, being image-based, they can easily incorporate advances in video compression algorithms.

However, both of the proposed representations are targeted for applications with a predefined user paradigm (3DTV, head-face parallax) and a preferably narrow-baseline capture setup. Advances in 3D capturing technology have enabled developers to more accurately represent the real world, enabling less restrictive applications to emerge [10] mainly in the fields of virtual and mixed reality. While recent work has been focused on coding tools for MVD, which can be applied to these new scenarios, we consider an alternative representation for a single frame, enabling higher compression from the start of the process.

The key advantage of the LDI representation over its alternatives is the fact that redundancy between views can be minimized during the encoding process [4]. However, its classical representation of a central + residual viewpoints establishes a main viewing direction in which data can be optimally visualized, having disadvantages in a free camera navigation scenario. This has a direct effect in the redundancy computation, meaning that the sampling rate of the data is lower in the direction of the optical rays coming from the central viewpoint (Section 3.1). Moreover, on wide baseline scenarios where cameras might be in an opposite direction to the central viewpoint, it might not be possible for all data to be captured by one chosen central viewpoint (Figure 1b), requiring a more distant virtual viewpoint to be generated which, besides not being a trivial task, will decrease the sampling rate of the data. Lastly, parallel surfaces to the central viewpoint optical rays are intersected once in each layer, increasing the number



(a) Surfaces parallel to an optical ray creates low populated additional layers.

(b) Choice of LDI central viewpoint might not encode part of the data.

Figure 1: Problems with the classical LDI representation

of low populated residual layers in order to encode the whole dataset (Figure 1).

We propose a novel image-based representation and encoding algorithm which successfully answers the aforementioned problems: the Multiview Layered Depth Image (MVLDI), where each layer is encoded according to a different viewpoint among the several capturing positions. By utilizing a modified view-generation algorithm, we have better redundancy detection and a smaller number of generated layers, while being able to correctly encode data for a wide-baseline scenario, which was only possible using MVD and no redundancy estimation.

We tested our technique with different datasets, camera setups and redundancy detection techniques, always achieving higher compression rates than the alternatives (MVD, LDI) without discarding necessary data. MVLDI is shown to provide a more efficient representation for a single frame while being applicable to video scenarios, and correctly incorporates the advances in video compression technology and depth encoding.

This article reviews related work in image-based representations, followed by a description of the MVLDI generation algorithm and redundancy detection. Finally, results are presented with an in-depth comparison between our approach and the current image-based representations.

2 RELATED WORK

Regarding image-based representations for video-based rendering or depth image-based rendering, two main lines of research can be identified: multiview+depth coding and layered depth images or videos. This section will describe relevant work in both areas.

Merkle et al. [8] and Smolic et al. [12] introduce multiview+depth coding, where depth data is associated with the video and encoded as a video stream. Although compression artifacts can be found in the rendered results, the authors claim that intermediate views are more easily generated at the user side with full data. More recent work by Do et al. [2] proposes an inpainting algorithm which has a fast GPU implementation where holes are filled with the average in a 5x5 neighborhood.

Recent developments in this area have been related to depth coding; how to avoid the loss of precision due to compression, and how to use inter frame relations between depth values make better use of predictive frames. The two main strategies are independent [6] and texture-assisted [5] depth coding. Kim et al. [3] propose a method to evaluate how depth coding influences the quality of the results.

Merkle et al. [7] introduces a plane fitting approximation to simplify blocks of data, segmenting by contour. This can be used to extract meshes (using Delaunay triangulation on the edges of the contour) and to simplify data by simplifying geometric information, thus having a better encoding of the data.

Despite allowing effective 3D representations, multiview+depth coding implies dense representations not considering the redundancy always present when multiple cameras image the same scenario. The LDI concept was introduced to allow saving data by reducing redundancy.

Yoon et al. [13] applies the LDI concept for VBR, proving it to be more compact than standard multiview coding. The data size of the multiview video linearly increases with the number of cameras. The authors suggest using the LDI representation to compress and transmit this data, mainly due to the fact that any redundant data seen by more than one point of view is not transmitted. All optimizations applied to MVD can be applied to LDIs, while not including repeated points, a claim that is supported by Kirshantan et al. [4]. In their following work [14], improvements to the LDI representation are proposed, in particular layer aggregation and layer filling, so temporal coding has a better performance.

Muller et al. [9] discuss image-based representations in the context of 3DV systems and head motion parallax as an interaction paradigm. The authors claim that for stereo video, V+D is enough, while for several views, multiview+depth where only a subset of views with depth would be transmitted and intermediate views synthesized at the receiver side. They then introduce a different concept of LDI which is focused on 3DV systems. Instead of transmitting all the views, they transmit a central view close to the desired by the user, and residual information from side views to correct errors.

More recent work has focused on proposing better encoding for residual layers. Daribo and Saito [1] propose a different residual layer estimation algorithm which includes inpainting and hole-filling. Also, Kirshantan et al. [4] propose efficient encoding for this residual information including pre-processing the data for easier layer generation.

In our work we propose a novel image-based representation and encoding algorithm, which varies the viewpoint among the several capturing positions, in order to allow larger acquisition and visualization baselines.

3 DESCRIPTION

The LDI is represented by a set of M layers $L_{ldi} = \{l_1, l_2, ..., l_M\}$ with each layer being an image-based representation of the world as seen from the same chosen "central viewpoint" v_c among the set of acquisition viewpoints of the data $V = \{v_1, v_2, ..., v_N\}$.

We propose a different representation, where one MVLDI will consist of a set of M layers $L_{mvldi} = \{l_1, l_2, ..., l_M\}$ where each layer l_i represents one of the viewpoints v_i in $V = \{v_1, v_2, ..., v_N\}$. Each viewpoint v_i has its own intrinsic and extrinsic calibration parameters in order to generate the layer information.

The number of layers M has no direct relation to the N in the case of the LDI, while in our approach, M is typically equal to N, only being higher in the case of camera calibration misalignment.

The classical representation for LDI has two disadvantages in wider baseline capture scenarios, as exemplified in Figure 1. Encoding of parallel surfaces to optical rays, and excluding data from the process due to the central viewpoint choice. This does not happen in MVD encoding, due to the fact that encoding is performed according to different points of view. No data is discarded since it is seen at least once by its original recording viewpoint, and unpopulated layers are not created by parallel surfaces due to the fact that each subsequent layer will have optical rays in different directions. Our proposed representation MVLDI combines the advantages of both approaches, removing redundant data, and correctly encoding wide-baseline scenarios.

```
Input: Point Cloud $P$, Acquisition viewpoints $V$
Output: MVLDI $M$
$L \leftarrow$ set of layers for each viewpoint
$R \leftarrow$ empty cloud
$t \leftarrow$ threshold distance
for all point $p_i \in P$ do
    for all layer $l_j \in L$ do
        $d_{ij} \leftarrow$ worldToImageSpace( $p_i$, $l_j$ )
        $e_j \leftarrow l_j[\, d_{ij}.u,\ d_{ij}.v \,]$
        if isInside( $d_{ij}$, $l_j$ ) then
            if isRedundant( $d_{ij}$, $e_j$, $t$ ) then
                break
            else
                if $e_j$ is empty then
                    addToLayer( $d_{ij}$, $l_j$ )
                    break
                else
                    if $d_{ij}.d < e_{ij}.d$ then
                        replaceInLayer( $d_{ij}$, $e_j$, $l_j$ )
                        retryPoint( $e_j$ )
                        break
    addPoint($R, e_j$) //not encoded in any layer
if $R$ is not empty then
    re run with R
$M \leftarrow$ all non-empty layers
```

Algorithm 1: MVLDI encoding algorithm

3.1 Encoding algorithm

Our layer generation process is similar to the traditional LDI algorithm described by Shade et al. [11]. Recently, different processes have been proposed for this step [1, 4], which targeted optimizations for specific use case scenarios. The proposed hole-filling and inpainting techniques were targeted to a head-face parallax user interaction paradigm, where corrections can be made at image space, and assuming all pixels must be filled at all times. If such assumptions were done about the scenario to be used with MVLDI, these could be easily incorporated. However, we established a more general scenario, only assuming depth values were available per pixel, which is common nowadays using commodity depth cameras (e.g. MS-Kinect, Asus Xtion Pro, Intel RealSense).

Initially, each of the RGBD frames is transformed into a point cloud using the u, v image coordinates of each pixel, the depth value $d(u, v)$, and the intrinsic parameters of the capture device. All resulting clouds are then combined in a single volume P, using the transformations in the extrinsic matrix. Algorithm 1 describes the encoding process for a given volume P.

We create N data structures, one for each layer l_j, where N is the number of cameras in the capture process. Each one is positioned according to the extrinsic calibration parameters for each one of the viewpoints, and is sized according to the recording resolution of the input devices.

For each input point p_i, we try to encode it in a layer l_j by projecting it to that layer image space as a depth pixel d_{ij}. If $isInside(d_{ij}, l_j)$, meaning that (u,v) coordinates of d_{ij} are positive and smaller than the width and height of the layer, we check for redundancy $isRedundant(d_{ij}, e_j, t)$ (described in Section 3.2) where e_j is the depth pixel in layer l_j at the (u,v) coordinates of d_{ij}. In the case the data point is redundant, we start processing p_{i+1}, marking p_i accordingly, not including it in l_j. Otherwise, the point is encoded if the depth pixel e_{ij} is currently empty. If it is not, and d_{ij} has smaller depth than e_{ij}, we add d_{ij} to l_j, and e_{ij} is added for re-encoding. If e_{ij} is already filled with a point with smaller depth than d_{ij}, we go to the next layer.

Finally, in the case the point can not be placed in any of the layers, we add it to a collection R which will be processed with newly created layers. This is only the case when calibration errors exist. In all of the tested datasets these points were always encoded to a single layer, and represented less than 0.1% of the original point cloud. The final step will be adding all non empty layers to the MVLDI M.

3.2 Redundancy detection

In the classical LDI definition, a point P_{uvd} is considered to be redundant if $\|P_{uvd}.d - L_i[u,v].d\| < t$. Due to the fact that a central viewpoint is defined, and only depth comparisons are made in this particular image-space, sampling of data is not equal in all directions. When parallel to the viewing plane, pixel distance is evaluated, which in world coordinates is higher proportionally to the depth value and focal length f. When along the optical rays coming from the LDI viewpoint, a fixed threshold value t is used.

On a head-face parallax or 3DTV scenario, this was not seen as a problem, due to the fact that visualization is meant to be parallel to the central viewpoint. The sampling rate in the viewing direction is not perceived in these scenarios due to the visualization position restrictions. The lower sampling rate in the background pixels is also not noticed due to the fact that a parallel movement to the central viewpoint viewing plane does not change their image-space distance, not revealing possibly empty pixels.

Algorithm 2 shows our approach to the same problem. The first key aspect of our method is using world coordinates to estimate the distance between the pretended point and the correspondent point in the layer, opposed to just the depth value. The further away from the central viewpoint of an LDI, the greater the discrepancy between a depth-based threshold, and one based in Euclidean coordinates.

```
Input: $P_{uvd}, P_{xyz}$ point to be encoded in image and world coordinates
Output: true or false
$L \leftarrow$ current Layer
$t \leftarrow$ threshold distance
$f \leftarrow$ focal lenght from intr. matrix
$s \leftarrow \frac{P_{uvd}.d}{f}$
$n \leftarrow \frac{t}{s}$
$v_0 \leftarrow d_{ij}.v - n$
for $i = P_{uvd}.u - n$ to $i = P_{uvd}.u + n$ do
    for $i = P_{uvd}.v - n$ to $i = P_{uvd}.v + n$ do
        $Q_{uvd} \leftarrow L[i,j]$
        if $\|Q_{xyz} - P_{xyz}\| < t$ then
            return true
return false
```

Algorithm 2: Redundancy detection algorithm. n surrounding pixels are checked in order to properly evaluate all points that might be under the threshold distance t.

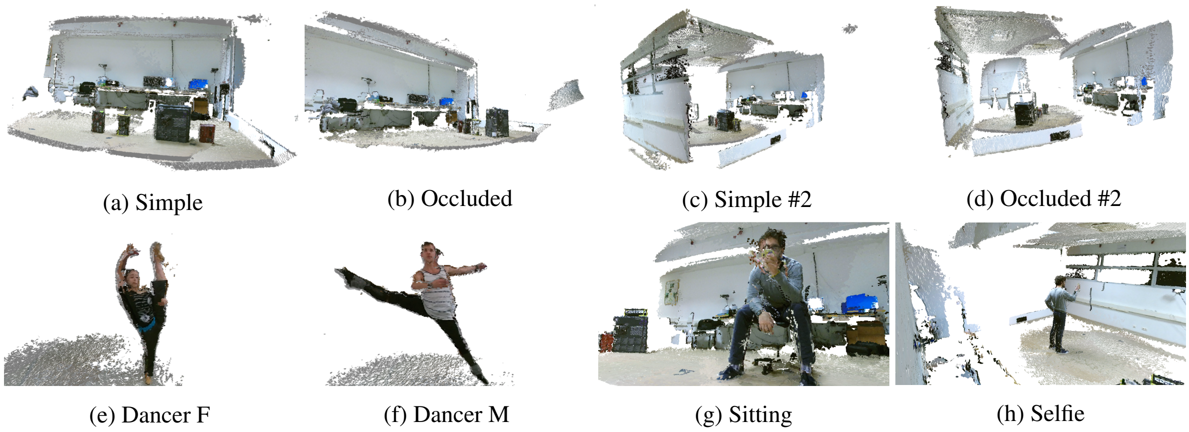
The second aspect is the fact that we also consider surrounding pixels to P_{uvd}. We first calculate the world distance s between pixels at that depth by calculating $\frac{P_{uvd}.d}{f}$ where $P_{uvd}.d$ is the depth value of the point being analysed, and f the focal distance from the intrinsics matrix. We then calculate the number of pixels to be checked around the encoded point by dividing the threshold t by s. By doing so, we very efficiently check every possible point that might be in a distance smaller than t from the considered point. Further layers do not need to be checked since the order of layer consideration is the same for each point, so in the case of a point not being considered redundant in that layer, we will naturally move to the next one to perform the same test.

Name	# Cameras	Baseline	Pt. number
Dancer M	3	Wide	68.932
Dancer F	3	Wide	75.146
Simple	4	Narrow	563.315
Simple #2	4	Wide	491.707
Simple #3	7	Wide	828.822
Occluded	4	Narrow	567.331
Occluded #2	4	Wide	505.232
Sitting	4	Narrow	580.736
Selfie	4	Wide	511.161

Table 1: Description of the tested datasets.

4 RESULTS

We tested our approach with varied datasets captured with the Microsoft Kinect sensor, each one holding different properties. Table 1 gives a brief description of each dataset, and Figure 2 shows a snapshot of each. We compared MVLDI with LDI using both the proposed global thresholding algorithm, and the traditional



(a) Simple (b) Occluded (c) Simple #2 (d) Occluded #2

(e) Dancer F (f) Dancer M (g) Sitting (h) Selfie

Figure 2: Snapshot of our used datasets. Simple #3 is the same as Simple #2 but having 3 more cameras.

image-based technique in order to individually evaluate the effect of each contribution. MVLDI with global thresholding represents our proposed technique, and LDI with image-based thresholding, the traditional LDI implementation. Our goal was to minimize the number of generated layers, while more effectively detecting redundant data. Computing time of the MVLDI is typically smaller since less layers are created. However, using a large threshold on the redundancy detection may increase computation time.

Both narrow and wide baseline were contemplated in order to validate our claim of MVLDI having a superior performance than LDI on a wide scenario. Different numbers of acquisition devices were also tested in order to validate the scalability of the process, and also validate the claim that redundancy is proportional to the number of capture devices.

Finally, while most of our datasets were controlled lab scenarios in order to control the disposition of the objects related to the cameras, we also included data captured from a realistic dance scenario (Figures 2e 2f). "Simple" datasets contained lines of boxes visible by most of the cameras (Figures 2a 2c) , "Occluded" datasets had a line of boxes occluded in one of the points of view (Figures 2b 2d). Sitting and Selfie (Figures 2g 2h) included a person in a controlled scenario in order to better evaluate if redundancy detection had a negative effect in the perception of more delicate shapes.

4.1 Quantitative Analysis

Figure 3 shows the achieved results regarding redundancy detection. The values in the table are calculated regarding the total number of points considered for encoding. Table 2 shows the percentage of the total dataset that was discarded in the case of LDI due to being outside the viewing volume of the central viewpoint.

Our approach had a superior performance over LDI in all tested datasets. Notably, the LDI approach had a high number of low populated layers (over 100 in Simple, Occluded #2 and Selfie, as seen in Table 3) This experimentally confirms the problems exemplified in Figure 1a, experimentally confirmed in Figure 5. With a wide-baseline capture setup, surfaces perpendicular to the central viewpoint viewing plane will create an elevated amount of created layers. Also, redundancy detection on the wide-baseline scenarios was lower with the classical approach. This is related to the amount of non-encoded points (over 40% of in some scenarios (Table 2, example in Figure1b) which would include walls and floor sections where typically a lot of redundancy was found, but also due to the redundancy detection algorithm.

The proposed global thresholding technique performed better in all scenarios. The difference was smaller in narrow scenarios, since the data was captured and sampled from the same perspective, so a pixel-based comparison still had an impact, albeit smaller. The biggest difference was found in wide baseline scenarios, where several points under the used threshold were located in neighboring pixels but not considered redundant only considering the depth value.

When comparing solely the difference in the number of points of view, the multiview approach had a smaller number of layers in all scenarios, being close or equal to the baseline for comparison (MVD, where the number of layers is equal to the number of input devices). The multiview approach did not suffer from the problem presented in Table 2, where big segments of the point cloud were discarded due to not being visualized by the central viewpoint. This is essential in a free visualization application through a virtual camera, specially in wide-baseline scenarios. Redundancy detection in MVLDI was higher in all scenarios except for the Dancer scenarios where only three cameras were used, and no data was left out of the encoding process.

An argument could be made about the choice of the central viewpoint for the LDI, and considering a virtually generated point of view that would include all of the

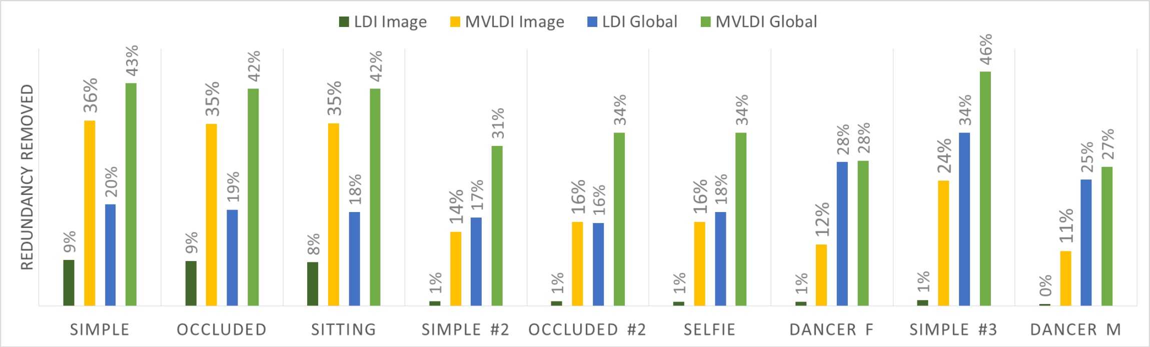


Figure 3: Percentage of data detected as redundant from the original cloud.

Dancer M	Dancer F	Simple	Simple #2	Simple #3	Occluded	Occluded #2	Sitting	Selfie
0%	0%	19%	48 %	39%	19%	46%	20 %	45%

Table 2: Percentage of points discarded by the LDI approaches due to being outside of the frustum of the central viewpoint.

data. The main problem with this proposition is the fact that a single viewpoint that includes all the data would necessarily be placed further away than the existing acquisition viewpoints, which certainly increases the problem in Figure 1a. The further the viewpoint is from the data, the more likely the planar surfaces in the cloud are aligned with the optical rays, increasing the number of low-populated layers.

Several previous works on LDI have reported higher rates of redundancy detection than the ones presented in this article [13]. The datasets typically used for benchmark are the breakdancers and ballet sequences from microsoft research, which are aimed to a head-face parallax interaction, or 3DTV. Cameras are separated by approximately 20cm from each other, which is a very narrow baseline. Also, depth precision is considerably lower, with only 256 values to represent the whole range of the scene. The depth cameras used in this work have a precision in the order of milimiters, which is why less values with the same depth are encountered using the image-based algorithm.

4.2 Qualitative Analysis

Although we report high redundancy removal in all of the presented scenarios, the quality of the visualization was not compromised. Figure 4 shows a side by side comparison of the input cloud, the redundant data, and the encoded result. On the dancer example where 27% redundancy was reported, data from the dancers back, silhouette, and floor were correctly reported as redundant (Figure 4b), not compromising the final visualization, as seen in Figure 4c.

On "Occluded" where a narrow baseline setup was used, we can clearly see a large amount of data (42%) consisted of walls, floor, and only the front part of the non-occluded box as being reported as redundant (Figure 4e), with the remainder of the data, included the boxes behind the front one, being perfectly visible in the encoded version (Figure 4f).

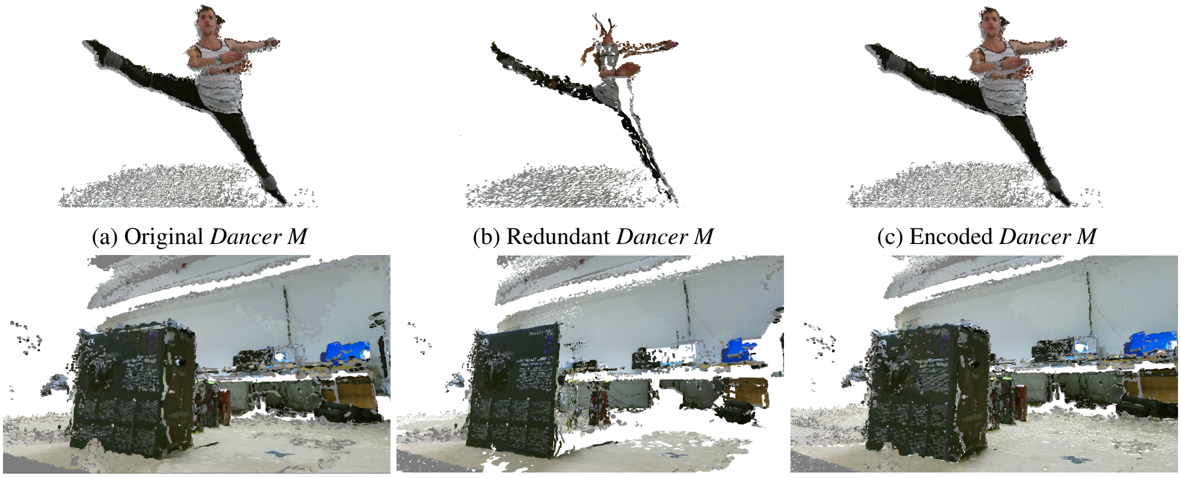
Also, considerations about the viability of MVLDI on a video scenario can be made. Our generated layers are less sparse as seen on figure 5, and in a smaller number, as seen in Table 3. We can more easily guarantee coherent matches between frames using block matching algorithms due to having more densely populated regions on the image. Also, our final number of layers is typically equal to the number of acquisition viewpoints, unlike LDI's which are inherently dependent on the content of the scene, and might create uneven distribution of layers per frame, which are less reliable for temporal matching.

Name	LDI Image	MVLDI Image	LDI Global	MVLDI Global
Dancer M	23	4	7	3
Dancer F	20	4	6	3
Simple	10	5	8	4
Simple #2	115	5	11	4
Simple #3	88	11	16	8
Occluded	13	5	12	5
Occluded #2	119	5	10	4
Sitting	17	5	12	5
Selfie	125	5	9	4

Table 3: Number of layers generated by each approach. MVLDI with global thresholding has the overall lower number of layers.

5 CONCLUSION

We presented MVLDI, an alternative data representation for a single frame of multiview video that allows wide-baseline VBR applications to take advantage of the redundancy detection of the LDIs. Moreover, MVLDI is also a more efficient alternative to represent a point cloud for an image-based rendering scenario.



(a) Original *Dancer M* (b) Redundant *Dancer M* (c) Encoded *Dancer M*

(d) Original *Occluded* (e) Redundant *Occluded* (f) Encoded *Occluded*

Figure 4: Result of the encoding process. Original (a, d), redundancy removed (b, e), and encoded data (c, f) for a wide and a narrow baseline scenario (top vs bottom row).

An alternative redundancy detection technique was introduced, which considers points in a global space opposed to the image-space thresholding of LDI, ensuring a homogeneous sampling of the data.

Our results showed that the proposed approach creates a smaller number of layers and detects redundancy at higher rates in both narrow and wide baseline scenarios, while also showing higher efficiency in scenarios with more cameras. The generated Layers are more dense than the typical residual LDI layers. They can be effectively used for temporal compression on video, and also geometry estimation, as proposed by [7].

Although results with a higher number of cameras were positive, the more cameras we have, the order of evaluation of viewpoints becomes more important. Our future work will be focused on evaluating the optimal order of encoding for the data, or the generation of alternatives viewpoints that provide an efficient encoding, and evaluating the application of temporal compression and depth encoding to our work. This technique will also be essential on encoding a point cloud that has no capture viewpoint information, such as a single sensor performing a sweep scan of large structures.

6 ACKNOWLEDGMENTS

This work was supported by the European Research Council under the project (Ref. 336200). This work was partially supported by national funds through FCT - Fundação para a Ciência e Tecnologia, under project PEst-OE/EEI/LA0021/2013 and by national funds through FCT with reference UID/CEC/50021/2013.

7 REFERENCES

[1] I. Daribo and H. Saito. A novel inpainting-based layered depth video for 3dtv. *Broadcasting, IEEE Transactions on*, 57(2):533–541, 2011.

[2] L. Do, G. Bravo, S. Zinger, and P.H.N. de With. Gpu-accelerated real-time free-viewpoint dibr for 3dtv. *Consumer Electronics, IEEE Transactions on*, 58(2):633–640, May 2012.

[3] W. S. Kim, A. Ortega, P. Lai, and D. Tian. Depth map coding optimization using rendered view distortion for 3d video coding. *IEEE Transactions on Image Processing*, 24(11):3534–3545, Nov 2015.

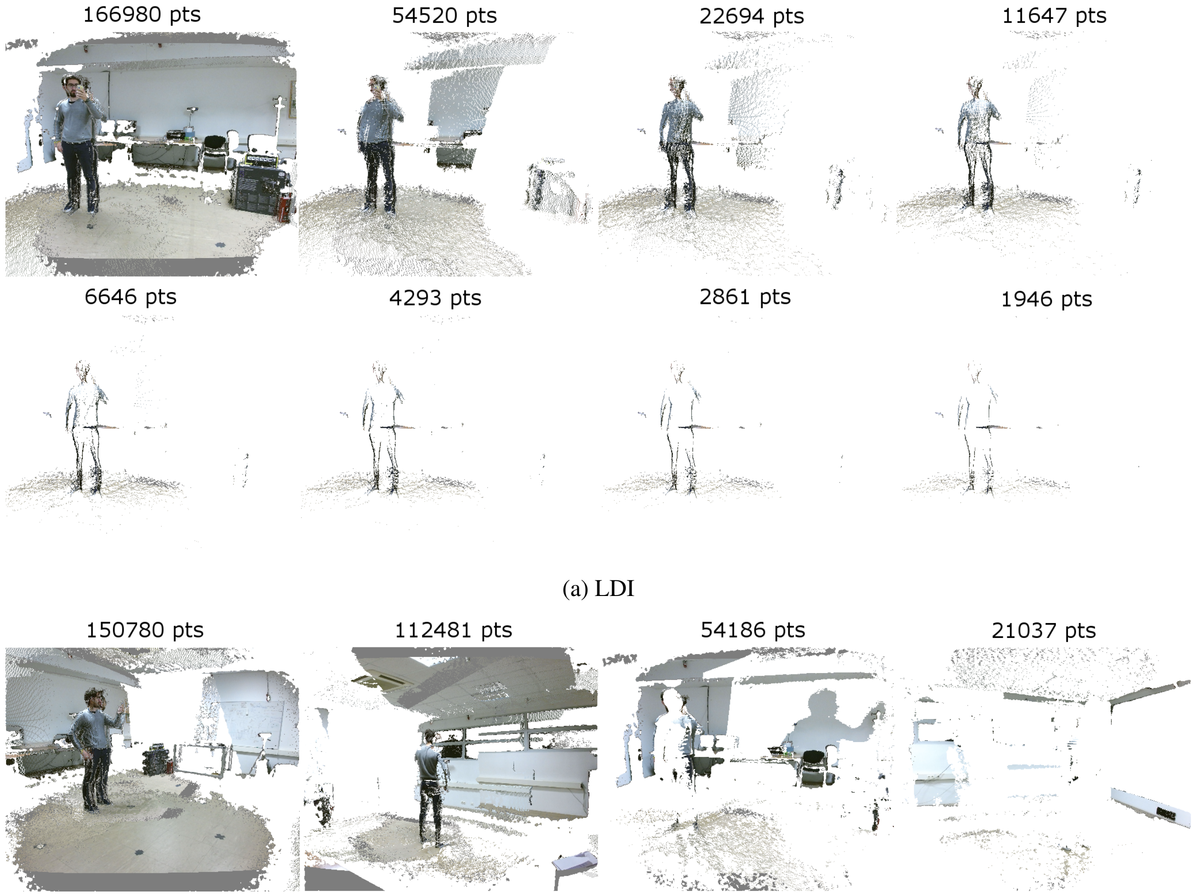
[4] S. Kirshanthan, L. Lajanugen, P.N.D. Panagoda, L.P. Wijesinghe, D.V.S.X. De Silva, and A.A. Pasqual. Layered depth image based hevc multi-view codec. In G. et al Bebis, editor, *Advances in Visual Computing*, volume 8888 of *Lecture Notes in Computer Science*, pages 376–385. Springer International Publishing, 2014.

[5] J. Lei, S. Li, C. Zhu, M. T. Sun, and C. Hou. Depth coding based on depth-texture motion and structure similarities. *IEEE Transactions on Circuits and Systems for Video Technology*, 25(2):275–286, Feb 2015.

[6] P. Merkle, Y. Morvan, A. Smolic, D. Farin, K. Müller, P.H.N. de With, and T. Wiegand. The effects of multiview depth video compression on multiview rendering. *Signal Processing: Image Communication*, 24(1-2):73 – 88, 2009. Special issue on advances in three-dimensional television and video.

[7] P. Merkle, K. Müller, D. Marpe, and T. Wiegand. Depth intra coding for 3d video based on geometric primitives. *IEEE Transactions on Circuits and Systems for Video Technology*, 26(3):570–582, March 2016.

[8] P. Merkle, A. Smolic, K. Müller, and T. Wiegand. Multi-view video plus depth representation and coding. In *Image Processing, 2007. ICIP 2007.*



(a) LDI

(b) MVLDI

Figure 5: Comparison between the first 8 LDI layers, and the full 4 MVLDI layers of the "Selfie" dataset, with number of points per layer. The silhouette of the subject, floor and a table in the background generate low populated layers being parallel to the optical rays.

IEEE International Conference on, volume 1, pages I – 201–I – 204, Sept 2007.

[9] K. Müller, A. Smolic, K. Dix, P. Kauff, and T. Wiegand. Reliability-based generation and view synthesis in layered depth video. In *Multimedia Signal Processing, 2008 IEEE 10th Workshop on*, pages 34–39, 2008.

[10] S. Orts-Escolano, C. Rhemann, S. Fanello, W. Chang, A/ Kowdle, Y. Degtyarev, D. Kim, P. L. Davidson, S. Khamis, M. Dou, V. Tankovich, C. Loop, Q. Cai, P. A. Chou, S. Mennicken, J. Valentin, V. Pradeep, S. Wang, S. B. Kang, P. Kohli, Y. Lutchyn, C. Keskin, and S. Izadi. Holoportation: Virtual 3d teleportation in real-time. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*, UIST '16, pages 741–754, New York, NY, USA, 2016. ACM.

[11] Jonathan Shade, Steven Gortler, Li-wei He, and Richard Szeliski. Layered depth images. In *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '98, pages 231–242, New York, NY, USA, 1998. ACM.

[12] A. Smolic, K. Muller, K. Dix, P. Merkle, P. Kauff, and T. Wiegand. Intermediate view interpolation based on multiview video plus depth for advanced 3d video systems. In *2008 15th IEEE International Conference on Image Processing*, pages 2448–2451, 2008.

[13] Seung-Uk Yoon, Eun-Kyung Lee, Sung-Yeol Kim, and Yo-Sung Ho. A framework for multi-view video coding using layered depth images. In *Advances in Multimedia Information Processing-PCM 2005*, pages 431–442. Springer, 2005.

[14] Seung-Uk Yoon, Eun-Kyung Lee, Sung-Yeol Kim, and Yo-Sung Ho. A framework for representation and processing of multi-view video using the concept of layered depth image. *The Journal of VLSI Signal Processing Systems for Signal, Image, and Video Technology*, 46(2-3):87–102, 2007.

Digital Transmission of Subjective Material Appearance

Rodrigo Martín Michael Weinmann Matthias B. Hullin
University of Bonn
Institute of Computer Science II
{rodrigo, mw, hullin}@cs.uni-bonn.de

ABSTRACT

The digital recreation of real-world materials has a substantial role in applications such as product design, on-line shopping or video games. Since decisions in design or shopping are often driven by qualities like “softness” or “beautifulness” of a material (rather than its photo-accurate visual depiction), a digital material should not only closely capture the texture and reflectance of the physical sample, but also its subjective feel. Computer graphics research constantly struggles to trade physical accuracy against computational efficiency. However, the connection between measurable properties of a material and its perceived quality is subtle and hard to quantify. Here, we analyze the capability of a state-of-the-art model for digital material appearance (the spatially-varying BRDF) to transport certain subjective qualities through the visual channel. In a psychophysical study, we presented users with measured material SVBRDFs in the form of rendered still images and animations, as well as photographs and physical samples of the original materials. The main insight from this experiment is that photographs reproduce better those qualities associated with the sense of touch, particularly for textile materials. We hypothesized that the abstraction of volumetric materials as opaque and flat textures destroys important visual cues especially in border regions, where fluff and protruding fibers are most prominent. We therefore performed a follow-up experiment where the border regions have been removed from the photographs. The fact that this step greatly reduced the capability of photos to transport important qualities suggests strong directions of future research in applied perception and computer graphics.

Keywords
Material perception; digital material appearance; SVBRDF; visual psychophysics

1 INTRODUCTION

The recent progress in the photo-realistic depiction of digitized materials has led to a paradigm change in important applications where the conventional approach of communicating objects in terms of photos taken by experts is more and more replaced by virtual surrogates. This methodology allows new possibilities such as cooperative product design, product advertising in prototype phase, exhibition of furniture or wearables in specific environments or visualization of cultural heritage objects. The entertainment industry has also drawn a major benefit from advanced digital material models as they allow a more realistic experience of virtual scenarios. While a remarkable reproduction quality has been achieved for virtual/digitized materials, there is still a gap in appearance between them and their physical counterparts which, in application, may distort the perception of the product. In particular, the accurate reproduction of surface reflectance behavior under varying illuminations and viewing conditions still remains a challenging task. For this reason, many material catalogs [Hal10] still opt for using pictures or even physical material samples to illustrate their collections instead of digitized models, despite the potential benefits that they entail.

In this paper, we aim at investigating this breach in appearance between digitized materials and their physical counterparts by analyzing how perceptual material information is transmitted through different stimuli. For this purpose, we consider the perception of materials by assessing a set of subjective qualities that can be assigned to either the tactile, visual or affective category, depending on the nature of the interaction that best reveals them. We then conducted a psychophysical study to compare the communication of these attributes based on different representations given by real material samples, photographs of these samples as well as static and animated renderings of the digitized materials represented by the spatially-varying BRDFs (SVBRDF) model [Nic77], which is deemed to be a standard representation in research and industry [Ell12]). The materials evaluated belonged to two semantic categories, leathers and fabrics. A key observation obtained from this experiment is that the SVBRDF model is not capable of preserving important qualities of material appearance, especially the tactile ones. Even a dynamic change of viewpoint does not seem to improve the perception of materials. Thus, the loss of information is presumably not caused by the limited resolution of the digitized samples but due to the abstractions intrinsic to the model. Upon closer inspection of the stimuli we observed that the differences between photos and virtual materials are most prominent at grazing angles, where the SVBRDF model fails to capture the volumetric material structure, intricate light scattering effects and the partial transparency of protruding fibers. Consequently, the perception of material properties such as softness, stiffness or transparency is not accurately recreated in the digitized representation, deviating from the correspondent photos and physical samples. With that in mind, we designed a follow-up study in which the respective border regions were digitally removed from the photos for a subset of relevant materials. Indeed, this step led to a significant deterioration in the transmission of tactile and affective properties, confirming our initial suppositions.

Our main findings are:

- Digitized materials (SVBRDF model) are not capable of adequately transmitting certain perceptual material information, being outperformed by simple photos of material samples. However, there is also an gap between photos and real materials.
- There are no significant differences in material-quality perception between static and animated digitized representations.
- The depictions of digitized materials suffer from a significant loss of information at grazing angles, where the SVBRDF model cannot represent appearance accurately.

To the best of our knowledge, this is the first perceptually-motivated work in evaluating how subjective material appearance is transmitted through digitized models (SVBRDF) in comparison to photos and real material samples. Conclusions from this set of studies are restricted to the given stimuli, but can provide useful insights for future research in developing realistic material representations.

2 RELATED WORK

In this section, we provide a condensed synopsis of research on the perception of subjective material qualities and the evaluation of digital material appearance models for graphics applications.

Perception of Materials and Their Qualities. The interest in unveiling the principles and reasons that determine the visual appearance of materials as well as in how humans visually perceive materials and their properties has received an increasing attention over the last years. Respective surveys [Ade01, And11] provide a discussion of the main problems and challenges in this area of research including the perception of material surface and properties. A further examination of the challenges in material perception is provided by [Fle14], where the author outlines a new theory of material perception based on 'statistical appearance models'. Among the phenomena that contribute to material appearance, glossiness has received a considerable amount of attention. Several approaches aimed at finding perceptually meaningful reparameterizations of material gloss by exploring the relationships between physical parameters and the perceptual dimensions of glossy appearance [Pel00, Wil09]. The human capability of perceiving material gloss (gloss constancy) under varying motion, disparity and color conditions was investigated by Wendt et al. [Wen10]. In addition to gloss, Ho et al. [Ho06] researched the visual estimation of surface roughness, discovering that its perception is strongly influenced by the illuminant angle.

Motion is another aspect that has an important impact on the appearance of materials' surface. By analyzing the optical flow, Doerschner et al. [Doe11] identified three motion cues, in which the brain could rely in order to identify material shininess. Our investigation further evaluates which additional subjective information (if any) is revealed by motion when compared to still renderings of digital materials. Other than motion, shape and geometry have proven to be critical aspects in the perception of materials [Van07]. The importance of the shape for material categorization is well-known [Ade01] and also can be used as an additional cue for material recognition [DeG16]. In this regard, one of the conclusions of our research is the emphasis on geometry and appearance under grazing angles, as a decisive feature to accurately assess material qualities. Indeed, the tasks of material categorization and material property judgment are closely related as demonstrated by Fleming et al. [Fle13]. Their studies revealed a high degree of consistency between these two assignments, implying that humans access similar information about materials when performing both tasks.

Although the experimental procedure initially involves purely visual stimuli, the participants also rated the same attributes for the real material samples in a sort of interaction that makes use of all senses (multimodal or full-modal interaction). The described approach relate to previous studies in multimodal material perception [Fuj15, Mar15], which highlight the importance of the tactile and auditory channels in the perception of material information. Their work also relies on ratings not only for surface material properties, but also a set of affective attributes.

Perceptual Evaluation of Material Appearance Models. Both material appearance acquisition and modeling have been deeply researched [Hai13, Wei16]. Widely used digital representations of materials exhibiting a spatially varying appearance include Spatially-Varying BRDFs (SVBRDFs) [Nic77] and Bidirectional Texture Functions (BTFs) [Dan97]. Both representations model material appearance depending on the spatial position $\mathbf{x}$ on the object surface, the light direction ω_i and the view direction ω_o. SVBRDFs allow a more compact modeling of surface reflectance behavior than BTFs at the cost of neglecting effects of light exchange at subtle surface structures. As SVBRDFs have become a standard in industry [Ell12], we use this representation to analyze differences in the human perception of real and digitized materials.

Regarding the perceptual evaluation of appearance models, several investigations focused on analyzing the level of realism achieved by a concrete model. In this context, Meseth et al. [Mes06] verified the ability of BTF models to achieve photo-realism in comparison to standard representations (BRDFs) and photographs, at a coarse and fine scale. It was demonstrated that BTF materials entail a significant increase of realism over BRDFs at both scales, albeit being still inferior to the scene photographs. A study from Filip et al. [Fil16] determined and predicted the critical viewing distances at which a certain BTF can be replaced by the correspondent BRDF representation without decreasing the overall visual impression. Additionally, Jarabo et al. [Jar14] examined the effects of approximate filtering on the appearance of BTFs in different domains (spatial, angular and temporal). The authors identified interesting correlations between high-level descriptors and perceptually equivalent levels of filtering as well as with low-level BTF statistics.

3 EXPERIMENT 1: METHODS

The proposed experiment investigates the performance of a well-known appearance model when transmitting subjective material qualities in comparison to equivalent photographs from real materials. In addition, the exercise examines whether the consideration of a higher spatial resolution through motion in digital scenes provides additional cues in the aforementioned task. Throughout this section the stimuli acquisition, selection of material qualities and experimental procedure will be detailed.

3.1 Stimuli

Selection of Materials In the scope of this research, we explore the perception of physical and affective material qualities for two semantic classes (leathers and fabrics). Restricting our selection to these two concrete, well-known categories allows us to keep the study and its conclusions manageable. Next, we have chosen ten material samples pertaining to these classes, each of them with an approximate size of 120×120 mm^2, with nearly flat geometry to match the requirements of the acquisition device, which is described in the following paragraphs. With this fine selection, we intended to maximize the relative intra-class heterogeneity not only in terms of the physical properties but also the aesthetic characteristics.

Photographs of Materials. In order to make the real and the virtual materials as comparable as possible, both the real and the virtual scene should share comparable geometry and illumination conditions. With that intention, our real scene was composed by a cardboard cylinder (80 mm diameter) to which the sample was attached. Cylindrical geometries have been frequently used in previous perceptual studies [Fil16] because of its well-defined texture mapping and for being one of the most discriminating shapes [Van07]. We covered the uppermost and lower part of the sample with white pieces of cardboard, which gently fixed the material to the cylinder. The height of the visible part of the sample along the vertical axis was approximately 90 mm. A reflecting sphere with a diameter of 50 mm was situated 10 mm right from the cylinder, and the whole setup was placed under natural illumination using a white, uniform piece of cloth as background. This arrangement is not arbitrary, given that during our internal tests we learned that subjects are more adept at this kind of subjective exercises when some context regarding the scene is provided. The complete setup can be observed in Figure 1, where the digital camera (Nikon 1 J5, resolution of 5568×3712 pixels) was situated at a distance of 280 mm in front of the material sample. We took a picture for each specimen while keeping the light and viewing conditions constant. The images were then corrected regarding white-balance, cropped and scaled to match the resolution of the final device (see Section 3.3). Moreover, during the photo session we used a remote-controlled 360° spherical panoramic camera (Ricoh Theta S) to probe the scene illumination. The resulting high-dynamic-range environment map was utilized to illuminate our virtual scenes.

Digitized Materials. The digitization of the material samples was carried out using a commercial scanning device [XR16] that allows the measurement of (flat) material samples. After taking images of the material sample from different viewpoints and under different illumination conditions, a surface normal map is obtained and the reflectance behavior is stored in terms of a Ward-SVBRDF. We refer to the supplementary material for more details on the material digitization process. The output format (AxF) is supported natively by several rendering applications such as Autodesk VRED, which was employed to generate the renderings used

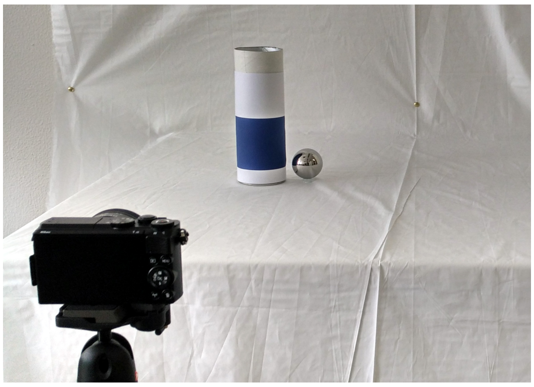


Figure 1: View of the photo setup

Tactile	Visual	Affective
rough–smooth hard–soft thick–thin stiff–flexible	shiny–matte bright–dark transparent–opaque homogeneous–heterogeneous	expensive–cheap natural–synthetic beautiful–ugly unrealistic–believable

Table 1: Opposite-meaning quality pairs

in this study. We approximated the geometry of the described photographic setup in a virtual scene and used VRED's Full Global Illumination algorithm to render it, lighting the scene with the previously calculated environment map. For the animated scene, we rotated the camera 60° back and forth around the cylinder in the Y-axis, and rendered the scene at 60 frames per second to get a clip with a duration of 4 seconds. The resulting photos and renderings are shown in Figure 2.

Real Materials. During the course of the experiment, we handed samples from the actual materials to the participants, hence, allowing a full-modal experience of the individual material qualities. Instead of the samples that were used for the acquisition, we used smaller portions of the same sample (approximately 70×70 mm^2) to avoid damaging the originals due to the interaction and in favor of the scalability of the process.

3.2 Selection of Material Qualities

In an initial step, we focused on finding a meaningful subspace of subjective adjectives that characterizes our selection of materials. The importance of this task was first addressed by Rao and Lohse [Rao96] for the concrete case of visual textures. We collected a list of 42 subjective material qualities organized in 21 opposite-meaning adjectives, which were observed to be the most recurrent ones in related literature regarding material perception [Fle13, Fuj15, Jar14, Mar15]. Such qualities were conceptually separated in three different groups with respect to their tactile, visual or affective nature. In pursuance of getting a smaller subspace of qualities that maximizes the transported information about our particular material collection, we conducted a pilot experiment in which we handed out the 10 original material samples to 7 participants along with a list of the 42 individual adjectives. The subjects were asked to mark the adjectives that better describe each sample. There was no restriction regarding the number of adjectives to choose. From the results, we selected the most voted attribute pairs in each of the three groups for our experiments, leading to a final assortment of 11 adjectives (see Table 1). Although it was not in our original list, we additionally included the pair 'unrealistic-believable', which provides information about the level of realism portrayed by the virtual materials.

3.3 Experimental Procedure

The user study was conducted using tablet computers (Toshiba Excite Pro 10.1, resolution of 2560×1600 pixels) running a custom Android application. This experimental setup makes our study scalable to larger surveys in addition to representative of contemporary consumer hardware. The procedure was carried out in a quiet, well-illuminated room and organized in sessions with a maximum of 7 participants. An introductory presentation was provided before performing the exercise to explain the procedure and clarify inquiries. Participants were instructed to infer the qualities which were not evidently revealed in a particular representation.

Different techniques were contemplated to perform perceptual quality ratings across our stimuli. Although double stimulus ratings or forced-choice pairwise comparisons may lead to smallest measurement variance, they would also increase the number of trials and, thus, make the whole study more difficult to accomplish. Therefore we decided to employ single stimulus ratings in which, for each stimulus, the subjects had to rate the selected qualities on a 7-point Likert scale characterized by a slider with values ranging from -3 to 3 (see supplementary material). Each of the values in the slider was consistently labeled with a term indicating the intensity of the stimuli in both axes (e.g., very bright, bright, a bit bright, neutral, a bit dark, dark, very dark). The actual procedure consisted of four different presentations or conditions, in which different material images were presented to the participants in randomized order along with the rating questionnaire. In addition, the participants had the chance to examine the real samples, serving the respective ratings as ground truth. The conditions that compose the experiment are illustrated in Figure 3 and listed below:

- Photographs (PH) taken from the real materials.
- Digitized static renderings (DR) from materials using the SVBRDF reflectance model.
- Digitized video renderings (DV) using the same reflectance model, where the camera rotates around the sample in the Y-axis.
- Full-modal condition (FM). Physical material samples were given to the participants so that they could interact with them.

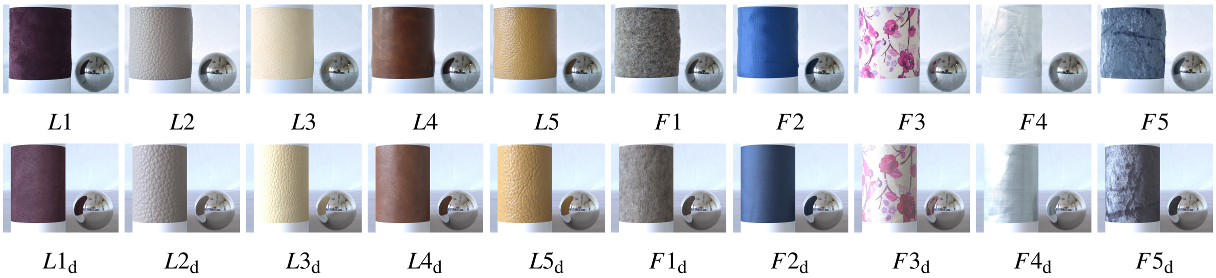


Figure 2: In the upper row, the pictures from the samples utilized in the study. In the lower row, the correspondent digitized material renderings. Larger stimuli images are provided in the supplementary material.

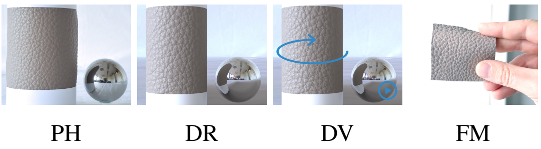


Figure 3: Stimuli presented in the psychophysical experiment corresponding to the four different conditions for an example material $L2$. From left to right, photograph (PH), digitized render (DR), digitized video (DV) and the physical sample (FM).

As the interaction with the real samples may bias the rest of the task, the condition FM was constrained to be the final one, while the order of the remaining conditions was randomized. Additionally, the application was instrumented to identify incorrect realizations of the assignment (e.g. skipping a material), in order to make the data more reliable. A total of 20 subjects (13 females, mean age 27.69; 7 males, mean age 27.00) participated voluntarily in the experiment. They were all naïve to the purpose of the experiment and reported normal or corrected-to normal visual acuity. They also provided informed consent and were compensated economically for their participation. From all the combinations of the conditions, materials, qualities and subjects, we obtained $4 \times 10 \times 12 \times 20 = 9600$ rating responses that are analyzed in the next section.

4 EXPERIMENT 1: RESULTS

To evaluate how the subjective attributes were perceived in the aforementioned material presentations we performed a conjoint analysis of the participants' preferences and non-parametric tests. The participants' rating responses were deemed reliable (Cronbach's $\alpha = .93$). In addition, participants' mean ratings and confidence intervals for each material and quality are included in the supplementary material.

4.1 Conjoint Analysis

With the purpose of gaining a general understanding regarding how material perception differs between the individual conditions, we made use of conjoint analysis techniques [Gre90] on the subjects' ratings. This method has been used extensively in market research to measure the preferences of the customers among multi-attributed products and services. In our experiment, we analyzed the three visual conditions C_i with $i \in \{\mathrm{PH}, \mathrm{DR}, \mathrm{DV}\}$ in the conjoint analysis and considered the following question: 'To what degree does condition C_i transmit the quality $q_k \in Q$ in comparison to the other conditions?'. Due to our experimental procedure based on single ratings, we cannot directly compare two conditions. Instead, we can evaluate them with respect to the full-modal representation (FM). From this, we can infer that a certain condition C_i is more suitable to represent an individual property than another C_j if the participants' ratings better agree to the ones obtained for the full-modal condition (FM). In contrast, if the ratings are distant, the depiction is less realistic and, consequently, less suitable.

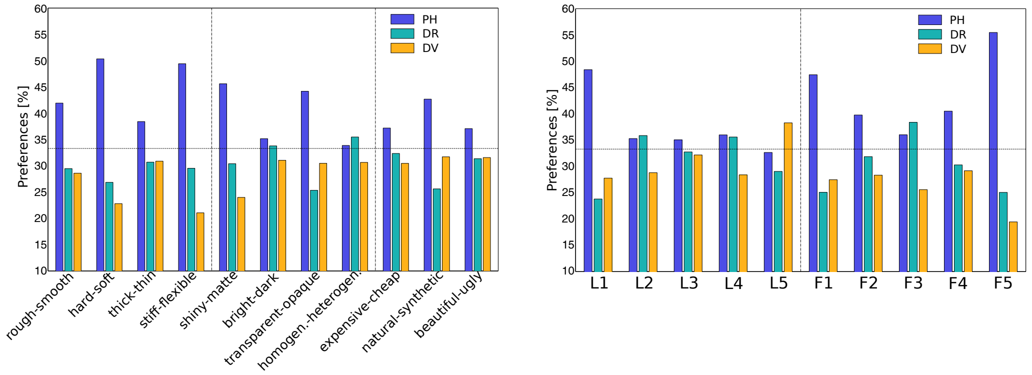
In order to carry out this comparisons, we use the weighed voting schema described in Martín et al. [Mar15] to compute the 'utility scores' (or 'part-worth utilities') s_i. For a certain combination of material, quality and subject, r_i and r_j denote the ratings for two particular conditions (C_i and C_j with $i, j \in \{\mathrm{PH}, \mathrm{DR}, \mathrm{DV}\}$) and r_{FM} denotes the ratings for the full-modal task which serves as ground truth. The calculated intermediate utility scores ($s_{i,j}$) are defined according to

$$s_{i,j} = \begin{cases} |r_{\mathrm{FM}} - r_i| - |r_{\mathrm{FM}} - r_j| & \text{if } |r_{\mathrm{FM}} - r_i| > |r_{\mathrm{FM}} - r_j| \\ 0 & \text{else} \end{cases} . \quad (1)$$

To compute the final utility scores s_i, and the normalized 'importance scores' $T = (t_i)$ for each condition, we consider the matrix composed of the calculated intermediate scores $S = (s_{i,j})$ with $S \in \mathbb{N}^{N \times N}$ and $s_{i,i} = 0$. Note that, in general, the matrix S is not symmetric. Then T is given by

$$T = \frac{\sum_i s_{i,j}}{\sum_{i,j} s_{i,j}} \quad \text{where} \quad i, j \in \{\mathrm{PH}, \mathrm{DR}, \mathrm{DV}\}. \quad (2)$$

The resulting scores, separated by property and material, can be seen in Figure 4. A clear evidence regarding



(a) Preferences arranged by quality

(b) Preferences arranged by material

Figure 4: Summary of the conjoint analysis revealing participants' preferences for each condition according to our voting schema. The preferences are separated by quality (left figure) and by material (right figure). The PH condition is preferred for almost every quality and performs particularly better for fabric materials.

the preference of a certain condition with respect to the other ones would imply that the respective condition depicts the reality more accurately, for the corresponding material or quality. Indeed, the obtained results indicate a clear predilection towards PH for almost all qualities and materials. This preference is particularly noticeable for the tactile adjective pairs (e.g. 'hard-soft', 'stiff-flexible') but can also be observed for visual properties (e.g. 'shiny-matte', 'transparent-opaque') and affective properties (e.g. 'natural-synthetic'). Considering the preferences organized per material, the condition PH is especially favored for fabric specimens. In fact, if applying conjoint analysis between the two material classes, the scores obtained for digitized leathers ($t_{\mathrm{PH}} = 38.09\%$, $t_{\mathrm{DR}} = 31.15\%$ and $t_{\mathrm{DV}} = 30.76\%$) are higher than the ones for digitized fabrics ($t_{\mathrm{PH}} = 45.26\%$, $t_{\mathrm{DR}} = 29.23\%$ and $t_{\mathrm{DV}} = 25.51\%$). Another interesting finding shows up when comparing the importance scores among static and dynamic renderings (DR and DV). Initially, the video presentation only performs better when transmitting transparency and naturalness. Applying conjoint analysis between DR and DV exclusively led to a more balanced overall preference of $t_{\mathrm{DR}} = 52.60\%$ and $t_{\mathrm{DV}} = 47.40\%$. The pair 'unrealistic-believable' does not apply to the real material stimuli and hence, it was not considered during the analysis.

This way, conjoint analysis provides insights regarding how well individual qualities are transmitted by the different conditions and, hence, which of the corresponding representations is most suitable. In the next section, we intend to additionally discover if and where significant differences among the ratings of the conditions are manifested.

4.2 Non-Parametric Tests

In addition to compare each condition against the ground truth (FM), we would also like to detect whether, and if so also where, meaningful discrepancies between the individual conditions occur. This may help us to understand how differently these representations transmit material qualities. A preliminary Shapiro-Wilk normality test determined that, for certain combinations of material and quality, our data do not come from a normally distributed population. This fact, together with the ordinal nature of the Likert scales, discredit analyses based on group means. Thus, we applied non-parametric tests (Friedman and Wilcoxon) in order to detect significant differences between the ratings (dependent variable) of the four conditions (independent variable).

Given our experimental design, we will be able to draw valid conclusions only for a single material-quality pair at a time ($p_{\mathrm{i}} = \{m_{\mathrm{j}}, q_{\mathrm{k}}\}$, given a material $m_{\mathrm{j}} \in M$ and quality $q_{\mathrm{k}} \in Q$), across all subjects. For better understanding, we first consider the pair p_{i} given by the combination of material $L1$ and the quality 'rough-smooth'. Applying Friedman's test revealed that the effect of the different conditions on the subjects' judgments is significant ($\chi^2(3) = 19.86, p < .05, r = .00$). The post-hoc analysis with the Wilcoxon signed-rank procedure resulted into rejecting the null hypotheses for the comparisons FM $\leftrightarrow$ DR, FM $\leftrightarrow$ DV and DV $\leftrightarrow$ PH, i.e. these representations have a significantly different effect on the participants' ratings. In contrast, the comparisons FM $\leftrightarrow$ PH, DV $\leftrightarrow$ DR and DR $\leftrightarrow$ PH showed no interaction effect on the ratings. In order to extend our findings to the complete collection of M materials and Q qualities, we performed the same analysis for each possible combination of material and quality $p_i \in P$. Then, we summed up the number of occurrences in which, for a particular p_i, we rejected the null hypotheses and, therefore, the ratings among conditions were determined to be significantly different (at least $p < .05$ for all the cases). We refer to this sum hereafter as the "dissimilarity score". Here, the presence of a high dissimilarity score between FM and another condition would outline how good or bad the respective depiction

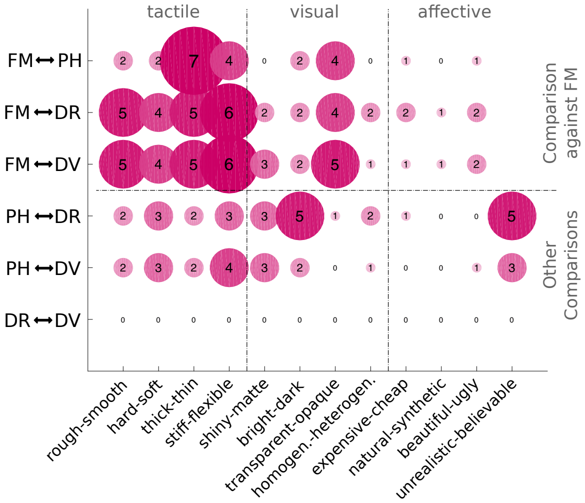


Figure 5: Number of tests with a significant effect on subjects' ratings (at least $p < .05$), summed along all ten materials. Each row compares two conditions while each column represents a quality pair. The largest scores are located in the second row (FM ↔ DR) and third row (FM ↔ DV), especially for tactile qualities. In contrast, there are no significant effects in the lower row (DR ↔ DV).

transmits real world information. In addition, by means of the same evidence for the rest of the scores, we may learn how differently photographs and digitized materials illustrate the individual qualities and if there is any significant impact in the ratings coming from motion. The outcome for all ten materials is separated by quality and shown in Figure 5.

As can be observed, the largest scores are mainly concentrated when the conditions FM ↔ DR and FM ↔ DV are compared, and this is especially appreciable for qualities categorized as tactile (upper-left quadrant). Besides, we can observe high scores between FM and the rest of the conditions for the adjective pairs 'thick-thin', 'stiff-flexible' and 'transparent-opaque'. This fact indicates that none of our representations is able to fully communicate these concrete qualities. Furthermore, the small scores found between the conditions FM ↔ PH in the remaining adjective pairs suggests that photos transmit most of the qualities good enough. In general, these results correlate well with the findings from the conjoint analysis, as they also tend to indicate the predominance of photographs over our digitized materials, especially in the tactile domain. In fact, the differences in the perceived realism ('unrealistic-believable' dimension) between PH and virtual materials confirm this trend. Finally, no significant dissimilarities were discovered in the comparison DR ↔ DV, i.e. the overall perception of material qualities is not affected by motion.

5 EXPERIMENT 2

During the course of the previous experiment, we observed that the samples with padded and fluffy appearance do not transmit appropriately material appearance in the digitized conditions and, hence, were deemed to be more unrealistic (see supplementary material). These features are more salient in the distinctive border regions, which possibly behaved as one of the main sources of information in favor of photographs. Due to the limited resolution of the reconstructed surface geometry, these structures are not accurately captured and the SVBRDF model is not capable of reproducing surface effects like self-occlusions, interreflections or transparency. To better understand how this matter influences the transmission of material appearance and which subjective attributes are most affected, we designed a follow-up study in which the perception of digitized materials was compared to the perception of real materials within photos where the border features have been digitally removed. The description of the experimental procedure and results are provided in the following sections. The supplementary material additionally provides the mean response ratings and confidence intervals for each material and quality.

5.1 Methods

From the materials selected for the previous experiment, we chose a subset of samples whose digitized stimuli were perceived to be particularly different from their correspondent photos in the experimental analysis, failing to transmit many of the considered attributes. According to this, we selected the set $M_2 = \{L1, L5, F1, F5\}$, where material $L5$ was only included to have an equal number of leathers and fabrics in the scope of this study. From the original photographs we removed the visible material borders from the cylindrical geometry to which the sample was attached using Adobe Photoshop, resulting into a flat silhouette shape as shown in Figure 6. Accordingly, we rendered again the digitized materials to match the new resolution from the cropped photographs. Other than that, we also aimed at comparing our visual stimuli against the real materials and we considered the same assortment of perceptual qualities as in the previous experiment. Nevertheless, in this experiment no motion was included, i.e. the considered conditions are:

- Cropped photographs (PH_c) taken from the real materials, where the borders have been removed.
- Digitized static renderings (DR) from materials using the SVBRDF reflectance model.
- Full-modal condition (FM). Physical material samples were given to the participants so that they could interact with them.

Again, the order of the materials and conditions was randomized except for FM, which was constrained to be

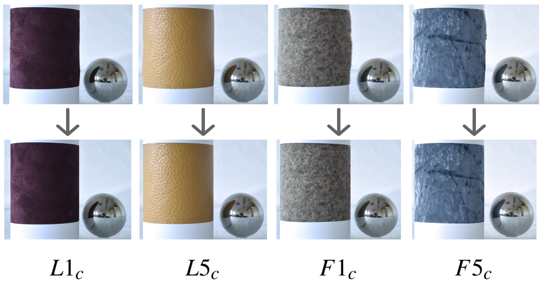


Figure 6: Pictures from the two leathers ($L1$, $L5$) and two fabrics ($F1$, $F5$) selected for the follow-up experiment, before and after crop operation.

the last one. 19 subjects (12 females mean age 27.08; 7 males, mean age 28.57) took part in the experiment under the same conditions as the previous one. The resulting $3 \times 4 \times 12 \times 19 = 2736$ rating responses are evaluated in the next section.

5.2 Results

In the following, we show the outcome of performing conjoint analysis and non-parametric tests on the subjects' ratings and compare them w.r.t. the results of the previous experiment. A Cronbach's alpha value of $\alpha = .89$ confirms the reliability of the ratings.

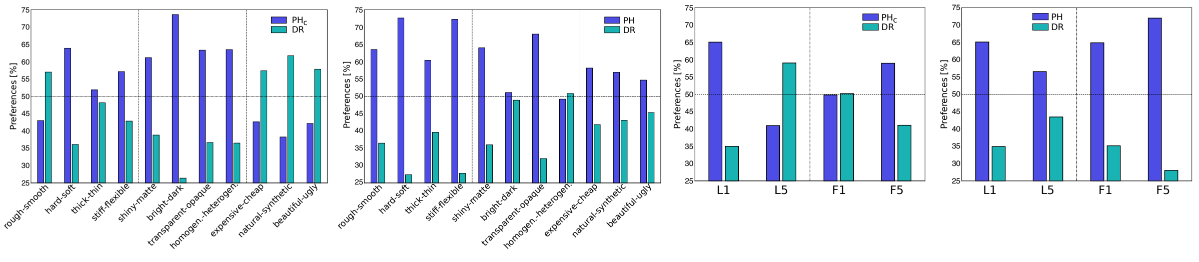
Conjoint Analysis Similar to Section 4.1 we performed a conjoint analysis in order to reply the question: 'To what degree do the conditions PH_c and DR transmit the quality $q_\text{k} \in Q$ in comparison to FM?'. Figure 7a illustrates the importance scores per quality for the current experiment, in which the borders were removed, while Figure 7b shows the scores obtained for Experiment 1, if only the data from the subset M_2 of materials were taken into account. Direct comparison between the scores corresponding to the conditions PH_c (Experiment 2) and PH (Experiment 1) reveals how the preferences for the cropped photographs become significantly smaller for all the tactile and affective attributes so that, for certain cases, these are surpassed by the DR scores. Certainly, the score difference between conditions PH and PH_c in both experiments should be a good indicator regarding which perceptual attributes were most damaged with the border-feature removal. According to this, the most deteriorated pair was 'rough-smooth' (−20.52%), followed by 'natural-synthetic' (−18.63%), 'expensive-cheap' (−15.55%) and 'stiff-flexible' (−15.10%). Contrarily, the pairs 'bright-dark' (+22.50%) and 'homogeneous-heterogeneous' (+14.30%), were surprisingly better communicated without the borders. In this case, the silhouette information present in the photos from Experiment 1 could have acted as a misleading cue to judge homogeneity and brightness. Finally, the importance scores separated by material are shown in Figure 7c. When compared to the scores obtained in Experiment 1 (Figure 7d), we notice substantial changes as the preferences for PH_c diminish in favor of the ones for the DR condition except for the material $L1$, whose scores remain relatively constant.

Non-Parametric Tests As in our previous study, we perform non-parametric tests (Friedman and Wilcoxon) to detect meaningful differences among the respective ratings (dependent variable) for the three conditions PH_c, DR and FM (independent variable). Anew, we carried out multiple comparison tests between the conditions (applying Wilcoxon signed-rank procedure) and generalized our findings by summing the resulting occurrences of rejected null hypotheses for each material-quality pair $p_i \in P$. The resulting dissimilarity scores are displayed in Figure 8a together with the scores resulting when applying the same test in Experiment 1 (Figure 8b), for the material subset M_2. the DV condition was ignored as the video stimuli were not used in the follow-up study.

From the results depicted in the figures, we can outline three main observations. First, the large dissimilarity scores found in the bottom comparison PH $\leftrightarrow$ DR for Experiment 1 have disappeared when moving to $PH_c \leftrightarrow$ DR in Experiment 2, which suggests that both representations lie much closer in the follow-up study. Second, the middle row comparing FM $\leftrightarrow$ DR only contains subtle changes in the scores obtained for both experiments. This fact is coherent with the stimuli as these conditions have not changed between experiments. Third, the top row comparing FM $\leftrightarrow PH_c$ in Experiment 2 presents, for most of the considered qualities, higher scores as in the original study. This fact suggests that the perception of photographs and real materials differs more significantly when the silhouette-border information is not present. However, the pair 'thick-thin' displays an unexpected opposite tendency. Again, borders may have acted as a misleading cue to judge thickness on these concrete samples.

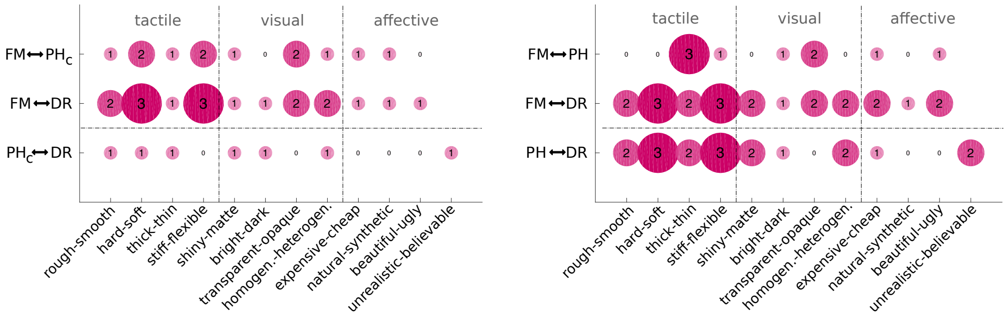
6 CONCLUSIONS

In the scope of this investigation, we have studied the perceptual differences between stimuli based on standard digital material appearance models in terms of Spatially Varying BRDFs, photos of real materials (leathers and fabrics) and the actual material samples on the task of transmitting a rigorously selected group of subjective qualities. Additionally, we explored the effect of motion on the perception of the stimuli based on digitized material representations. Because of the observation that the appearance of photographed materials and their digitized counterparts differ particularly at the material borders, a second experiment was designed to explore to what degree the appearance of materials under flat viewing angles could cause the loss of information between photographs and renderings.



(a) Prefs. by quality (Exp. 2) (b) Prefs. by quality (Exp. 1, M_2) (c) Prefs. by material (Exp. 2) (d) Prefs. by material (Exp. 1, M_2)

Figure 7: Summary of the conjoint analysis showing the participants' preferences for each condition in Experiment 2, where the material borders were removed from the photos, in contrast to Experiment 1, separated by quality (left figures) and material (right figures). The lower scores for the condition $\mathrm{PH_c}$ in comparison to PH show how the transmission of tactile and affective qualities as well as fabric samples deteriorates when the borders are removed.



(a) Dissimilarity score (Exp. 2) (b) Dissimilarity score (Exp. 1)

Figure 8: N° of tests with a significant effect on subjects' ratings (at least $p < .05$), summed along the subset M_2 of four materials. On the left, the dissimilarity scores for Experiment 2, where the borders were removed from the photos. On the right, the respective scores for Experiment 1. Note the high scores in the comparison PH $\leftrightarrow$ DR for Experiment 1, whereas they partially move to the first row FM $\leftrightarrow$ PH_c in Experiment 2. Meanwhile the middle row presents little variation.

One of the main findings of our investigations is that the considered digitized models are not able to fully transmit basic subjective properties according to the reality. Most of the analyzed perceptual qualities were better perceived in photos of real materials in comparison to renderings, but there is also a perceptual gap between photos and physical materials. This effect has proven to be true especially, but not exclusively, for tactile attributes and the fabric samples. Furthermore, motion information did not affect the perception of digitized materials significantly. The latter is especially relevant for the 'shiny-matte' dimension as they may contradict the documented fact that motion cues can override static ones while judging shininess [Doe11, Wen10]. Nevertheless, their experiments are based in much simpler appearance models (isotropic Ward model and grayscale Phong model respectively) which probably led to a better shininess isolation and recognition. Finally, our investigations indicate that more attention has to be paid to the accurate reconstruction of the distinctive material geometry as well as the acquisition of material appearance under grazing angles. In our measurements, the lowest camera was mounted with a zenith angles of 67.5° and, hence, these particular appearance effects cannot be recovered.

Although our studies provide interesting evidences, they cannot be extrapolated to other material categories (e.g. paper, stone, wood, etc.) for which additional experiments would have to be performed. We also acknowledge certain aspects that could have limited the expressiveness of the digitized materials used in our experiments, including:

- The generation of the virtual scene is approximate, i.e. the virtual camera position and the scene geometry slightly deviate from their physical counterparts.
- Scale differences between virtual and real material sample. Due to restrictions of the acquisition process, the digitized material represents a slightly smaller patch from the original one.
- The environmental light varied during the photo session due to the movement of sun and clouds.
- A color shift between the real and the virtual materials which also comes from the acquisition process.
- Not all the materials presented in this study were suitable to be represented by the SVBRDF appearance model, since it does not account for important surface effects. Consequently, some digitized representations were visibly defective.

By and large, we consider the results presented in this investigation an important step in the immense task of unveiling the perception of digital environments to improve the overall experience. To conclude, we point out the necessity of research in several directions such as the application of more appropriate, material-specific appearance representations. In this regard, BTFs models might help regarding the reproduction of fine effects of light exchange within the digital material representation at the cost of rather long acquisition times. Another interesting avenue of research could be to explore the linkage between perceptual qualities and physical measurable material properties (i.e. stiffness or roughness). Finally, the transmission of material qualities could benefit from a multisensory approach. In particular, the use of sound has proven to be beneficial for the assessment of tactile qualities [Mar15], which were not successfully transmitted using purely visual models.

7 ACKNOWLEDGMENTS

This work was developed in the X-Rite Graduate School on Digital Material Appearance at the University of Bonn. We greatly thank Max Hermann and Christopher Schwartz for their assistance during the material acquisition process.

8 REFERENCES

[Ade01] Adelson, E. On seeing stuff: the perception of materials by humans and machines. In Proc. SPIE, 4299:pp. 1–12, 2001.

[And11] Anderson, B. Visual perception of materials and surfaces. In Current Biology, 21(24):pp. R978 – R983, 2011.

[Dan97] Dana, K. J., Nayar, S. K., van Ginneken, B., and Koenderink, J. J. Reflectance and texture of real-world surfaces. In *Procc of the Conf. on Computer Vision and Pattern Recognition (CVPR)*, pp. 151–157. 1997.

[DeG16] DeGol, J., Golparvar-Fard, M., and Hoiem, D. Geometry-informed material recognition. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pp. 1554–1562. IEEE, 2016.

[Doe11] Doerschner, K., Fleming, R., Yilmaz, O., Schrater, P., Hartung, B., and Kersten, D. Visual motion and the perception of surface material. In Current Biology, 21(23):pp. 2010 – 2016, 2011.

[Ell12] Ellens, M. and Lamy, F. From color to appearance in the real world. In *Proc. SPIE*, vol. 8291, pp. 82910B–82910B–8. 2012.

[Fil16] Filip, J., Vávra, R., Havlíček, M., and Krupička, M. Predicting visual perception of material structure in virtual environments. In Computer Graphics Forum, 2016.

[Fle13] Fleming, R., Wiebel, C., and Gegenfurtner, K. Perceptual qualities and material classes. In Journal of Vision, 13(8):p. 9, 2013.

[Fle14] Fleming, R. W. Visual perception of materials and their properties. In Vision Research, 94:pp. 62 – 75, 2014.

[Fuj15] Fujisaki, W., Tokita, M., and Kariya, K. Perception of the material properties of wood based on vision, audition, and touch. In Vision Research, 109:pp. 185 – 200, 2015.

[Gre90] Green, P. and Srinivasan, V. Conjoint analysis in marketing: New developments with implications for research and practice. In Journal of Marketing, 54(4):pp. 3–19, 1990.

[Hai13] Haindl, M. and Filip, J. *Visual texture: Accurate material appearance measurement, representation and modeling*. Springer Science & Business Media, 2013.

[Hal10] Hallett, C. and Johnston, A. *Fabric for Fashion: The Swatch Book*. Laurence King Publishing, 2010.

[Ho06] Ho, Y., Landy, M., and Maloney, L. How direction of illumination affects visually perceived surface roughness. In Journal of Vision, 6(5):p. 8, 2006.

[Jar14] Jarabo, A., Wu, H., Dorsey, J., Rushmeier, H., and Gutierrez, D. Effects of approximate filtering on the appearance of bidirectional texture functions. In IEEE Trans. on Visualization and Computer Graphics, 20(6):pp. 880–892, 2014.

[Mar15] Martín, R., Iseringhausen, J., Weinmann, M., and Hullin, M. Multimodal perception of material properties. In *ACM SIGGRAPH Symposium on Applied Perception*, pp. 33–40. ACM, 2015.

[Mes06] Meseth, J., Müller, G., Klein, R., Röder, F., and Arnold, M. Verification of rendering quality from measured btfs. In *Proc. of the 3rd Symposium on Applied Perception in Graphics and Visualization*, pp. 127–134. ACM, 2006.

[Nic77] Nicodemus, F. E., Richmond, J. C., Hsia, J. J., Ginsberg, I. W., and Limperis, T. Geometrical considerations and nomenclature for reflectance. National Bureau of Standards Monograph #160, U.S. Dept. of Commerce, 1977.

[Pel00] Pellacini, F., Ferwerda, J., and Greenberg, D. Toward a psychophysically-based light reflection model for image synthesis. In *Proc. of SIGGRAPH*, pp. 55–64. 2000.

[Rao96] Rao, A. and Lohse, G. Towards a texture naming system: Identifying relevant dimensions of texture. In Vision Research, 36(11):pp. 1649 – 1669, 1996.

[Van07] Vangorp, P., Laurijssen, J., and Dutré, P. The influence of shape on the perception of material reflectance. In ACM Trans. Graphics, 26(3), 2007.

[Wei16] Weinmann, M., Langguth, F., Goesele, M., and Klein, R. Advances in geometry and reflectance acquisition. In *Eurographics - Tutorials*. The Eurographics Association, 2016.

[Wen10] Wendt, G., Faul, F., Ekroll, V., and Mausfeld, R. Disparity, motion, and color information improve gloss constancy performance. In Journal of Vision, 10(9):p. 7, 2010.

[Wil09] Wills, J., Agarwal, S., Kriegman, D., and Belongie, S. Toward a perceptual space for gloss. In ACM Trans. Graphics, 28(4):pp. 103:1–103:15, 2009.

[XR16] X-Rite. Tac7 scanner. http://www.xrite.com/categories/Appearance/tac7, 2016. Accessed at 25th April 2017.

From Contours to Ground Truth: How to Evaluate Edge Detectors by Filtering

Hasan Abdulrahman,
Ecole des Mines d'Alès,
6 avenue de Clavières
30319 Alès, France

Baptiste Magnier
Ecole des Mines d'Alès,
6 avenue de Clavières
30319 Alès, France

Philippe Montesinos
Ecole des Mines d'Alès,
6 avenue de Clavières
30319 Alès, France

ABSTRACT

Edge detection remains a crucial stage in numerous image processing applications. Thus, an edge detection technique needs to be assessed before use it in a computer vision task. As dissimilarity evaluations depend strongly of a ground truth edge map, an inaccurate datum in terms of localization could advantage inaccurate precise edge detectors or/and favor inappropriate a dissimilarity evaluation measure. Hence, in this work, we demonstrate how to label these ground truth data in a semi-automatic way. Moreover, several referenced-based boundary detection evaluations are detailed and applied toward an objective assessment. Thus, each measure is compared by varying the threshold of the thin edges. Indeed, theoretically, the minimum score of the measure corresponds to the best edge map, compared to the ground truth. Finally, experiments on many images using six edge detectors show that the new ground truth database allows an objective comparison of numerous dissimilarity measures.

Keywords
Edge detection, ground truth, supervised evaluation, distance measure, objective evaluation.

1 INTRODUCTION

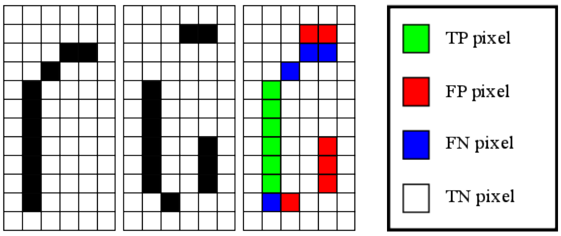
Over the last decades, edge detection remains a crucial role in the computer vision community [30][28][1][42][4][41][8]. This segmentation is considered as a fundamental step in many image processing applications or analysis, pattern recognition, as well as in human vision. Moreover, contours include the most important structures in the image. Typically, edges occur on the boundary between two different regions in an image. In other words, an edge is the boundary between an object and the background or between two different objects.

There exist many different edge detection methods. Nevertheless, an important problem in image processing remains an efficient edge detector comparison and which parameter(s) correspond(s) to the best setting to obtain an accurate edge detection results. Indeed, a robust boundary detection method should create a contour image containing edges at their correct locations with a minimum of misclassified pixels. In order to objectively quantify the performance of an edge detector, a supervised measure computes a similarity/dissimilarity between a segmentation result and a ground truth obtained from synthetic data or a human judgment [2].

In this paper, we detail several edge dissimilarity measures and present how to evaluate filtering edge detection technique involving these considerate measures. In a second time, we demonstrate how to build a new ground truth database which can be used in supervised contour detection evaluation. Indeed, results presented show the importance of the choice of the ground truth. Finally, considering these new ground truth images, results obtained by the measures are exposed.

2 SUPERVISED ERROR MEASURES

To assess an edge detector, the confusion matrix remains a cornerstone in boundary detection evaluation methods. Let G_t be the reference contour map corresponding to ground truth and D_c the detected contour map of an original image I. Comparing pixel per pixel G_t and D_c, the first criterion to be assessed



(a) G_t (b) D_c (c) $G_t \cup D_c$ (d) Legend for (c)

Figure 1: Ground truth vs. desired contour. In (b), D_c is contaminated with 6 FPs ans 4 FNs.

Table 1: List of error measures involving only statistics.

Complemented $Performance\ measure$ [3] [4]
$P_m^*(G_t,D_c) = 1 - \frac{TP}{TP+FP+FN}$

Complemented Φ measure [5]
$\Phi^*(G_t,D_c) = 1 - \frac{TPR \cdot TN}{TN+FP}$

Complemented χ^2 measure [6]
$\chi^{2*}(G_t,D_c) = 1 - \frac{TPR-TP-FP}{1-TP-FP} \cdot \frac{TP+FP+FPR}{TP+FP}$

Complemented F_α measure [7]
$F_\alpha^*(G_t,D_c) = 1 - \frac{PREC \cdot TPR}{\alpha \cdot TPR + (1-\alpha) \cdot PREC}$, with $PREC = \frac{TP}{TP+FP}$ and $\alpha \in]0;1]$

is the common presence of edge/non-edge points, as illustrated in Fig. 1. A basic evaluation is compounded from statistics; to that effect, G_t and D_c are combined. Afterwards, denoting $|\cdot|$ as the cardinality of a set, all points are divided into four sets:

- **True Positive points (TPs)**, common points of G_t and D_c: $TP = |D_c \cap G_t|$,
- **False Positive points (FPs)**, spurious detected edges: $FP = |D_c \cap \neg G_t|$,
- **False Negative points (FNs)**, missing boundary points of D_c: $FN = |\neg D_c \cap G_t|$,
- **True Negative points (TNs)**, common non-edge points of G_t and D_c: $TN = |\neg D_c \cap \neg G_t|$.

Computing only FPs and FNs enables a segmentation assessment to be performed [8]. The complemented *Performance measure* P_m^* presented in Table 1 considers directly and simultaneously the three entities TP, FP and FN to assess a binary image [3] [4]. The measure is normalized and decreases with improved quality of detection, with $P_m^* = 0$ qualifying perfect segmentation.

By combining FP, FN, TP and TN, another way to display evaluations is to create Receiver Operating Characteristic (ROC) [9] curves or Precision-Recall (PR) [7], involving *True Positive Rates* (TPR) and *False Positive Rates* (FPR): $TPR = \frac{TP}{TP+FN}$ and $FPR = \frac{FP}{FP+TN}$. Derived from TPR and FPR, the three measures Φ, χ^2 and F_α (detailed in Table 1) are frequently used in edge detection assessment. Using the complement of these measures, a score close to 1 indicates a poor segmentation, whereas a value close to 0 a good segmentation. Among these three measures, F_α remains the most stable because it does not consider the TNs, which are dominant in edge maps. Indeed, taking into consideration TN in Φ and χ^2 influences solely the measurement (as is the case in huge images).

These measures evaluate the comparison of two edge images, pixel per pixel, tending to severely penalize a (even slightly) misplaced contour, as illustrated in Fig. 2 (g) and (h). Thus, to perform an edge evaluation, the assessment should penalize a misplaced edge point proportionally to the distance from its true location.

Table 2: List of normalized error measures compared in this work, with the parameter $\kappa \in]0;1]$.

Figure of Merit (FoM) [10]
$FoM(G_t,D_c) = 1 - \frac{1}{\max(\lvert G_t\rvert, \lvert D_c\rvert)} \cdot \sum_{p \in D_c} \frac{1}{1+\kappa \cdot d^2_{G_t}(p)}$

FoM of over-segmentation [11]
$FoM_e(G_t,D_c) = 1 - \frac{1}{\max(e^{-FP}, FP)} \cdot \sum_{p \in D_c \cap \neg G_t} \frac{1}{1+\kappa \cdot d^2_{G_t}(p)}$

FoM revisited [12]
$F(G_t,D_c) = 1 - \frac{1}{\lvert G_t \cup D_c\rvert} \cdot \sum_{p \in G_t} \frac{1}{1+\kappa \cdot d^2_{D_c}(p)}$

Combination of FoM and statistics [13]
$d_4(G_t,D_c) = \frac{1}{2} \cdot \sqrt{S + FoM(G_t,D_c)}$ with $S = \frac{(TP - \max(\lvert G_t\rvert, \lvert D_c\rvert))^2 + FN^2 + FP^2}{(\max(\lvert G_t\rvert, \lvert D_c\rvert))^2}$

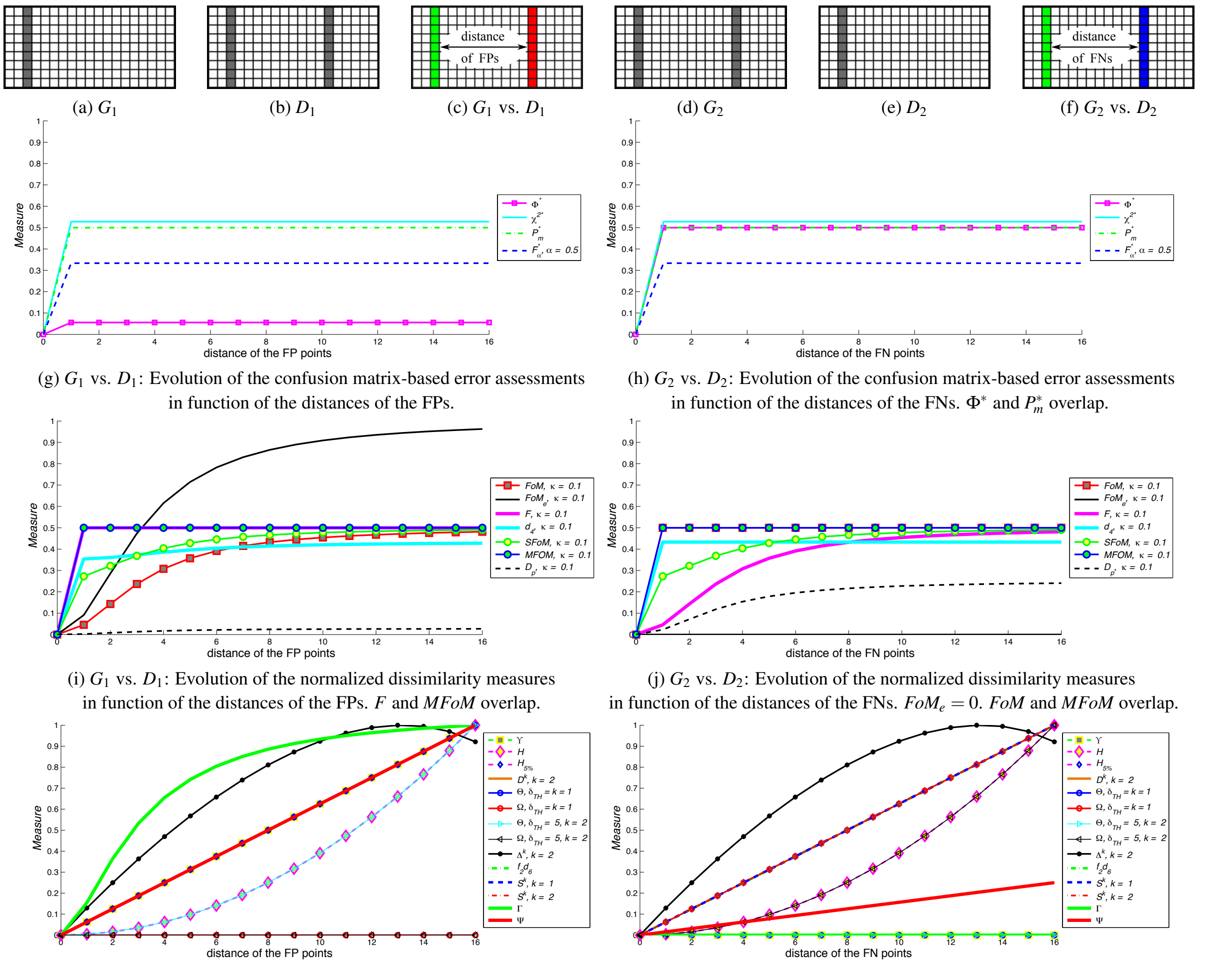
Symmetric Figure of Merit [14]
$SFoM(G_t,D_c) = \frac{1}{2} \cdot FoM(G_t,D_c) + \frac{1}{2} \cdot FoM(D_c,G_t)$

Maximum Figure of Merit [14]
$MFoM(G_t,D_c) = \max(FoM(G_t,D_c), FoM(D_c,G_t))$

Edge map quality measure [15]
$D_p(G_t,D_c) = \frac{1/2}{\lvert I\rvert - \lvert G_t\rvert} \cdot L \ + \ \frac{1/2}{\lvert G_t\rvert} \cdot R$ $L = \sum_{p \in D_c} 1 - \frac{1}{1+\kappa \cdot d^2_{G_t}(p)}$ and $R = \sum_{p \in G_t} 1 - \frac{1}{1+\kappa \cdot d^2_{G_t \cap D_c}(p)}$

A reference-based edge map quality measure requires that a displaced edge should be penalized in function not only of FPs and/or FNs but also of the distance from the position where it should be located. Tables 2 and 3 review the most relevant measures in the literature. The common feature between these evaluators corresponds to the error distance $d_{G_t}(p)$ or/and $d_{D_c}(p)$. Indeed, for a pixel belonging to the desired contour $p \in D_c$, $d_{G_t}(p)$ represents the minimal euclidian distance between p and G_t. On the contrary, if a pixel p belongs to the ground truth G_t, $d_{D_c}(p)$ is the minimal euclidian distance between p and D_c. On the one hand, some distance measures are specified in the evaluation of over-segmentation (i.e. presence of FPs), like: FoM_e, Υ, D^k, Θ and Γ (see also [26]). On the other hand, Ω measure assesses an edge detection by computing only an under segmentation (i.e. missing ground truth points, see also [26]). Other edge detection evaluation measures consider both FPs and FNs.

First, to achieve a quantitative index of edge detector performance, one of the most popular descriptors is the Figure of Merit (FoM). This distance measure ranges from 0 to 1, where 0 corresponds to a perfect segmentation [10]. Widely utilized for comparing several different segmentation methods, in particular thanks to its normalization criterion, this assessment approach nonetheless suffers from a main drawback. Whenever FNs are created, the distance of FNs ($d_{D_c}(p)$) are not recorded. Indeed, FoM can be rewritten as:



(a) G_1 (b) D_1 (c) G_1 vs. D_1 (d) G_2 (e) D_2 (f) G_2 vs. D_2

(g) G_1 vs. D_1: Evolution of the confusion matrix-based error assessments in function of the distances of the FPs.

(h) G_2 vs. D_2: Evolution of the confusion matrix-based error assessments in function of the distances of the FNs. Φ^* and P^*_m overlap.

(i) G_1 vs. D_1: Evolution of the normalized dissimilarity measures in function of the distances of the FPs. F and $MFoM$ overlap.

(j) G_2 vs. D_2: Evolution of the normalized dissimilarity measures in function of the distances of the FNs. $FoM_e = 0$. FoM and $MFoM$ overlap.

(k) G_1 vs. D_1: Evolution of the non-normalized dissimilarity measures in function of the distances of the FPs. $\Omega = 0$. H and $\Theta_{\delta_{TH}} = 5$ overlap. Υ, $H_{5\%}$, D^k, $\Theta_{\delta_{TH}} = 1$, $f_2 d_6$, S^k and Ψ overlap.

(l) G_2 vs. D_2: Evolution of the non-normalized dissimilarity measures in function of the distances of the FNs. $\Upsilon = \Theta = D^k = \Gamma = 0$. H and $\Omega_{\delta_{TH}} = 5$ overlap. $H_{5\%}$, $\Omega_{\delta_{TH}} = 1$, $f_2 d_6$ and S^k overlap.

Figure 2: Evolution of dissimilarity measures in function of the the distance of the false positive/negative points. A vertical line of false positive points (b) or false negative points (d) is shifted by a maximum distance of 16 pixels and the measure scores are plotted in function of the displacement of the desired/undesired contour.

$$\begin{aligned} FoM(G_t,D_c) &= 1 - \frac{\sum\limits_{p\in D_c\cap G_t} \frac{1}{1+\kappa\cdot d^2_{G_t}(p)} + \sum\limits_{p\in D_c\cap\neg G_t}\frac{1}{1+\kappa\cdot d^2_{G_t}(p)}}{\max(|G_t|,|D_c|)} \\ &= 1 - \frac{TP + \sum\limits_{p\in D_c\cap\neg G_t}\frac{1}{1+\kappa\cdot d^2_{G_t}(p)}}{\max(|G_t|,|D_c|)}, \end{aligned} \qquad (1)$$

because, for $p \in D_c \cap G_t$, $d^2_{G_t}(p) = 0$ and $\frac{1}{1+\kappa\cdot d^2_{G_t}(p)} = 1$. Knowing that $TP = |G_t| - FN$, for the extreme cases, the FoM measures takes the following values:

$$\begin{cases} \text{if } FP = 0: FoM(G_t,D_c) = 1 - \frac{TP}{|G_t|}, \\ \text{if } FN = 0: FoM(G_t,D_c) = 1 - \frac{1}{\max(|G_t|,|D_c|)} \cdot \sum\limits_{p\in D_c\cap\neg G_t} \frac{1}{1+\kappa\cdot d^2_{G_t}(p)}. \end{cases} \qquad (2)$$

When $FP = 0$, FoM behaves like matrix-based error assessments. Moreover, for $FP > 0$, as $\frac{1}{1+\kappa\cdot d^2_{G_t}(p)} < 1$, the FoM measure penalizes the over-detection very low compared to the under-detection. The curve in Fig. 2 shows that the penalization of missing points (FNs) becomes higher whereas it is weaker concerning FP. On the contrary, the F measure computes the distances of FNs:

$$F(G_t,D_c) = 1 - \frac{TP + \sum\limits_{p\in\neg D_c\cap G_t}\frac{1}{1+\kappa\cdot d^2_{D_c}(p)}}{|G_t \cup D_c|}. \qquad (3)$$

F behaves inversely to FoM:

$$\begin{cases} \text{if } FP = 0: F(G_t,D_c) = 1 - \frac{|D_c| + \sum\limits_{p\in\neg D_c\cap G_t}\frac{1}{1+\kappa\cdot d^2_{D_c}(p)}}{|G_t|}, \\ \text{if } FN = 0: F(G_t,D_c) = 1 - \frac{|G_t|}{|D_c|}. \end{cases} \qquad (4)$$

Also, d_4 measure depends particularly on TP, FP, FN and FoM. Nonetheless, this measure penalizes FNs like the FoM measure, as shown in Fig. 2 (j). $SFoM$ and $MFoM$ take into account both distances of FNS and FPs, so they can compute a global evaluation of a contour image, but as illustrated in Figs. 2 (i) and (j), $MFoM$ does not considers FPs and FNs a the same time, contrary to $SFoM$. Another way to compute a global measure is presented in [15] with the edge map

Table 3: List of non-normalized error measures. In the literature, the most common values are $k = 1$ or $k = 2$.

Yasnoff measure [16]
$\Upsilon(G_t,D_c) = \frac{100}{\lvert I\rvert} \cdot \sqrt{\sum_{p \in D_c} d^2_{G_t}(p)}$
Hausdorff distance [17]
$H(G_t,D_c) = \max\left(\max_{p \in D_c} d_{G_t}(p), \max_{p \in G_t} d_{D_c}(p)\right)$
Distance to G_t [18] [17][19][20]
$D^k(G_t,D_c) = \frac{1}{\lvert D_c\rvert} \cdot \sqrt[k]{\sum_{p \in D_c} d^k_{G_t}(p)}$, $k \in \mathbb{R}^+, \quad k = 1$ for [18]
Maximum distance [19]
$f_2 d_6(G_t,D_c) = \max\left(\frac{1}{\lvert D_c\rvert} \cdot \sum_{p \in D_c} d_{G_t}(p), \frac{1}{\lvert G_t\rvert} \cdot \sum_{p \in G_t} d_{D_c}(p)\right)$
Oversegmentation [21][22]
$\Theta(G_t,D_c) = \frac{1}{FP} \cdot \sum_{p \in D_c} \left(\frac{d_{G_t}(p)}{\delta_{TH}}\right)^k$, $k \in \mathbb{R}^+$ and $\delta_{TH} \in \mathbb{R}^+_*$ [22], $k = \delta_{TH} = 1$ for [21]
Undersegmentation [21][22]
$\Omega(G_t,D_c) = \frac{1}{FN} \cdot \sum_{p \in G_t} \left(\frac{d_{D_c}(p)}{\delta_{TH}}\right)^k$, $k \in \mathbb{R}^+$ and $\delta_{TH} \in \mathbb{R}^+_*$ [22], $k = \delta_{TH} = 1$ for [21]
Baddeley's Delta Metric [23]
$\Delta^k(G_t,D_c) = \sqrt[k]{\frac{1}{\lvert I\rvert} \cdot \sum_{p \in I} \lvert w(d_{G_t}(p)) - w(d_{D_c}(p))\rvert^k}$, $k \in \mathbb{R}^+$ and a convex function $w : \mathbb{R} \mapsto \mathbb{R}$
Symmetric distance [19][20]
$S^k(G_t,D_c) = \sqrt[k]{\frac{\sum_{p \in D_c} d^k_{G_t}(p)) + \sum_{p \in G_t} d^k_{D_c}(p)}{\lvert D_c \cup G_t\rvert}}$, $k \in \mathbb{R}^+, \quad k = 1$ for [19]
Magnier *et al.* measure [24]
$\Gamma(G_t,D_c) = \frac{FP+FN}{\lvert G_t\rvert^2} \cdot \sqrt{\sum_{p \in D_c} d^2_{G_t}(p)}$
Symmetric distance measure [25] [14]
$\Psi(G_t,D_c) = \frac{FP+FN}{\lvert G_t\rvert^2} \cdot \sqrt{\sum_{p \in G_t} d^2_{D_c}(p) + \sum_{p \in D_c} d^2_{G_t}(p)}$

quality measure D_p. The over-segmentation measure (left term) evaluates d_{D_c}, the distances between the FPs and G_t. The under-segmentation measure (right term) computes the distances of the FNs between the closest correctly detected edge pixel, i.e. $G_t \cap D_c$. That means that FNs and their distances are not counted without the presence of TP(s), and D_p is more sensitive to FNs than FPs, see Figs. 2 (i) and (j).

A second measure widely computed in matching techniques is represented by the Hausdorff distance H, which measures the mismatch of two sets of points [17]. This max-min distance could be strongly deviated by only one pixel which can be positioned sufficiently far from the pattern. To improve the measure, one idea is to compute H with a proportion of the maximum distances (for example 5% of the values [17]); let us note $H_{5\%}$ this measure. Nevertheless, as pointed out in [19], an average distance from the edge pixels in the candidate image to those in the ground truth is more appropriate for matching purposes than H and $H_{n\%}$. To achieve this task, D^k, Υ, Θ and Γ which represent errors of distance only in function of d_{G_t}, they correspond to a measure of over-segmentation (only FPs), as indicated by the curves in Figs. 2 (l) where the curves stagnate at 0. On the contrary, the sole use of a distance d_{D_c} instead of d_{G_t} enables an estimation of the FN divergences, representing an under-segmentation (as in Ω). Nevertheless, as concluded in [27], a complete and optimum edge detection evaluation measure should combine assessments of both over- and under-segmentation, as $f_2 d_6$, S^k and Ψ. Also, combining both d_{D_c} and d_{G_t}, Baddeley's Delta Metric (Δ^k) [23] is a measure derived from the Hausdorff distance which is intended to estimate the dissimilarity between each element of two binary images. Finally, curves in Figs. 2 (k) and (l) illustrate that H, $H_{5\%}$, Δ^k, $f_2 d_6$ and S^k behave similarly in function of the FPs or FNs distances. Note that the Ψ measure is more sensitive to the distance of the FPs. The scores of the non-normalized measures in Figs. 2 (k) and (l) are normalized using the following equation for easy visual comparison. Denoting by $f \in [0;+\infty[$ the score vectors of a distance measure such that:

$$\begin{cases} m & = \min\left(\min(f(G_1,D_1)), \min(f(G_2,D_2))\right), \\ M & = \max(\max(f(G_1,D_1)), \max(f(G_2,D_2))); \end{cases}$$

then the normalization $\mathcal{N}$ of a measure is computed by:

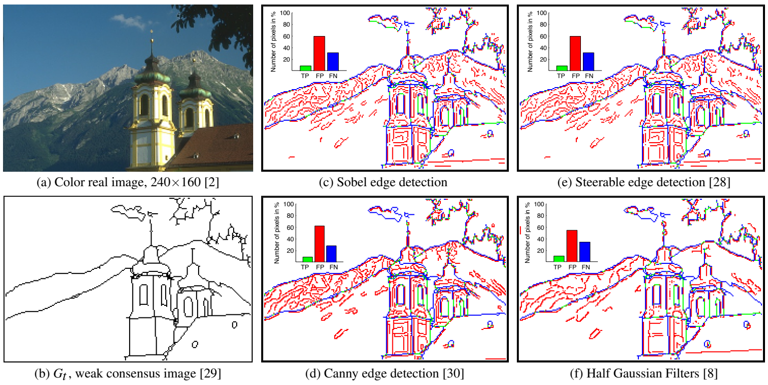
$$\mathcal{N}(f) = \begin{cases} 0 & \text{if } M = m = 0 \\ 1 & \text{if } M = m \neq 0 \\ \dfrac{f-m}{M-m} & \text{if } M > 1 \text{ and } m \neq 0 \\ f & \text{otherwise.} \end{cases} \tag{5}$$

Other details and behaviors of the different measures are available in [25] and [14]. In the rest of this communication and the supplementary material[1] , the values indicated in the Tables or curves correspond to the true scores of each measure.

3 HOW TO CREATE PRECISE GROUND TRUTH IMAGES? HOW TO EVALUATE A FILTERING TECHNIQUE?

An edge detector is considered as robust when the evaluation score of the dissimilarity with a given G_t is close to 0. Table in Fig.3 reports different assessments for four edge detection methods on a real image (color): Sobel, Canny [30], Steerable Filters (S-F) [28] and Half Gaussian Kernels (H-K) [8]. Only the comparison of D_c with a G_t is studied here. Segmentations are classified together by comparing the scores of the dissimi-

[1] The supplementary material is available at `http://media.wix.com/ugd/c95124_b9338752ae3a4e47852e0fa7bcc58b28.pdf`.



(a) Color real image, 240×160 [2]

(c) Sobel edge detection

(e) Steerable edge detection [28]

(b) G_t, weak consensus image [29]

(d) Canny edge detection [30]

(f) Half Gaussian Filters [8]

Meas.	Sobel	Canny	S-F	H-K
P_m^*	0.908	0.908	0.988	0.893
Φ^*	0.766	0.777	0.799	0.779
χ^{2*}	0.987	0.986	0.988	0.979
F_α^*	0.831	0.831	0.838	0.807
FoM	0.399	0.381	0.408	0.344
F	0.705	0.679	0.649	0.6148
Υ	1.007	1.041	1.058	1.025
$\Theta_{\delta_{TH}=1}$	4.323	4.369	4.852	4.577
$\Omega_{\delta_{TH}=1}$	2.783	1.721	1.410	1.640
$H_{5\%}$	19.76	18.45	19.56	20.89
Δ^k	9.454	7.019	8.517	11.13
$S_{k=2}^k$	5.799	5.421	5.626	5.776
Γ	0.499	0.486	0.465	0.393
Ψ	0.584	0.499	0.471	0.402

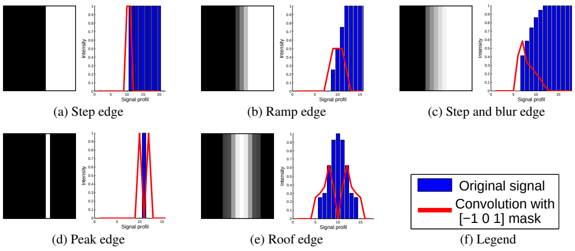
Figure 3: Edge detection after the non-maximum suppression [31] and comparison with a ground truth image.

larity measures and the smallest score for a given measure indicates the best segmentation. Indeed, for example, Sobel corresponds to the best edge detector for Υ, Canny for Δ^k, S-F for Ω and H-F for FoM. However, this assessment suffers from two main drawbacks. Firstly, segmentations are compared using the threshold (voluntary) chosen by the user, this evaluation is very subjective and not reproducible [14]. Secondly, some deficiencies appear in real ground truth contour maps, which could disturb the evaluation of efficient segmentation methods, or, on the contrary, advantage weak/biased edge detectors. Thus, according to the used measure or threshold any detector is classified the first one or the last one.

3.1 Ground truth images

In edge detection assessment, the ground truth is considered as a perfect segmentation. The most common method for ground-truth definition in natural images remains manual labeling by humans [2] [32]. These data sets are not optimal in the context of the definition of low-level segmentation. Firstly, labelers have marked mainly edges of salient objects, whereas equally strong edges in the background or around less important objects are missing. Moreover, errors may be created by human labels (oversights or supplements); indeed, an inaccurate ground truth contour map in terms of localization penalizes precise edge detectors and/or advantages the rough algorithms. There is another problem with the perception. In human perception, images can be ambiguous [33], image structures tend to retain their initial reference (desired shapes) frames, even when rotated or with scale variation(s). In manual segmentation, the perception is influenced by the effects of the particular expectations, the labeler tends to mark easier the contours of a desired object which should be labeled and it influences the result. Finally, in [34], the question is raised concerning the reliability of the datasets regarded as ground truths for novel edge detection methods. Thus, an incomplete ground truth penalizes an algorithm detecting true boundaries and efficient edge detection algorithms obtain between 30% and 40% of errors. Furthermore, when G_t maps are built from a consensus which consists in the combination of several human-labelled images [35][27][29], the deficiencies recalled above remain present. These reasons accentuate the importance of the relevant development of the ground truth labeling.

In real digital images, various profile edge types determine contours such as: step, ramp, roof of peaks. Pure step edges are seldom present in real image scenes, but they can be created in synthetic data. As illustrated in Fig. 4, edge positions correspond to the points of the higher gradient magnitude. For a 2D signal, pixels of contours are measured having the higher slope and are localized in the perpendicular direction of the slope of the image function. Considering synthetic data, true edges are positioned between two different colors/gray levels. Nevertheless, the edge position of an object



(a) Step edge

(b) Ramp edge

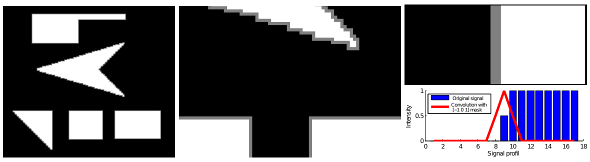
(c) Step and blur edge

(d) Peak edge

(e) Roof edge

(f) Legend

Figure 4: The different types of edges and result of a convolution with a $[\,-1\;0\;1\,]$ mask (absolute value).

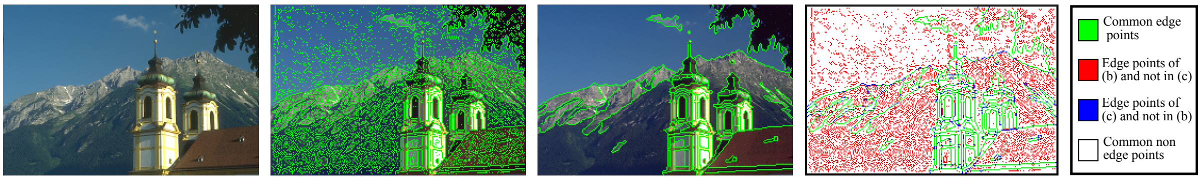


(a) Image 270×238

(b) Zoom in (a)

(c) Position of the detected contour

Figure 5: Synthetic data with a 1 pixel width gray around each shape: value of white pixels = 1, values of black pixels = 0, values of gray pixels = 0.5.


🟩 Common edge points
🟥 Edge points of (b) and not in (c)
🟦 Edge points of (c) and not in (b)
□ Common non edge points

(a) Original image (b) Thin edges with $[-1\,0\,1]$ mask (c) Adjustment by hand (d) Image in (b) vs image in (c) (e) Legend

Figure 6: Image of our database are built after an edge detection involving a $[-1\,0\,1]$ mask and concluded by hand.

could be interpreted in different ways: for a vertical step edge, an edge can be located either on the left, or on the right. In Fig. 5, several white shapes are immersed in a black background, creating step edges. To avoid the problem of edge pixel placements, a blur must be voluntary created by adding a 1 pixel width of gray around each shape. Thus, the ground truth corresponds to the points where the slope of the image surface is maximum, i.e. to this gray. These points could be extracted involving odd filters (derivative filters of order 1). In the one hand, a $[-1\,0\,1]$ mask allows to extract the edges at the correct position, i.e. the gray pixels in Fig. 5, contrary to edge detector involving smoothing parameters which delocalize edge positions (especially corners and small objects [1]). In the other hand, using an odd filter, two edges are extracted corresponding to the two boundaries of the roof/peak (see Fig. 4 and [24]); however, human labelers, in majority, indicate only one edge. The new database of contour images issued of real images takes into account all these properties. This paper presents ground truth edge maps which are labeled in a semi-automatic way in order to evaluate the performance of filtering step/ramp edge detectors. Therefore, the motivations to create new ground truth edge images are:

1. To obtain contours accurately localized,
2. To extract edges of the secondary objects or in the background,
3. To exclude boundaries inside noisy/textured regions.

In fact, this new label processes in return to hand made ground truth. Indeed, in a first time, the contours are detected involving the convolution of the image with $[-1\,0\,1]$ vertical and horizontal masks followed by a computation of a gradient magnitude and a suppression of local non-maxima in the gradient direction [31]. Concerning color images, $[-1\,0\,1]$ vertical and horizontal masks are applied to each channel of the image followed by a structure tensor [36]. In a second time, undesirable edges are deleted while missing points are added both by hand. Fig. 6 illustrates the steps to obtain new ground truth images. Using the $[-1\,0\,1]$ mask enables to capture the majority of edge points and corners without deforming small objects, contrary to edge detectors involving Gaussian filters (see for example Fig. 6 in [37]). Moreover, this process enables to detect the good positions of the contours while avoiding the addition of too much imprecise ground truth points, as shown in Fig. 4 and Fig. 5.

3.2 Minimum of the measure

Instead of thresholding manually or automatically [38][39] and then comparing the segmentation of several edge detectors, as in Fig. 8 (c) and (d), the dissimilarity measures are used for an objective assessment. Indeed, the purpose is to compute the minimal value of a dissimilarity measure by varying the threshold Th of the thin edges computed by an edge detector (thin edges are created after the non-maximum suppression of the absolute gradient [31]). Indeed, compared to a ground truth contour map, the ideal edge map for a measure corresponds to the desired contour at which the evaluation obtains the minimum score for the considered measure among the thresholded gradient images. Theoretically, this score corresponds to the threshold at which the edge detection represents the best edge map, compared to the ground truth contour map [40][27][25]. Fig. 7 illustrates all this process. Since a small threshold leads

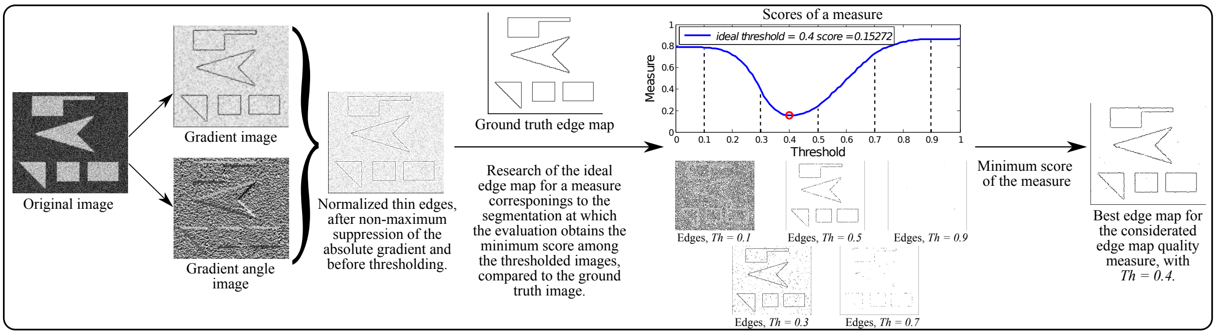
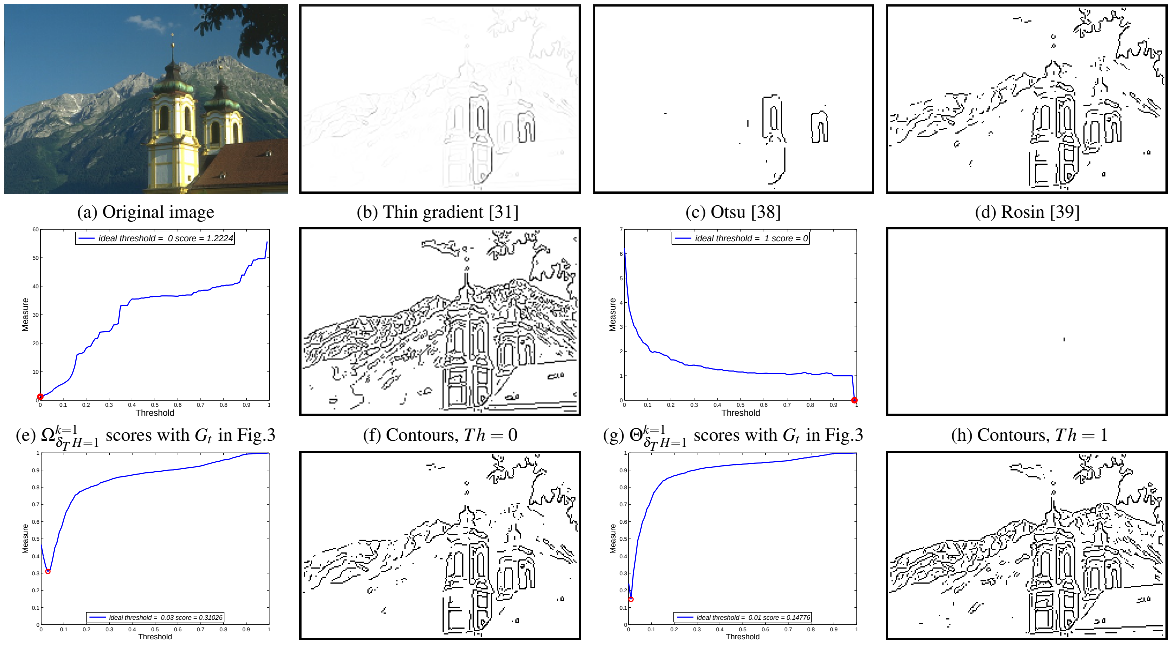

Original image
Gradient image
Gradient angle image
Normalized thin edges, after non-maximum suppression of the absolute gradient and before thresholding.
Ground truth edge map
Research of the ideal edge map for a measure corresponings to the segmentation at which the evaluation obtains the minimum score among the thresholded images, compared to the ground truth image.
Scores of a measure
ideal threshold = 0.4 score =0.15272
Measure
0
0.2
0.4
0.6
0.8
1
Threshold
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
Edges, Th = 0.1
Edges, Th = 0.5
Edges, Th = 0.9
Edges, Th = 0.3
Edges, Th = 0.7
Minimum score of the measure
Best edge map for the considered edge map quality measure, with Th = 0.4.

Figure 7: The most relevant edge map for a dissimilarity measure is indicated by its minimum score.



(a) Original image (b) Thin gradient [31] (c) Otsu [38] (d) Rosin [39]

(e) $\Omega^{k=1}_{\delta_T H=1}$ scores with G_t in Fig.3 (f) Contours, $Th = 0$ (g) $\Theta^{k=1}_{\delta_T H=1}$ scores with G_t in Fig.3 (h) Contours, $Th = 1$

(i) *FoM* scores with G_t in Fig.3 (j) Contours, $Th = 0.03$ (k) *FoM* scores with G_t in Fig. 6 (l) Contours, $Th = 0.01$

Figure 8: Scores of the measures depending on the threshold of the thin gradient image [30].

to heavy over-segmentation and a strong threshold may create numerous false negative pixels, the minimum score of an edge detection evaluation should be a compromise between under- and over-segmentation. As illustrated in Fig. 8 (e) the best score for the under-segmentation evaluation corresponds to $Th = 0$, because false negative points penalize the Ω measure. On the contrary, false positive points penalize over-segmentation dissimilarity measures, as FoM_e, Υ, D^k, Θ and Γ measures, see Fig. 8 (g). Consequently, the best score concerning an over-segmentation measure corresponds to $Th \approx 1$. As G_t are not the same for the evaluation in Fig. 8 (g) and (h), the two curves are different.

4 EXPERIMENTAL RESULTS

The purpose of the experiments presented here is to obtain the best edge map in a supervised way. In order to study the performance of the contour detection evaluation measures, each measure is compared by varying the threshold of the thin edges computed by until six edge detectors: Sobel, Canny [30], Steerable Filters of order 1 (SF_1) [28], Steerable Filters of order 5 (SF_5) [41], Anisotropic Gaussian Kernels (AGK) [42] and Half Gaussian Kernels (HK) [8]. In the one hand, experiments are led on two synthetic noisy images. In the other hand, contour detection evaluations are compared on seven real images where G_t edge maps are labelled by a semi-automatic way (section 3.1). Finally, compared to a ground truth contour map, the ideal edge map for a measure corresponds to the desired contour at which the evaluation obtains the minimum score for the considered measure among the thresholded thin gradient images [30].

Firstly, to evaluate the performances of the dissimilarity measures, the original image in Fig. 5(a) is disturbed with random Gaussian noise and edges are extracted from the noisy images (4dB and 3.3dB, see supplementary material). Generally, the scores of Φ^*, d_4 and D_p measures allow to correctly extract the edges at the price of numerous FPs. Moreover, Δ^k is more sensitive to FPs than the other dissimilarity measures and the best score corresponds to a contour edge map with many discontinuous contours. As pointed out in section 2, concerning the image corrupted by a noise at a level of 4dB, *FoM* penalizes strongly FNs to the detriment of FPs apparitions, and it considers that anisotropic edge detectors are less performant than the Canny edge detector. Other measures classify the Sobel method as the less efficient one and the H-K as the best one.

The second experiment concerns a real image presented in Fig. 6(a); G_t is available in Fig. 6(c). The best edge maps based on minimum of the scores of different measures are presented in Fig. 9. Statistical measures and d_4 consider that Sobel is the best edge detector for this image because edges are well localized. Even though edge maps are different, the scores obtained by *FoM* and F are similar for the different filtering techniques. Oriented kernels, however, are qualified as reliable by distance measures and edge maps corresponding to the minimum scores are less noisy. In the supplementary material are compared the best edge maps obtains with our G_t and G_t of Berkeley segmentation image (Fig. 3(b)). Excepted for Φ^*, d_4 and D_p measures, the best edge map for all the other measures contains many holes in the contour chains and it is clearly impossible to conclude which edge detector is the most efficient. Table 4 mentions scores involving the two different G_t: by hand made, and semi-automatic. It is important to note that the scores for each measure is smaller concerning G_t built in a semi-automatic way.

Other results presented in the supplementary material show that the minimum scores concerning the distance measures. When objects appear clear, like in image 56 and buildings, most of the measure scores indicate that the edge detectors are equivalent. By contrast, as soon as images contain blur or/and noise, as in image 109 and parkingmeter, the evaluation measures involving error distances considerate that oriented and anisotropic filters produce better-defined contours. Finally, image 109 is a noisy image, however Δ^k and D_p evaluate that Sobel detects better edge, whereas it creates many undesirable contour points, contrary to filtering techniques involving smoothing effects.

Numerous experiments show that $S^k_{k=1\,or\,k=2}$ and Ψ dissimilarity measures are best fitted in the problem of supervised edge evaluation. Indeed, the minimum evaluation scores are coherent and the edge detectors are qualified as best when the filtering technique is adapted to the image structure (blur, noise, small objects). Moreover, the edge map corresponding to the minimum score delimit correctly the object with a majority of continuous contours points without much undesirable points.

5 CONCLUSION

This study presents a review of supervised edge detection assessment methods in details. Moreover, based on the theory of these dissimilarity evaluations, a technique is proposed to evaluate filtering edge detection methods involving the minimum score of the considerate measures. Indeed, to evaluate an edge detection technique, the result which obtains the minimum score of a measure is considerate as the best one and represents an objective evaluation. Theoretically and with the backing of many experiments is demonstrated that the minimum score of the $S^k_{k=1\,or\,k=2}$ and Ψ dissimilarity measures correspond to the best edge quality map evaluations. These two measures take into account both the distances of false positive and false negative points. Many experiments of edge detection on synthetic and real images involving several edge detectors illustrate this conclusion. Experiments show the significance of the ground truth map choice: an inaccurate ground truth contour map in terms of localization penalizes precise edge detectors and/or advantages the rough algorithms. That is the reason why is described in this conversation how to build a new ground truth edge map labelled in semi-automatic way in real images. Firstly, the contours are detected involving the convolution of the image with $[-1\;0\;1]$ masks. Secondly, undesirable edges are removed while missing points are added both by hand, thus a more accuracy ground truth edge map image is built and can be used for supervised contour detection evaluation. By comparison with a real image where contours points are not precisely labelled, experiments illustrate that the new ground truth database allows to evaluate the performance of edge detectors by filtering. Finally, the advantage to compute the minimum score of a measure involving this new ground truth database is that it does not require tuning parameters. For this purpose, we plan in a future study to compare the robustness several edge detection algorithms by adding noise and blur on real images presented in the supplementary material and then using the optimum threshold computed by the minimum of the evaluation.

Table 4: Comparison of scores of dissimilarity measures using a ground truth from [2] (Fig. 3 (b)) image and a constructed ground truth by a semi-automatic way. Contour images and curves for all the measures are available in the supplementary material.

Meas.	Sobel		Canny		SF_1 [28]		AGK [42]		H-K [8]	
	Berkeley G_t	Our G_t	Berkeley G_t	Our G_t	Berkeley G_t	Our G_t	Berkeley G_t	Our G_t	Berkeley G_t	Our G_t
Φ^*	0.738	0.298	0.757	0.430	0.971	0.447	0.813	0.496	0.761	0.504
χ^{2*}	0.979	0.635	0.975	0.725	0.983	0.712	0.982	0.759	0.973	0.502
P^*_m	0.901	0.530	0.901	0.603	0.909	0.594	0.917	0.637	0.893	0.778
F^*_α	0.820	0.360	0.819	0.432	0.834	0.422	0.847	0.468	0.808	0.483
FoM	0.303	0.168	0.310	0.147	0.309	0.164	0.299	0.154	0.277	0.146
F	0.592	0.346	0.579	0.352	0.572	0.310	0.589	0.337	0.589	0.367
d_4	0.675	0.333	0.671	0.379	0.687	0.375	0.695	0.412	0.667	0.424
$SFoM$	0.297	0.145	0.289	0.134	0.270	0.111	0.271	0.119	0.268	0.128
D_P	0.173	0.036	0.184	0.058	0.193	0.056	0.208	0.065	0.183	0.072
H	40.02	29.52	19.41	15.175	18.97	18.02	35.35	14.76	36.87	15.03
$H_{5\%}$	13.72	9.406	11.89	9.142	11.53	6.781	14.18	6.048	14.56	7.165
Δ^k	6.632	4.094	5.039	3.000	4.844	2.462	6.044	2.040	6.562	2.576
f_2d_6	2.851	1.066	2.498	1.294	2.467	0.900	2.625	0.895	2.582	0.983
$S^k_{k=1}$	2.584	1.005	2.315	0.990	2.316	0.877	2.471	0.866	2.432	0.966
$S^k_{k=2}$	4.270	2.323	3.725	2.361	3.690	1.819	4.172	1.667	4.281	2.029
Ψ	0.213	0.041	0.181	0.044	0.173	0.032	0.224	0.032	0.222	0.038

6 ACKNOWLEDGEMENTS

The authors would like to thank the Iraqi Ministry of Higher Education and Scientific Research for funding and supporting this work, and, Philippe Borianne for its helpful comments and suggestions.

7 REFERENCES

[1] D. Ziou and S. Tabbone, “Edge detection techniques: an overview,” *Int. J. on Patt. Rec. and Image Anal.*, vol. 8, no. 4, pp. 537–559, 1998.

[2] D. Martin, C. Fowlkes, D. Tal, and J. Malik, “A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics,” in *IEEE ICCV*, vol. 2, pp. 416–423, 2001.

[3] P. Sneath and R. Sokal, *Numerical taxonomy. The principles and practice of numerical classification.* 1973.

[4] C. Grigorescu, N. Petkov, and M. Westenberg, “Contour detection based on nonclassical receptive field inhibition,” *IEEE TIP*, vol. 12, no. 7, pp. 729–739, 2003.

[5] S. Venkatesh and P. L. Rosin, “Dynamic threshold determination by local and global edge evaluation,” *CVGIP*, vol. 57, no. 2, pp. 146–160, 1995.

[6] Y. Yitzhaky and E. Peli, "A method for objective edge detection evaluation and detector parameter selection," *IEEE TPAMI*, vol. 25, no. 8, pp. 1027–1033, 2003.

[7] D. Martin, C. Fowlkes, and J. Malik, "Learning to detect natural image boundaries using local brightness, color, and texture cues," *IEEE TPAMI*, vol. 26, no. 5, pp. 530–549, 2004.

[8] B. Magnier, P. Montesinos, and D. Diep, "Fast anisotropic edge detection using gamma correction in color images," in *IEEE ISPA*, pp. 212–217, 2011.

[9] K. Bowyer, C. Kranenburg, and S. Dougherty, "Edge detector evaluation using empirical roc curves," in *CVIU*, pp. 77–103, 2001.

[10] I. E. Abdou and W. K. Pratt, "Quantitative design and evaluation of enhancement/thresholding edge detectors," *Proc. of the IEEE*, vol. 67, pp. 753–763, 1979.

[11] C. Strasters and J. Gerbrands, "Three-dimensional image segmentation using a split, merge and group approach," *Patt. Rec. Let.*, vol. 12, no. 5, pp. 307–325, 1991.

[12] A. J. Pinho and L. B. Almeida, "Edge detection filters based on artificial neural networks," in *ICIAP*, pp. 159–164, Springer, 1995.

[13] A. G. Boaventura and A. Gonzaga, "Method to evaluate the performance of edge detector," 2009.

[14] H. Abdulrahman, B. Magnier, and P. Montesinos, "A new normalized supervised edge detection evaluation," in *IbPRIA - to appear-*, 2017.

[15] K. Panetta, C. Gao, S. Agaian, and S. Nercessian, "A new reference-based edge map quality measure," *IEEE Trans. on Systems Man and Cybernetics: Systems*, vol. 46, pp. 1505–1517, 2016.

[16] W. Yasnoff, W. Galbraith, and J. Bacus, "Error measures for objective assessment of scene segmentation algorithms.," *Analytical and Quantitative Cytology*, vol. 1, no. 2, pp. 107–121, 1978.

[17] D. Huttenlocher and W. Rucklidge, "A multi-resolution technique for comparing images using the hausdorff distance," in *IEEE CVPR*, pp. 705–706, 1993.

[18] T. Peli and D. Malah, "A study of edge detection algorithms," *CGIP*, vol. 20, no. 1, pp. 1–21, 1982.

[19] M.-P. Dubuisson and A. Jain, "A modified hausdorff distance for object matching," in *IEEE ICPR*, vol. 1, pp. 566–568, 1994.

[20] C. Lopez-Molina, B. De Baets, and H. Bustince, "Quantitative error measures for edge detection," *Patt. Rec.*, vol. 46, no. 4, pp. 1125–1139, 2013.

[21] R. Haralick, "Digital step edges from zero crossing of second directional derivatives," *IEEE TPAMI*, vol. 6, no. 1, pp. 58–68, 1984.

[22] C. Odet, B. Belaroussi, and H. Benoit-Cattin, "Scalable discrepancy measures for segmentation evaluation," in *IEEE ICIP*, vol. 1, pp. 785–788, 2002.

[23] A. J. Baddeley, "An error metric for binary images," *Robust Computer Vision: Quality of Vision Algorithms*, pp. 59–78, 1992.

[24] B. Magnier, A. Le, and A. Zogo, "A quantitative error measure for the evaluation of roof edge detectors," in *IEEE IST*, pp. 429–434, 2016.

[25] B. Magnier, "Edge detection: a review of dissimilarity evaluations and a proposed normalized measure," *Multimedia Systems for Critical Engineering*, 2017.

[26] A.-B. Goumeidane, M. Khamadja, B. Belaroussi, H. Benoit-Cattin, and C. Odet, "New discrepancy measures for segmentation evaluation," in *IEEE ICIP*, vol. 2, pp. 411–414, 2003.

[27] S. Chabrier, H. Laurent, C. Rosenberger, and B. Emile, "Comparative study of contour detection evaluation criteria based on dissimilarity measures," in *EURASIP J. on Image and Video Proc.*, pp. 1–10, 2008.

[28] W. T. Freeman and E. H. Adelson, "The design and use of steerable filters," *IEEE TPAMI*, vol. 13, pp. 891–906, 1991.

[29] C. Lopez-Molina, B. De Baets, and H. Bustince, "Twofold consensus for boundary detection ground truth," *Knowledge-Based Syst.*, vol. 98, pp. 162–171, 2016.

[30] J. Canny, "A computational approach to edge detection," *IEEE TPAMI*, no. 6, pp. 679–698, 1986.

[31] A. Rosenfeld and M. Thurston, "Edge and curve detection for visual scene analysis," *IEEE Trans. on Computers*, vol. 100, no. 5, pp. 562–569, 1971.

[32] M. Heath, S. Sarkar, T. Sanocki, and K. Bowyer, "A robust visual method for assessing the relative performance of edge-detection algorithms," *IEEE TPAMI*, vol. 19, no. 12, pp. 1338–1359, 1997.

[33] M. Peterson, J. Kihlstrom, P. Rose, and M. Glisky, "Mental images can be ambiguous: Reconstruals and reference-frame reversals," *Memory & Cognition*, vol. 20, no. 2, pp. 107–123, 1992.

[34] X. Hou, A. Yuille, and C. Koch, "Boundary detection benchmarking: Beyond f-measures," in *IEEE CVPR*, pp. 2123–2130, 2013.

[35] N. Fernández-García, A. Carmona-Poyato, R. Medina-Carnicer, and F. Madrid-Cuevas, "Automatic generation of consensus ground truth for the comparison of edge detection techniques," *IVC*, vol. 26, no. 4, pp. 496–511, 2008.

[36] S. Di Zenzo, "A note on the gradient of a multi-image," *CVGIP*, vol. 33, no. 1, pp. 116–125, 1986.

[37] P. Perona and J. Malik, "Detecting and localizing edges composed of steps, peaks and roofs," in *ICCV*, pp. 52–57, IEEE, 1990.

[38] N. Otsu, "A threshold selection method from gray-level histograms," *Automatica*, vol. 11, no. 285-296, pp. 23–27, 1975.

[39] P. L. Rosin, "Unimodal thresholding," *Patt. Rec.*, vol. 34, no. 11, pp. 2083–2096, 2001.

[40] N. Fernández-Garcıa, R. Medina-Carnicer, A. Carmona-Poyato, F. Madrid-Cuevas, and M. Prieto-Villegas, "Characterization of empirical discrepancy evaluation measures," *Patt. Rec. Lett.*, vol. 25, no. 1, pp. 35–47, 2004.

[41] M. Jacob and M. Unser, "Design of steerable filters for feature detection using canny-like criteria," *IEEE TPAMI*, vol. 26, no. 8, pp. 1007–1019, 2004.

[42] J. Geusebroek, A. Smeulders, and J. van de Weijer, "Fast anisotropic gauss filtering," *ECCV*, pp. 99–112, 2002.

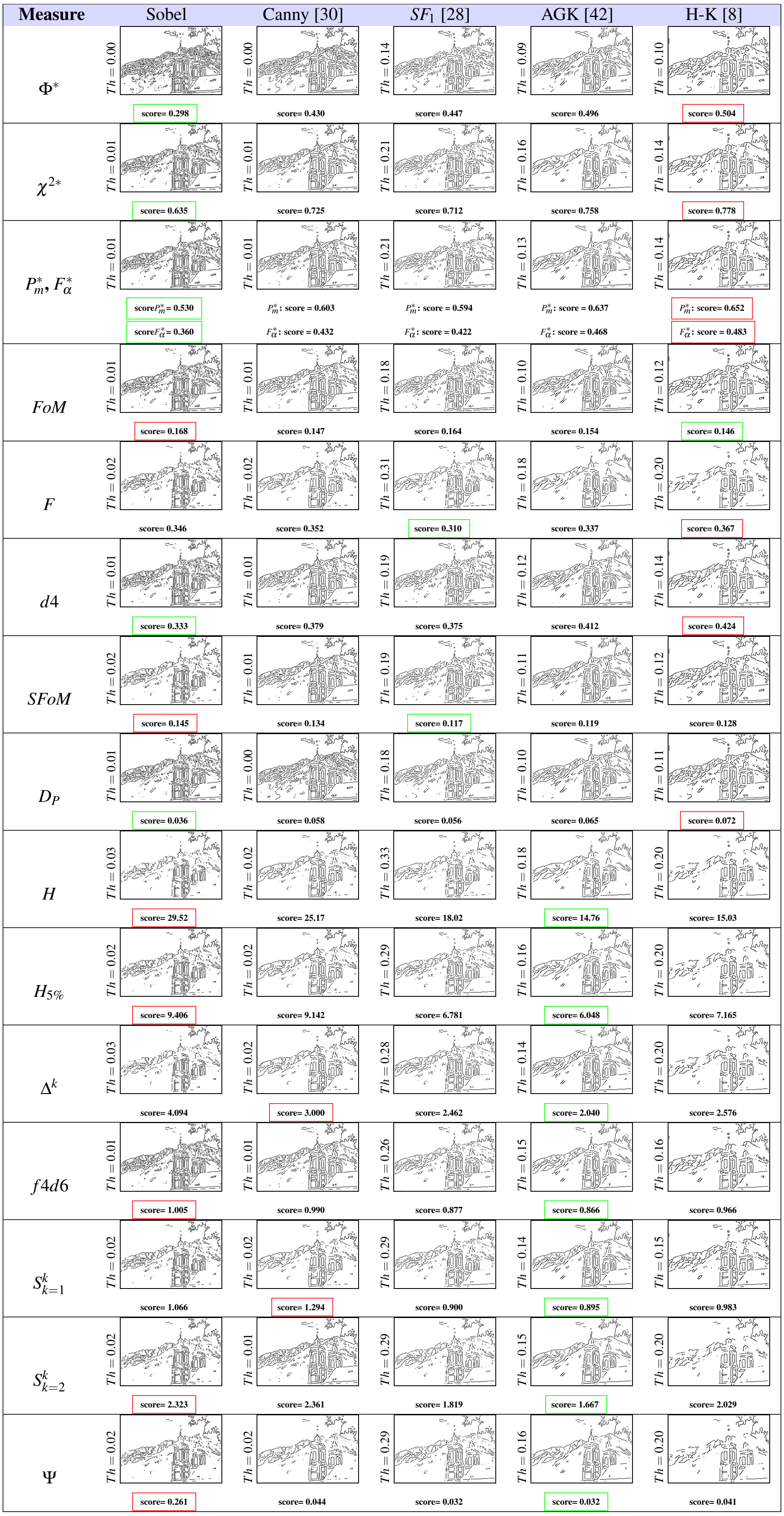


Figure 9: Best edge maps for each dissimilarity measure concerning a real image and G_t in Fig. 6.

Automatic detection of neurons, astrocytes, and layers for NISSL-stained mouse cortex

Svetlana Nosova
1st author's affiliation
1st line of address
2nd line of address
Country (ZIP) code, City, State
nosova.sv.a@gmail.com

Ludmila Snopova
2nd author's affiliation
1st line of address
2nd line of address
Country (ZIP) code, City, State
lsnopova@nizhgma.ru

Vadim Turlapov
3rd author's affiliation
1st line of address
2nd line of address
Country (ZIP) code, City, State
vadim.turlapov@gmail.com

ABSTRACT

We present an image processing algorithm for automatic detection of cortex layers and cells from optical microscopy images for Nissl-stained mouse brain. For every layer of cortex we automatically detect a shape and localization of following cortex cells type: neurons of molecular layer, pyramidal neurons, stellate neurons, and astrocytes. The algorithm includes the steps of: preprocessing, neurons and astrocytes localization, neurons classification, refined cortex layer detection, neurons reclassification. For preprocessing we use converting to gray image, Gaussian blurring, converting to black-white image after background removing, and rough estimate of layers. We use morphological operations with variation radius of structure element for neurons localization and neurons classification.

Keywords

image processing; automatic detection; optical microscopy; NISSL-staining; mouse cortex; cortex layers; neuron; astrocyte; preprocessing; morphology features

1. INTRODUCTION AND RECENT SOLUTIONS

The problem of fully automatic segmentation, decomposition and analysis of microscopy data of mouse brain (and different part of it) is known in global community. A reconstruction of microscopy data is a first step for understanding, analysis, and simulation of brain. In the optical microscopy it is needed to create special algorithms and software for each type of staining, which would help human to analyze and to segment data, and, in particular, to give information for diagnosis. In this paper we focused on NISSL-stained mouse cortex because Nissl staining is the easiest and mass for optical microscopy mouse brain.

In [Kol14] researches described three types of segmented cells: neurons, glial cells, epithelial cells. Authors review necessary features for every type of the cells. They focused on segmentation of pyramidal neurons based on special preprocessing and watershed algorithm. But the paper has no information about statistics of cortex layer and about quality of neurons and astrocytes in every layer.

In the community publications we can find a sufficient amount of sources about segmentation of neurons in NISSL-stained data. In [Das15] a segmentation based on machine learning (random forest) is used. [Hey15] describes algorithm based on normalized graph cut. Special method ANRA proposed in [Ing08]. It’s also based on machine learning techniques and has good accuracy.

What about layer decomposition, we should notice that there are different 3D mouse brain atlases with information about brain areas and layers ([All04] and [Bra05]). And in [Sen11] we can see atlas-based segmentation of NISSL-stained mouse brain data. In [Mes15] we can find segmentation of hippocampus, and in [Bas16] you can see MRI-based segmentation of cortical area.

We analyzed NISSL-stained mouse brain data. The data for our research were provided by Nizhny Novgorod State Medical Academy. In this work we consider the follow questions: features and structure of different types of neurons and astrocytes; neuron localization detection; detection of layer localization and collection of statistics for neurons in every layer.

2. DATA DESCRIPTION

Input data is section set of mouse cortex images (see Fig.1). The input data is images set of the mouse cortex slice. There are intersections, i.e., lower portion of the first image may include the same data as the upper part of the second image. It’s problem for statistics collection and data processing.

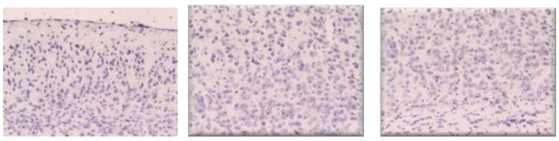


Figure 1: Section set of NISSL-stained mouse cortex.

We need algorithm and software for forming data in one image without duplicate parts. We implemented an automatic algorithm based on the algorithm for gluing panoramas as part of this work. But the algorithm was unstable, and because it was not the original purpose of this work, we decided to glue images by hand (see Fig. 2). This picture was marked by a professional pathologist.

Cortex layers, neurons and astrocytes feautures

Following cells are present on the images: molecular layer neurons (MLN), pyramidal neurons(PN), stellate neurons (SN), astrocytes. Also we have following layers in mouse cortex structure: layer I, layer II-III, layer IV, layer V, layer VI. In the layer I (the molecular layer) molecular layer neurons and astrocytes are presented. In the layer II stellate, pyramidal neurons and astrocytes are presented. In the layer IV only stellate neurons and astrocytes are presented. In the layer V all cells type are presented.and In the layer VI we can see only stellate neurons and astrocytes.

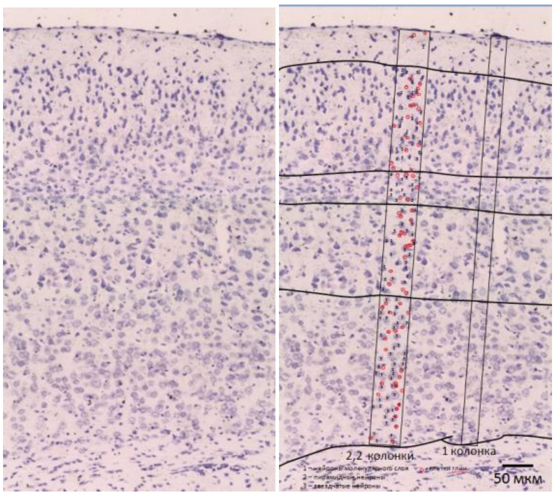


Figure 2: Thin slice (7 micron) of the mouse brain cortex in the original form, and marked up by a qualified morphologist.

We did hand segmentation to get astrocytes color statistics. For every type of neurons we did watershed segmentation from seed points and collect information about shape and color for every neuron type. You can see summary color statistics information in Fig.3.

As you see, some part of neurons can be classified using only histogram information (as have different mean and dispersion for the intensity distribution). Shape characterization of every neuron type you can see in Tab.1. This information used for neuron classification.

We noticed also that pixel has color vector value less than (200_r,80_g,200_b) for astrocytes.

	MLN	PN	SN
Area	448	965	1205
Perimeter	100	145	175
Extent	0,669	0,660	0,664
Minor Axis	19	30	33
Major Axis	34	47	50
Orientation	94	87	88
Aspect Ratio	0,595	0,645	0,669
Circularity	0,604	0,565	0,478
Solidity	0,862	0,858	0,813
Convexity	0,909	0,887	0,819
Equivalent Diameter	23	34	37

Table 1. Shape features for different types of neurons.

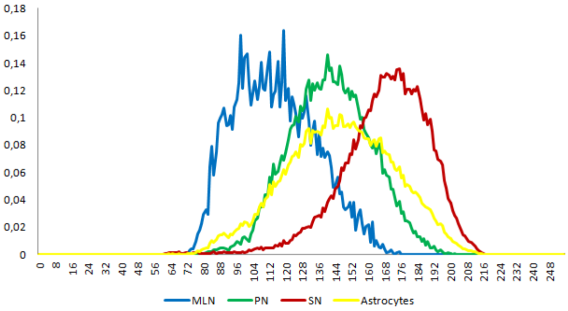


Figure 3: Normalized intensity histograms for different cells type.

3. PROPOSED SOLUTION

We propose the algorithm for automatically layer detection, neurons type classification and astrocytes detection. Main steps of the algorithm are presented in Fig.4. Auto detecting of the layers and detailed statistics collecting are main goals for our work. You’ll find detailed description for every algorithm step.

Preprocessing

Preprocessing is preparatory step for detection. It include two phases: noise removing and background removing. We use 5x5 Gauss filter for noise removing from source image. All next steps of preprocessing are used for background removing. We form image in which colored pixels comply cells and black pixels are background. Detailed description of background removing with our comments you can find in Alg.1.

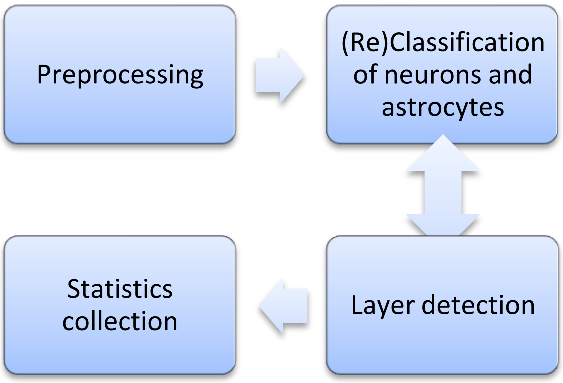


Figure 4: Algorithm for layer detection and cells classification.

```
If(source(i,j) >= min_background_value &&
source(i,j) <= max_background_value)
        bw(i,g) = white;
Else
        bw(i,j) = black;
```

Figure 5: First binarization pseudocode.

Step 1. Create a binary image *bw(i,j)* using algorithm in Fig 5. Parameters *min_background_value* and *max_background_value* are setted by human.

Step 2. Delete noisy pixels from image and create fill "black regions" on our image. Apply morphological erosion and get *erode_bw(i,j)* image.

Step 3. Invert *erode_bw(i,j)* and get *inv_bw(i,j)* .

Step 4. Find all contours on *inv_bw(i,j)* and delete all contour which less than *min_contour_area*. You need fill this contours using black color on *inv_bw(i,j)*. On this stage we can see some too big white regions, in the center of which placed background pixels.

Step 5. Apply morphological erosion for *inv_bw(i,j)* with quite big radius of morphological element and write the result in *big_erode_bw(i,j)*. And final binary image is calculated as *final_bw(i,j)=inv_bw(i,j) XOR big_erode_bw(i,j).*

Step 6. Form final image for segmentation. We create new image: for white pixels in *final_bw(i,j)* we write source color; for other pixels we set up black color.

Algorithm 1. Preprocessing

The result of preprocessing step you can see in Fig.7

Astrocytes detection

Firstly, we apply binarization using simple thresholding. The value of threshold is $(200_r,80_g,200_b)$. After that we find all circle contours on binary images and calculate mass centers for every of them. This mass centers are locations of astrocytes.

First classification of neurons

Step 1. Apply morphological erosion. The size of morphological element is *r*. The shape of structural element is circle. We get *bw_for_neurons_seg(i,j)* image.

Step 2. Find contour on *bw_for_neurons_seg(i,j)*, calculate mass centers for every contours *(xc,yc)*. This is cell centers. Do following for every center:

Step 2.1. Calculate average intensity (*av_value*) for *(xc,yc)* in *r* radius.

Step 2.2. Classify center as pyramidal neuron center if *abs(av_value – pn_average)<abs(av_value – sn_average)*, where *pn_average*, *sn_average* – mean intensity value for pyramidal and stellate neurons. Otherwise, classify it as stellate neuron.

Step 2.3. Draw circle with center in *(xc,yc)* and radius *r* on *first_seg* image. Use green color for pyramidal neurons and red color for stellate neurons.

Algorithm 2. Main part for neurons start classification

On this stage we find cells centers firstly. We also get estimation of the radius of a cell. Secondly, we classify found objects on two types: pyramidal and stellate neurons. It's preclassification for layer detection only. We do steps from Alg.2 for all *r* from *max_neuron_radius* to *min_neuron_radius* with *step_for_neurons* downstep.

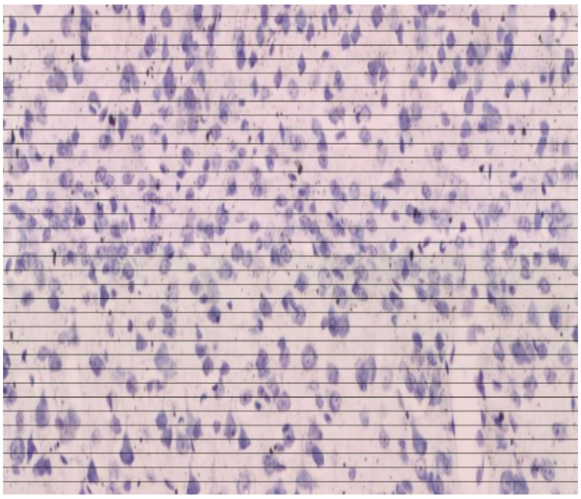


Figure 6: Bin separators (1 bin = 25 source lines) for layer detection

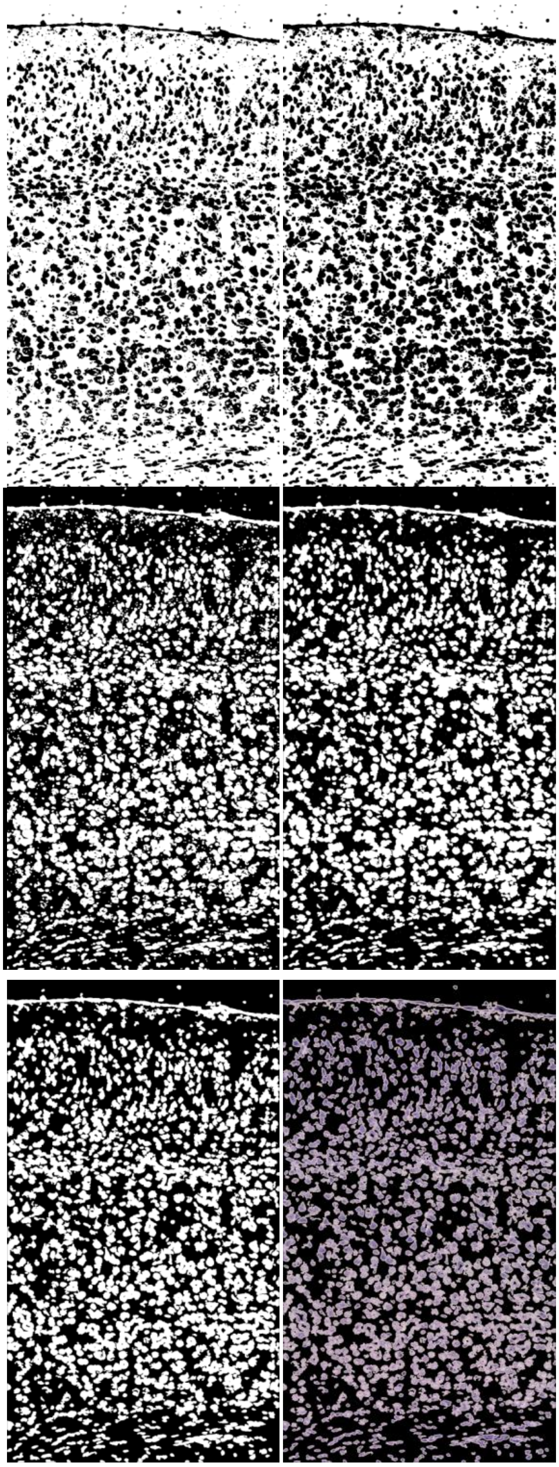


Figure 7: Steps 2-7 of preprocessing algorithm.

The result of preclassification algorithm you can on the left image of the Fig.8. In the Fig. 9 you can see results for cells detection and neurons preclassification for different data slices.

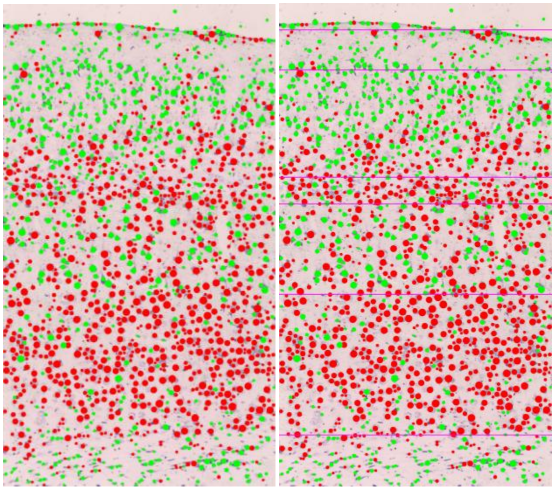


Figure 8: First neuron classification and layer detection based on it.

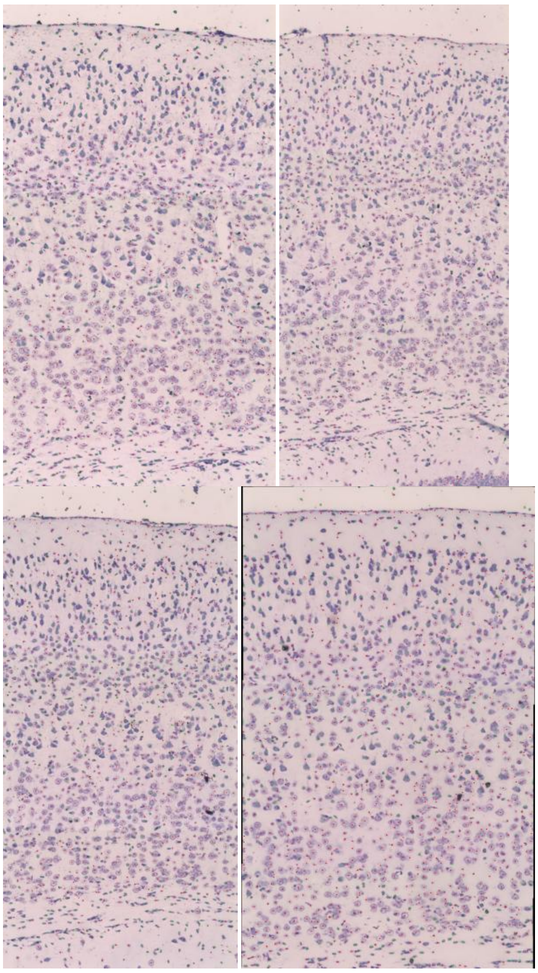


Figure 9. Cells detection and preclassification results for different data slices

Layer Detection

This stage include following two steps. The first is a calculation of cells density for quantized image by lines (Fig. 6). The algorithm for this step you can see in Alg.3. You can see density results on the Fig.9. The second is layer decomposition based on cells density distribution. To cortex segmentation by layers we need value for six separators: *l0,l1,l2,l3,l4,l5*. We use follow rules for separators settings which were described in Alg.4. You can find cortex layer decomposition on the right image of the Fig.8.

Step 1. Do quantization by source image rows. In every bin we have 25 lines, there are 145 bins summary.

Step 2. For every bin:

Step 2.1 calculate density of pyramidal neurons as:

$$PND = \frac{pixel\_PND\_count}{pixel\_count}$$

where *PND* is a pyramidal neurons density; *pixel_PND_count* is a number of green pixels in a bin of *firt_seg_image*; *pixel_count* is a number of pixels in a bin.

Step 2.2. Similarly calculate density for stellate neurons.

Step 2.3. Calculate summary cells density in a bin.

Step 2.4. Average density data using average box filter (with radius 5)

Algorithm 3. Algorithm for cells density calculation

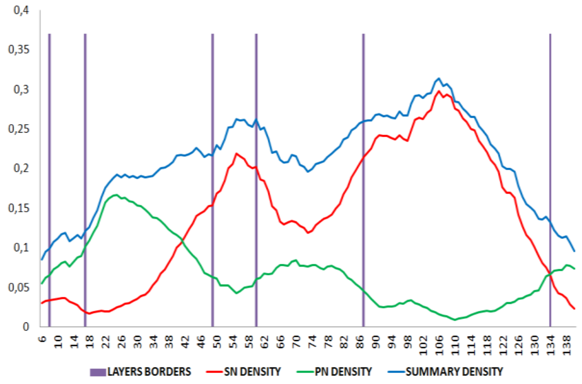


Figure 10: Density of pyramidal, stellate neurons and summary density in bins(1 bin = 25 source lines).

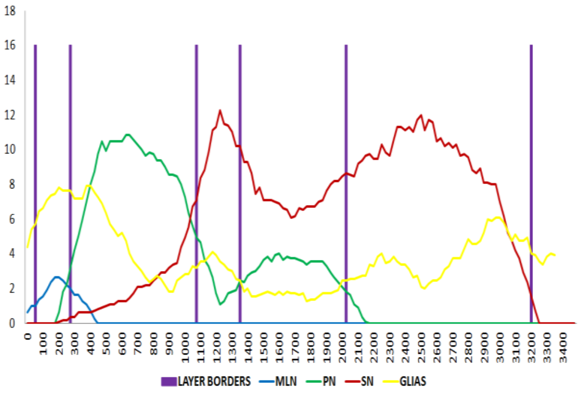


Figure 11: Average cells count along source lines.

l0 is the start of the molecular layer (layer I). *l0* has value as the first *i* in which *av_summary(i) >= level1*. *level1* is a changeable parameter.

l1 is end of layer I and start for layer II-III. *l1* has value as the first *i* in which *av_summary(i) >= level2 && i>l0*. *level2* is a changeable parameter.

l2 is end of layer II-III and start for layer IV. *l2* has value as the first *i* in which *av_sn(i) >= level3 && i>l1*. *Level3* is a changeable parameter.

l3 is end of layer IV and start for layer V. *l3* has value as the first *i* in which *av_sn(i) < level4 && i>l2*. *level4* is a changeable parameter.

l4 is end of layer V and start for layer VI. *l4* has value as the first *i* in which *av_summary(i) >= level5 && i>l3*. *level5* is a changeable parameter.

l5 is end of layer VI. *l5* has value as the first *i* in which *av_sn(i) < level6 && i>l4*. *level6* is a changeable parameter.

Algorithm 4.Rules for layers separation

Reclassification of neurons and astrocytes

Cells reclassification depends from layer number and was described in Alg.5.

LAYER I : If intensity of the cell center pixel is in the range *[min_glia_val, max_glia_val]* the cell is astrocyte. Otherwise, classify this one as molecular layer neuron.

LAYER II-III: If intensity of a cell center pixel is in the range *[min_glia_val, max_glia_val]*, than the cell is an astrocyte. Otherwise, check if the center was classified as stellate neurons:

if *abs((gray(x,y) – pn_average – 20) <= 15),*

than reclassify this center as a pyramidal neuron.

LAYER IV: If intensity of the cell center pixel is in the range *[min_glia_val, max_glia_val]* the cell is astrocyte. Otherwise check if center was classified as pyramidal neuron. If *abs(gray(x,y) – sn_average + 20) <= 15)*, reclassify this center a s stellate neuron and as astrocyte otherwise.

LAYER V: Relassification is not carried out.

LAYER VI: If intensity of the cell center pixel is in the range *[min_glia_val, max_glia_val]* the cell is astrocyte. Otherwise reclassify cell as astrocyte if it had pyramidal type.

Algorithm 5. Rules for cells reclassification in different cortex layers

Final result of layer decomposition and cells classification you can see at Fig. 12. Summary count statistics you can see in the Tab 4. and in the Tab. 5. Distribution cells along image rows you can see at the Fig.11.As you see, isolate stellate neurons has good quality of detection. In touched stellate neurons case algorithm detects more center point than there are actually.

4. EXPERIMENTAL RESULTS

	Layer number	I	II-III	IV	V	VI	Sum
MLN	Human	2	0	0	0	0	2
	Auto	3	0	0	0	0	3
PN	Human	0	27	0	10	0	37
	Auto	0	28	0	12	0	40
SN	Human	0	3	12	11	43	69
	Auto	0	12	11	38	34	95
Astr	Human	2	18	6	16	24	66
	Auto	2	12	3	5	14	36

Table 2. Comparison of the auto-segmentation and human-segmentation cells detection results.

	Auto	Human	From article
Neurons count	67	60	54
Glias count	25	44	33
Summary	92	104	87

Table 3. Comparison results for one cortex column(part of cortex) in our data (auto- and human-detection) with results from [Dav04] (not the same data)

As a result, in layer V we have 3 times more stellate neurons, and in layer VI we have about 20% fewer stellate neurons than actually.

Auto-detected astrocytes are twice smaller than there are actually. The reason is an inadequate contrast of some astrocytes. It's needed to do an additional preprocessing for better results.

The profile of density and neurons count (from the Fig. 10 and Fig. 11) looks similar with profile from [Mey10].

We also compare our results (in absolute values for one cortex column) with results from [Dav04]. In [Dav04] one can find estimation of neuron count. In our work we detect cells and estimate cell type. The result of comparison one can find in the Tab. 3.

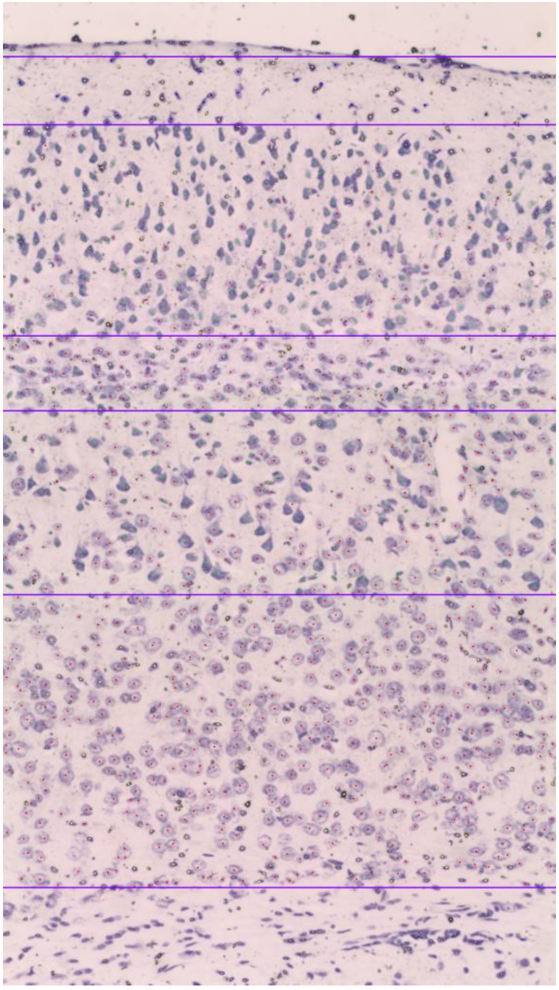


Figure 12. Final neuron and astrocytes centers, and final layer separation.

	Astrocytes	MLN	PN	SN
Layer I	63	49	0	0
Layer II-III	102	0	310	77
Layer IV	39	0	13	123
Layer V	58	0	113	344
Layer VI	113	0	0	358

Table 4. Cells count in one slice.

	Astrocytes	MLN	PN	SN
Layer I	2	1	0	1
Layer II-III	3	0	11	6
Layer IV	2	0	0	7
Layer V	4	0	6	20
Layer VI	14	0	0	15

Table 5. Cells count in one column.

	Detection			
	True positive		False positive	False negative
	Classification			
	Correct	Incorrect		
Summary	731	102	9	52
Astr.	178	-		
MLN	10			
PN	160			
SN	383			

Table 6. Counts of correct (incorrect) cells classification and different types of neuron detection errors

5. CONSLUSION AND FUTURE WORK

Firstly, we propose method for different type cell detection based on morphological operations and different statistics about neuron and astrocytes for all source images (without differences analysis for every cortex layer). Like in [Hey15] and [Kol14] it also depends from the start parameters. Our method allows to get cells localization and estimate size of this cell. The method from [Kol14] allows to get correct edges for every cell. But in [Kol14] there are no classification by neurons type and layers detection. Our method is quite accurate for molecular layer neurons, pyramidal neurons, isolated stellate neurons and astrocytes placed over neurons. Molecular layer neurons and pyramidal neurons have a good quality in the detection and classification. The quality of pyramidal neurons detection is enough for statistics collection, but the quality of stellate neurons detection is not enough. We have good localization results in those layers where there is only one neuron type. But we have inaccurate results in classification of found objects. As future work, we need collect and divide a statistic for every type of neurons by every layer (about color distribution, texture and shape features). It’s necessary to analyze a dependence between segmentation quality and noise deleting techniques. Also we need more information about false-positive and false – negative results for neuron and astrocyte segmentation. For correction of localization and segmentation of such results we need to use the same information from nearest slices (source images).

Secondly, we proposed algorithm for cortex layer detection based on detection statistics. All results placed in the article. The proposed algorithm has as good results in layer decomposition as atlas-based algorithms ([All04], [Sen11]).

Thirdly, we got statistics about neuron localization in every layer. For getting statistics and cells features a stereological study is very popular ([Gar12], [Gia12]). In stereological study one has two main steps: get object boundary and get estimation of volume and shape of the object. \Finding boundaries using automatic algorithms (f.e. Canny) may be useless. For our data, Canny algorithm gives inaccurate result for stellate neurons (for both isolated and attached ones). Our algorithm can estimate the location and size of every cell fully automatically.

As the next step we need to add a machine learning algorithm for cell classification. Also it is interesting to check a layer decomposition with initial layer segmentation based on atlas and ray methods. Finally, we should get comparative results with [All04] and [Bra05].

6. ACKNOWLEDGMENTS

The research was supported by the Federal Target Program of the Ministry of Education and Science of Russia, contract 14.581.21.0016 (RFMEFI58115X0016).

7. REFERENCES

[All04]Allen Brain Atlas. Data Portal, 2004-2006. http://mouse.brain-map.org/

[Bra05]Brain maps, 2005. http://brainmaps.org

[Bas16]Bastiani M., et al. Automatic Segmentation of Human Cortical Layer-Complexes and Architectural Areas Using Ex vivo Diffusion MRI and Its Validation. Front. Neurosci.10:487. doi: 10.3389/fnins.2016.00487, 2016.

[Das15]Das S., Keyser J., Choe Y. Random-forest-based automated cell detection in Knife-Edge Scanning Microscope rat Nissl data. Proceedings of the International Joint Conference On Neural Networks. 2015. DOI: 10.1109/IJCNN.2015.7280852.

[Dav04] Davanlou, M., & Smith, D. (2011). UNBIASED STEREOLOGICAL ESTIMATION OF DIFFERENT CELL TYPES IN RAT CEREBRAL CORTEX. Image Analysis & Stereology, 23(1), 1-11. doi:https://doi.org/10.5566/ias.v23.p1-11

[Gar12]Garcia-Amado M., Prensa L. Stereological Analysis of Neuron, Glial and Endothelial Cell Numbers in the Human Amygdaloid Complex. PLoS ONE 7(6): e38692. doi:10.1371/journal.pone.0038692

[Gia12]Giannaris E., Rosene D. A stereological study of the numbers of neurons and glia in the primary visual cortex across the lifespan of male and female rhesus monkeys. J Comp Neurol. 2012 October 15; 520(15): 3492–3508. doi:10.1002/cne.23101.

[Hey15]He Y., et al. ICut: An integrative cut algorithm enables accurate segmentation of touching cells. Scientific Reports 5(12089), DOI: 10.1038/srep12089, 2015.

[Ing08]Inglis A., et al. Automated identification of neurons and their locations. J. Microsc., 2008 June. 230(Pt 3), pp. 339–352., 2008.

[Kol14]Kolodziejczyk A., Habrat (Ladniak) M., Piorkowski A. Constructing software for analysis of neuron, glial and endothelial cell numbers and density in histological Nissl-stained rodent brain tissue. Journal of medical informatics & technologies, Vol. 23,pp. 77-86, 2014.

[Mes15]Mesejo P., Ugolotti R., Cunto F., Cagnoni S., Giacobini M. Automatic segmentation of hippocampus in histological images of mouse brains using deformable models and random forest. 25th IEEE International Symposium on Computer-Based Medical Systems (CBMS'12), Jun 2012, Rome, Italy. pp.1-4, 2012.

[Mey10] Meyer, H. S., Wimmer, V. C., Oberlaender, M., de Kock, C. P. J., Sakmann, B., & Helmstaedter, M. (2010). Number and Laminar Distribution of Neurons in a Thalamocortical Projection Column of Rat Vibrissal Cortex. Cerebral Cortex (New York, NY), 20(10), 2277–2286. http://doi.org/10.1093/cercor/bhq067

[Sen11]Senyukova, O.V., Lukin, A.S. & Vetrov, D.P. Automated atlas-based segmentation of NISSL-stained mouse brain sections using supervised learning. Programming and Computer Software, Vol. 37, No. 5, pp.245–251, 2011.