

BRNO UNIVERSITY OF TECHNOLOGY  
Faculty of Mechanical Engineering  
Institute of Mathematics

**Ing. Karel Martišek**

**Adaptive Filters for 2-D and 3-D Digital  
Images Processing**

Adaptivní filtry pro 2-D a 3-D zpracování digitálních obrazů

Short version of Ph. D. Thesis

Study field: Applied Mathematics

Supervisor: prof. RNDr. Miloslav Druckmüller, CSc.

Opponents: prof. RNDr. Ivanka Horová, CSc.  
doc. RNDr. Zdeněk Karpíšek, CSc.

**Keywords:** HDR images, confocal microscopy, PSF, digital image, DFT, frequency filters, adaptive histogram equalization, additive noise, 3-D object reconstruction

**Klíčová slova:** HDR obrazy, konfokální mikroskopie, PSF, digitální obraz, DFT, frekvenční filtry, adaptivní ekvalizace histogramu, aditivní šum, 3-D rekonstrukce objektů

The Ph. D. Thesis is available in the library of the Faculty of Mechanical Engineering, Brno University of Technology.

# Contents

<b>1</b>	<b>Motivation</b>	<b>4</b>
<b>2</b>	<b>Aims of the Ph.D. Thesis</b>	<b>4</b>
<b>3</b>	<b>Visualization of High Dynamic Range Images</b>	<b>5</b>
<b>4</b>	<b>Image</b>	<b>6</b>
4.1	Digital Space . . . . .	6
4.2	Digital Image . . . . .	10
<b>5</b>	<b>Digital Geometry Approach to 2-D Image Enhancement</b>	<b>11</b>
5.1	Basic Concepts . . . . .	11
5.2	Adaptive Histogram Equalization . . . . .	12
5.3	Adaptive Histogram Equalization with Adaptive Neighbourhood . . . . .	13
<b>6</b>	<b>Digital Geometry Approach to 3-D Image Enhancement</b>	<b>16</b>
6.1	Basic Concepts . . . . .	16
6.2	Adaptive Histogram Equalization in 3-D . . . . .	17
6.3	Adaptive Histogram Equalization with Adaptive Neighbourhood in 3-D . . .	18
6.4	Comparison of Results . . . . .	21
<b>7</b>	<b>Three-Dimensional Object Reconstruction</b>	<b>25</b>
<b>8</b>	<b>Applications</b>	<b>29</b>
<b>9</b>	<b>Conclusions</b>	<b>30</b>
	<b>References</b>	<b>31</b>
	<b>Author's Publications</b>	<b>33</b>
	<b>Other Author's Products</b>	<b>33</b>
	<b>Author's CV</b>	<b>34</b>
	<b>Abstract</b>	<b>35</b>

# 1 Motivation

Nowadays, digital images are omnipresent; starting with outputs from digital cameras and scanners, and ending with optical cuts provided by confocal microscopes, serving for various biological and medical purposes (the study of the complicated architecture of neural networks, the space structure of cells and tissues, etc.). With the increasing quality of digital images and their expanding applications, the requirements of the quality and the speed of their processing are increasing as well.

With the contrast of 1:300, which is typical for a computer monitor in a medium-lit room, a human eye is capable to distinguish about 150 – 200 levels of brightness. (Although more modern monitors offer theoretically a higher contrast, it is usually unachievable due to non-ideal observing conditions. The contrast of various printing techniques is even lower.) The eight-bit representation of an image offers  $2^8 = 256$  levels of brightness so that this representation is fully sufficient. However, in practice, we often encounter sixteen- and even more bit images (i.e.,  $2^{16} = 65,536$  and more levels of brightness). In these images, there is most of information unexploitable for human sight. That is why it is necessary to use mathematical algorithms to visualize such images.

## 2 Aims of the Ph.D. Thesis

The main aims of the Ph.D. thesis are following:

- Proper and mathematically correct description of fast algorithms of 2-D image enhancement procedures, their extension and improvement with new approaches, and their generalization into 3-D. The 3-D image enhancement procedures have not been described in literature yet. With suitable data, e.g., with a series of optical cuts, these procedures represent a very interesting and unexplored way, which is supposed to contribute to digital images processing significantly. If the 2-D processing does not yield a satisfying outcome, the 3-D processing will be the next step. Biological applications, described at the end of the thesis, can benefit from the enhanced outputs.
- Complex software implementation of the procedures discussed above. Especially the 3-D adaptive image enhancement procedures represent a difficult challenge, due to their high computational costs. An efficient and fast implementation of 3-D object reconstruction from a series of optical cuts is another important part of this software.

### 3 Visualization of High Dynamic Range Images

In image processing, computer graphics, and photography, high dynamic range imaging (HDRI or just HDR) is a set of techniques that allows a greater dynamic range between the lightest and darkest areas of an image than current standard digital imaging techniques or photographic methods. The two main sources of HDR imagery are computer renderings and merging of multiple low-dynamic-range (LDR) or standard-dynamic-range (SDR) photographs.

#### Comparison with Traditional Digital Images

Information stored in HDR images typically corresponds to the physical values of luminance or radiance that can be observed in the real world. This is different from traditional digital images, which represent colours that should appear on a monitor or a paper print. Therefore, HDR image formats are often called „scene-referred“, in contrast to traditional digital images, which are „device-referred“ or „output-referred“. Furthermore, traditional images are usually encoded for the human visual system (maximizing the visual information stored in the fixed number of bits). The values stored for HDR images are often gamma compressed (power law) or logarithmically encoded, or floating-point linear values, since fixed-point linear encodings are increasingly inefficient over higher dynamic ranges.

HDR images often use a higher number of bits per colour channel than traditional images to represent many more colours over a much wider dynamic range. 16-bit or 32-bit floating point numbers are often used to represent HDR pixels.

#### Representing HDR Images on LDR Displays

Popular tone-mapping techniques reduce the dynamic range, or contrast ratio, of the entire image, while retaining localized contrast (between neighbouring pixels), tapping into research on how the human eye and visual cortex perceive a scene, trying to represent the whole dynamic range while retaining realistic colour and contrast. However, these images have often their range over-compressed, creating a surreal low-dynamic-range rendering of a high-dynamic-range scene.

This thesis is concerned with visualization of HDR images using various mathematical algorithms. Let us emphasize that the target of these procedures is to create a suitable image for human sight, not for purposes of a further mathematical analysis. If we want to mathematically analyze the resulting image, it is necessary to maintain the original data and work with them.

## 4 Image

„Image“ is a quite difficult term to be defined precisely and it is used in various vague and unclear ways in many computer science publications. In the following chapters, digital images will be handled and processed and that is why it is necessary to define the term image correctly. This chapter was elaborated using [7].

### 4.1 Digital Space

Throughout this section,  $k \in \{1, 2, \dots, n\}$ ,  $n \in \mathbb{N}$ , holds. The following definitions also contain a few basic terms (equidistant partition of an interval, partition of a set, factor set, vector space, basis, etc.) from the functional analysis. You can find them, e.g., in [16].

#### Definition of Digital Space

**Definition 4.1** (multi-index set). *Let  ${}_k I = \{0, 1, \dots, i_k, \dots, m_k\}$  be index sets. The set  $\mathbf{I}^{(n)} = \bigotimes_{k=1}^n {}_k I$  is called a multi-index set and its element  $\mathbf{i} = [i_1, i_2, \dots, i_k, \dots, i_n]$ ,  $\mathbf{i} \in \mathbf{I}^{(n)}$ , is called a multi-index.*

**Definition 4.2** (support of a digital space). *Let  ${}_k J = \langle a_k, b_k \rangle$  be intervals. The set  $\mathbf{J}^{(n)} = \bigotimes_{k=1}^n {}_k J$  is called an  $n$ -dimensional support of a digital space.*

**Definition 4.3** (multi-partition). *Let  ${}_k D = \{{}_k x_0, {}_k x_1, \dots, {}_k x_{i_k}, \dots, {}_k x_{m_k}\}$  be equidistant partitions of intervals  ${}_k J = \langle a_k, b_k \rangle$ . The set  $\mathbf{D}^{(n)} = \bigotimes_{k=1}^n {}_k D$  is called an equidistant multi-partition of the support  $\mathbf{J}^{(n)}$ .*

**Definition 4.4** (digital space, resolution). *Let  $\mathbf{J}^{(n)} = \bigotimes_{k=1}^n {}_k J$  be a support of a digital space, and let  $\mathbf{D}^{(n)} = \bigotimes_{k=1}^n {}_k D$  be its equidistant multi-partition. The ordered pair  $\mathfrak{D}^{(n)} = (\mathbf{J}^{(n)}, \mathbf{D}^{(n)})$  is called an  $n$ -dimensional digital space. The ordered  $n$ -tuple  $\mathbf{r} = (m_1, m_2, \dots, m_k, \dots, m_n)$  is called the resolution of the space  $\mathfrak{D}^{(n)}$ .*

#### Physical Domain, Physical Space

**Definition 4.5** (physical domain). *A subset  $\mathbf{F}^{(n)} \subset \mathbf{J}^{(n)}$  of a support  $\mathbf{J}^{(n)}$  of a digital space  $\mathfrak{D}^{(n)} = (\mathbf{J}^{(n)}, \mathbf{D}^{(n)})$  is called a physical  $n$ -D domain if, and only if,*

$$\mathbf{F}^{(n)} = \langle {}_1 x_{i_1}, {}_1 x_{i_1+1} \rangle \times \dots \times \langle {}_k x_{i_k}, {}_k x_{i_k+1} \rangle \times \dots \times \langle {}_n x_{i_n}, {}_n x_{i_n+1} \rangle = \bigotimes_{k=1}^n \langle {}_k x_{i_k}, {}_k x_{i_k+1} \rangle.$$

*We write  $\mathbf{F}^{(n)} = \bigotimes_{k=1}^n \langle {}_k x_{i_k}, {}_k x_{i_k+1} \rangle = \mathbf{F}_{[i_1, i_2, \dots, i_k, \dots, i_n]}^{(n)} = \mathbf{F}_{\mathbf{i}}^{(n)}$ . The number  ${}_k v_{\mathbf{i}} = {}_k x_{i_k+1} - {}_k x_{i_k}$ ,  $i_k \in \mathbf{i}$ , is called the  $k$ -th dimension of the physical  $n$ -D domain  $\mathbf{F}^{(n)}$ .*

**Theorem 4.1.**  *$k$ -th dimensions  ${}_k v_i$  of all physical  $n$ -D domains  $\mathbf{F}_i^{(n)} \in \mathfrak{D}^{(n)}$  are equal.*

The previous theorem allows omitting the multi-index from the dimensions of physical  $n$ -D domains, i.e., we can write just  ${}_k v$ . If there is no danger of misunderstanding, we will also omit the  $n$ -D specification, i.e., we will write just domain.

**Theorem 4.2.** *The set  $\mathfrak{F}^{(n)} = \left\{ \mathbf{F}_i^{(n)} = \bigotimes_{k=1}^n \langle {}_k x_{i_k}, {}_k x_{i_k+1} \rangle, i_k \in {}_k I \right\}$  of all physical domains of the support  $\mathbf{J}^{(n)}$  of a digital space  $\mathfrak{D}^{(n)} = (\mathbf{J}^{(n)}, \mathbf{D}^{(n)})$  is a partition of the support  $\mathbf{J}^{(n)}$ .*

**Corollary.** Let  $\mathfrak{D}^{(n)} = (\mathbf{J}^{(n)}, \mathbf{D}^{(n)})$  be a digital space, and let  $A, B \in \mathbf{J}^{(n)}$  be arbitrary points of its support. The relation  $\rho \subset \mathbf{J}^{(n)} \times \mathbf{J}^{(n)}$ , defined as  $\rho(A, B) \Leftrightarrow \exists \mathbf{F}_i \in \mathfrak{F}^{(n)} : A \in \mathbf{F}_i \wedge B \in \mathbf{F}_i$ , is an equivalence on  $\mathbf{J}^{(n)}$ .

**Definition 4.6** (physical space). *The factor set  $\mathfrak{F}^{(n)} = \mathbf{J}^{(n)} / \rho$  from the previous corollary is called a physical space of a support  $\mathbf{J}^{(n)}$ , or, of a digital space  $\mathfrak{D}^{(n)} = (\mathbf{J}^{(n)}, \mathbf{D}^{(n)})$ . The resolution of the physical space  $\mathfrak{F}^{(n)}$  is understood as the resolution of the digital space  $\mathfrak{D}^{(n)}$ .*

Let us note that two important terms will be used throughout the thesis: a *pixel*, as the smallest displayable element on a given output device, and a *voxel*, as the smallest volume element. According to the theory above, both of these objects are just special cases of domains; a pixel is a physical 2-D domain, and a voxel is a physical 3-D domain.

## Logical Domain, Logical Space, Mapping

**Definition 4.7** (logical space, logical domain). *Let  $\mathfrak{F}^{(n)}$  be a physical space of a digital space  $\mathfrak{D}^{(n)}$ , and let  ${}_k v$  be dimensions of its physical domains. Let  $\mathbf{C} \in \mathbf{J}^{(n)} : \mathbf{C} = [c_1, c_2, \dots, c_k, \dots, c_n], c_k \in \langle 0, {}_k v \rangle$ . The set*

$${}_C \mathfrak{L}^{(n)} = \bigotimes_{k=1}^n \{ {}_k r_{i_k} \in \mathbb{R} \mid \forall k \in \{1, 2, \dots, n\} : {}_k r_{i_k} \in \langle {}_k x_{i_k}, {}_k x_{i_k+1} \rangle \wedge ({}_k r_{i_k} - {}_k x_{i_k} = c_k) \}$$

*is called a logical space of the digital space  $\mathfrak{D}^{(n)} = (\mathbf{J}^{(n)}, \mathbf{D}^{(n)})$ , and its elements  ${}_C L_i$ ,  $\mathbf{i} = [i_1, i_2, \dots, i_k, \dots, i_n]$ , are called logical domains. The resolution of the logical space  ${}_C \mathfrak{L}^{(n)}$  is understood as the resolution of the digital space  $\mathfrak{D}^{(n)}$ .*

**Theorem 4.3.** *Let  $\mathfrak{F}^{(n)}$  be a physical space and  ${}_C \mathfrak{L}^{(n)}$  a logical space of the same digital space  $\mathfrak{D}^{(n)} = (\mathbf{J}^{(n)}, \mathbf{D}^{(n)})$ . A mapping  ${}_C \varphi : \mathfrak{F}^{(n)} \rightarrow {}_C \mathfrak{L}^{(n)}$  such that  ${}_C \varphi(\mathbf{F}_i) = {}_C L_i \Leftrightarrow {}_C L_i \in \mathbf{F}_i$  is bijective.*

**Definition 4.8** (mapping of physical space, control point). *The mapping  ${}_C \varphi : \mathfrak{F}^{(n)} \rightarrow {}_C \mathfrak{L}^{(n)}$  from the previous theorem is called a mapping of the physical space  $\mathfrak{F}^{(n)}$ . The point  $\mathbf{C} \in \mathbf{J}^{(n)} : \mathbf{C} = [c_1, c_s, \dots, c_k, \dots, c_n], c_k \in \langle 0, {}_k v \rangle$ , is called its control point.*

As a corollary of Theorem 4.3, for each mapping  ${}_C\varphi : \mathfrak{F}^{(n)} \rightarrow {}_C\mathfrak{L}^{(n)}$ , there exists the inverse mapping, i.e.,  ${}_C\varphi^{-1} : {}_C\mathfrak{L}^{(n)} \rightarrow \mathfrak{F}^{(n)}$ . Thanks to this it is possible to assign a unique logical domain  ${}_C L_i$  to each physical domain  $\mathbf{F}_i$  using the mapping  ${}_C\varphi$ , and, vice versa, to assign a unique physical domain  $\mathbf{F}_i$  to each logical domain  ${}_C L_i$  using the mapping  ${}_C\varphi^{-1}$ .

**Definition 4.9** (vertex and central mapping). *The mapping  ${}_V\varphi : \mathfrak{F}^{(n)} \rightarrow {}_V\mathfrak{L}^{(n)}$  with the control point  $\mathbf{V} = [0, 0, \dots, 0]$  is called the vertex mapping. The mapping  ${}_S\varphi : \mathfrak{F}^{(n)} \rightarrow {}_S\mathfrak{L}^{(n)}$  with the control point  $\mathbf{S} = [\frac{1v}{2}, \frac{2v}{2}, \dots, \frac{kv}{2}, \dots, \frac{nv}{2}]$  is called the central mapping.*

**Definition 4.10** (global coordinate system). *Let  $\mathfrak{D}^{(n)} = (\mathbf{J}^{(n)}, \mathbf{D}^{(n)})$  be a digital space,  ${}_kv$  the dimensions of its domains, and  ${}_C\varphi : \mathfrak{F}^{(n)} \rightarrow {}_C\mathfrak{L}^{(n)}$  an arbitrary mapping. Let  $\mathbb{R}^n$  be the  $n$ -dimensional real vector space with the basis  $\{\mathbf{e}_k\}_{k=1}^n$ ,  $\mathbf{e}_k = (0, 0, \dots, {}_kv, \dots, 0)$ . The ordered  $(n+2)$ -tuple  $\mathcal{L}^{(n)} = \langle \mathfrak{L}^{(n)}, \mathbf{S}, \mathbf{e}_1, \mathbf{e}_2, \dots, \mathbf{e}_k, \dots, \mathbf{e}_n \rangle$  is called a global coordinate system of the logical space  $\mathfrak{L}^{(n)}$ . The ordered  $(n+2)$ -tuple  $\mathcal{F}^{(n)} = \langle \mathfrak{F}^{(n)}, \mathbf{S}, \mathbf{e}_1, \mathbf{e}_2, \dots, \mathbf{e}_k, \dots, \mathbf{e}_n \rangle$  is called a global coordinate system of the physical space  $\mathfrak{F}^{(n)}$  induced by the mapping  ${}_C\varphi$ . The ordered  $(n+2)$ -tuple  $\mathcal{D}^{(n)} = \langle \mathfrak{D}^{(n)}, \mathbf{S}, \mathbf{e}_1, \mathbf{e}_2, \dots, \mathbf{e}_k, \dots, \mathbf{e}_n \rangle$  is called a global coordinate system of the digital space  $\mathfrak{D}^{(n)}$  induced by the mapping  ${}_C\varphi$ .*

Let us note that the digital space  $\mathfrak{D}^{(2)}$  is called the digital plane, its physical and logical spaces  $\mathfrak{F}^{(2)}$  and  $\mathfrak{L}^{(2)}$  are called physical and logical planes, and their elements  $\mathbf{F}_i^{(2)}$  and  $\mathbf{L}_i^{(2)}$  are called physical and logical pixels.

## Digital Space Metrics

**Theorem 4.4.** *Let  $\mathfrak{F}^{(n)}$  be a physical space,  $c_k \in \mathbb{R}^+$ , and  $\mathcal{E}_{\mathfrak{F}}^{(n)}, \mathcal{P}_{\mathfrak{F}}^{(n)}, \mathcal{C}_{\mathfrak{F}}^{(n)}$  mappings  $\mathfrak{F}^{(n)} \times \mathfrak{F}^{(n)} \rightarrow \mathbb{R}$  such that  $\mathcal{E}_{\mathfrak{F}}^{(n)}(\mathbf{F}_i^{(n)}, \mathbf{F}_j^{(n)}) = \sqrt{\sum_{k=1}^n c_k^2 (i_k - j_k)^2}$ ,  $\mathcal{P}_{\mathfrak{F}}^{(n)}(\mathbf{F}_i^{(n)}, \mathbf{F}_j^{(n)}) = \sum_{k=1}^n c_k |i_k - j_k|$ ,  $\mathcal{C}_{\mathfrak{F}}^{(n)}(\mathbf{F}_i^{(n)}, \mathbf{F}_j^{(n)}) = \max \{c_k |i_k - j_k|\}_{k=1}^n$ . Then  $\mathcal{E}_{\mathfrak{F}}^{(n)}, \mathcal{P}_{\mathfrak{F}}^{(n)}, \mathcal{C}_{\mathfrak{F}}^{(n)}$  are metrics on  $\mathfrak{F}^{(n)}$ .*

**Definition 4.11** (weighted Euclidean, taxicab and maximum metric of physical space). *The metrics  $\mathcal{E}_{\mathfrak{F}}^{(n)}, \mathcal{P}_{\mathfrak{F}}^{(n)}, \mathcal{C}_{\mathfrak{F}}^{(n)}$  from the previous theorem are called the (weighted) Euclidean, taxicab and maximum metric of the physical space  $\mathfrak{F}^{(n)}$ , respectively.*

**Theorem 4.5.** *Let  $\mathfrak{L}^{(n)}$  be a logical space,  $c_k \in \mathbb{R}^+$ , and  $\mathcal{E}_{\mathfrak{L}}^{(n)}, \mathcal{P}_{\mathfrak{L}}^{(n)}, \mathcal{C}_{\mathfrak{L}}^{(n)}$  mappings  $\mathfrak{L}^{(n)} \times \mathfrak{L}^{(n)} \rightarrow \mathbb{R}$  such that  $\mathcal{E}_{\mathfrak{L}}^{(n)}(\mathbf{L}_i^{(n)}, \mathbf{L}_j^{(n)}) = \sqrt{\sum_{k=1}^n c_k^2 (i_k - j_k)^2}$ ,  $\mathcal{P}_{\mathfrak{L}}^{(n)}(\mathbf{L}_i^{(n)}, \mathbf{L}_j^{(n)}) = \sum_{k=1}^n c_k |i_k - j_k|$ ,  $\mathcal{C}_{\mathfrak{L}}^{(n)}(\mathbf{L}_i^{(n)}, \mathbf{L}_j^{(n)}) = \max \{c_k |i_k - j_k|\}_{k=1}^n$ . Then  $\mathcal{E}_{\mathfrak{L}}^{(n)}, \mathcal{P}_{\mathfrak{L}}^{(n)}, \mathcal{C}_{\mathfrak{L}}^{(n)}$  are metrics on  $\mathfrak{L}^{(n)}$ .*

**Definition 4.12** (weighted Euclidean, taxicab and maximum metric of logical space). *The metrics  $\mathcal{E}_{\mathfrak{L}}^{(n)}, \mathcal{P}_{\mathfrak{L}}^{(n)}, \mathcal{C}_{\mathfrak{L}}^{(n)}$  from the previous theorem are called the (weighted) Euclidean, taxicab and maximum metric of the logical space  $\mathfrak{L}^{(n)}$ , respectively.*

**Definition 4.13** (physical domain neighbour). Let  $\mathbf{F}_i^{(n)}, \mathbf{F}_j^{(n)}$  be two different physical domains of the same physical space  $\mathfrak{F}^{(n)}$ , and  $\overline{\mathbf{F}}_i^{(n)}, \overline{\mathbf{F}}_j^{(n)}$  their closures. The domain  $\mathbf{F}_j^{(n)}$  is called a neighbour of the domain  $\mathbf{F}_i^{(n)}$  if, and only if,  $\overline{\mathbf{F}}_i^{(n)} \cap \overline{\mathbf{F}}_j^{(n)} \neq \emptyset$ .

**Definition 4.14** (logical domain neighbour). Let  $\mathfrak{F}^{(n)}$  be a physical space,  $\mathbf{C}\mathfrak{L}^{(n)}$  a logical space of the same digital space  $\mathfrak{D}^{(n)}$ , and  $\mathbf{L}_i^{(n)}, \mathbf{L}_j^{(n)} \in \mathbf{C}\mathfrak{L}^{(n)}$  such that  $\mathbf{L}_i^{(n)} = \mathbf{C}\varphi(\mathbf{F}_i^{(n)})$ ,  $\mathbf{L}_j^{(n)} = \mathbf{C}\varphi(\mathbf{F}_j^{(n)})$ . The domain  $\mathbf{L}_j^{(n)}$  is called a neighbour of the domain  $\mathbf{L}_i^{(n)}$  if, and only if,  $\mathbf{F}_j^{(n)}$  is a neighbour of  $\mathbf{F}_i^{(n)}$ .

**Theorem 4.6.** A physical domain  $\mathbf{F}_j^{(n)}$  is a neighbour of a domain  $\mathbf{F}_i^{(n)}$  if, and only if,  $\mathcal{C}_{\mathfrak{F}}^{(n)}(\mathbf{F}_i^{(n)}, \mathbf{F}_j^{(n)}) = 1$ .

## Valuation and Digital Objects

**Definition 4.15** (binary valuation of physical space). Let  $\mathfrak{F}^{(n)}$  be a physical space. A mapping  $\beta_{\mathfrak{F}} : \mathfrak{F}^{(n)} \rightarrow \{0, 1\}$  is called a binary valuation of the physical space  $\mathfrak{F}^{(n)}$ .

**Definition 4.16** (binary valuation of logical space). Let  $\mathfrak{F}^{(n)}$  be a physical space and  $\beta_{\mathfrak{F}} : \mathfrak{F}^{(n)} \rightarrow \{0, 1\}$  its binary valuation. Let  $\mathbf{C}\mathfrak{L}^{(n)}$  be a logical space of the same digital space  $\mathfrak{D}^{(n)}$  and let  $\mathbf{C}\varphi : \mathfrak{F}^{(n)} \rightarrow \mathbf{C}\mathfrak{L}^{(n)}$  be a mapping. A mapping  $\beta_{\mathfrak{L}} : \mathbf{C}\mathfrak{L}^{(n)} \rightarrow \{0, 1\}$  such that

$$\forall \mathbf{L}_i^{(n)} \in \mathbf{C}\mathfrak{L}^{(n)} : \beta_{\mathfrak{L}}(\mathbf{L}_i^{(n)}) = 1 \Leftrightarrow \left( \mathbf{C}\varphi^{-1}(\mathbf{L}_i^{(n)}) = \mathbf{F}_i^{(n)} \right) \wedge \left( \beta_{\mathfrak{F}}(\mathbf{F}_i^{(n)}) = 1 \right)$$

is called a binary valuation of the logical space  $\mathbf{C}\mathfrak{L}^{(n)}$ .

**Definition 4.17** (general valuation of physical space). Let  $A$  be any set with at least two elements and let  $\mathfrak{F}^{(n)}$  be a physical space. A mapping  $\beta_{\mathfrak{F}} : \mathfrak{F}^{(n)} \rightarrow A$  is called a general valuation of the physical space  $\mathfrak{F}^{(n)}$ .

**Definition 4.18** (general valuation of logical space). Let  $A$  be any set with at least two elements, let  $\mathfrak{F}^{(n)}$  be a physical space and  $\beta_{\mathfrak{F}} : \mathfrak{F}^{(n)} \rightarrow A$  its general valuation. Let  $\mathbf{C}\mathfrak{L}^{(n)}$  be a logical space of the same digital space  $\mathfrak{D}^{(n)}$  and let  $\mathbf{C}\varphi : \mathfrak{F}^{(n)} \rightarrow \mathbf{C}\mathfrak{L}^{(n)}$  be a mapping. A mapping  $\beta_{\mathfrak{L}} : \mathbf{C}\mathfrak{L}^{(n)} \rightarrow A$  such that

$$\forall \mathbf{L}_i^{(n)} \in \mathbf{C}\mathfrak{L}^{(n)}, \forall a \in A : \beta_{\mathfrak{L}}(\mathbf{L}_i^{(n)}) = a \Leftrightarrow \left( \mathbf{C}\varphi^{-1}(\mathbf{L}_i^{(n)}) = \mathbf{F}_i^{(n)} \right) \wedge \left( \beta_{\mathfrak{F}}(\mathbf{F}_i^{(n)}) = a \right)$$

is called a general valuation of the logical space  $\mathbf{C}\mathfrak{L}^{(n)}$ .

If there is no need to distinguish a logical and physical space, we will write  $\beta : \mathfrak{D}^{(n)} \rightarrow A$  and call it a general valuation of the digital space. For practical purposes, valuations with number sets are used.

**Definition 4.19** (numerical valuation of digital space). Let  $\beta : \mathfrak{D}^{(n)} \rightarrow A$  be a valuation of a digital space. If  $A$  is a number set, the valuation  $\beta$  is called a numerical valuation. In particular, if  $A \subset \mathbb{N}$  ( $A \subset \mathbb{Z}, A \subset \mathbb{Q}, A \subset \mathbb{R}, A \subset \mathbb{C}$ ), the valuation  $\beta$  is called natural (integer, rational, real, complex).

**Definition 4.20** ( $m$ -ary valuation of digital space). Let  $M$  be any set with  $m$  elements and let  $\mathfrak{D}^{(n)}$  be a digital space. A mapping  $\beta : \mathfrak{D}^{(n)} \rightarrow M$  is called an  $m$ -ary valuation of the digital space  $\mathfrak{D}^{(n)}$ .

## 4.2 Digital Image

**Definition 4.21** (discretized function). Let  $\mathbf{I}^{(n)}$  be a support of a digital space. A function  $g(\mathbf{x})$ , defined for every  $\mathbf{x} = [x_1, x_2, \dots, x_n] \in \mathbb{R}^n$ , is called discretized if, and only if,  $\forall \mathbf{x} \in \mathbf{I}^{(n)} : g(\mathbf{x}) = g(\lfloor \mathbf{x} \rfloor)$  where  $\lfloor \mathbf{x} \rfloor = [\lfloor x_1 \rfloor, \lfloor x_2 \rfloor, \dots, \lfloor x_n \rfloor]$ .

**Definition 4.22** (image). Let  $W = \langle 0, w \rangle \subset \mathbb{R}$ ,  $w \in \mathbb{N}$  (Width),  $H = \langle 0, h \rangle \subset \mathbb{R}$ ,  $h \in \mathbb{N}$  (Height), and  $V = \langle v_1, v_2 \rangle \subset \mathbb{R}$  (Value Set) be intervals. A function  $I : W \times H \rightarrow V$  is called an (analog) image.

If the function  $I$  is discretized, we talk about a *discretized image*. If the domain  $W \times H$  of the function  $I$  is a physical (logical) plane, we talk about a *physical (logical) image*. The resolution of a physical (logical) image is understood as the resolution of its support. If the function  $I$  is discretized and  $H \subset \mathbb{N}$ , we talk about a *digital image*.

The last definition is very general and is fulfilled by any kind of „image“; a photo, a map, a drawing, and even a real visual perception of the surroundings. Everything depends on the codomain  $V$  of this image. If an analog image is supposed to be processed by a computer, it has to be converted into a digital one at first. This process is called *digitization of the image*.

Digitization occurs in two parts: discretization and quantization.

- *Discretization* means the reading of an analog signal, and, at regular time intervals (frequency), sampling the value of the signal at the point. Each such reading is called a sample and may be considered to have infinite precision at this stage.
- *Quantization* means the rounding of the samples to a fixed set of numbers (such as integers).

According to the previous theory, a digital image is an  $m$ -ary valuation of a digital plane.

Throughout the following text, we will work with digital images only. That is why the word „digital“ will be omitted.

## 5 Digital Geometry Approach to 2-D Image Enhancement

One of the main attributes of the human eye, comparing to digital cameras, is adaptivity, i.e., local adjustment of the aperture, sensitivity, white balance, etc. Mathematical methods based on this principle do not process the whole image but just some local neighbourhood of each pixel. This local neighbourhood is constructed by means of methods that simulate the attributes of the human eye; it is a set of pixels that belong together, according to a chosen characteristic (e.g., intensity).

In this chapter, a description and explanation of various kinds of 2-D adaptive histogram equalization procedures is given. These procedures are rarely described in the literature available, and if they are, then in a very vague way. One of the main tasks and targets of this thesis is to formalize the known theory, to extend it by introducing more complex algorithms, thus improving the adaptive histogram equalization performance, and to implement everything in the software developed as a part of this thesis.

Let us note that these 2-D procedures work within a single particular image. However, with suitable data, e.g., a series of optical cuts provided by a confocal microscope, it is reasonable to take into account also the adjacent images (cuts). This generalization into 3-D is discussed in Chapter 6 and constitutes the next task and target of this thesis.

### 5.1 Basic Concepts

*Digital image*  $I$  is an  $m$ -ary valuation  $\beta$  of the digital plane  $\mathfrak{D}^{(2)}$  (see Definition 4.19). Depending on  $m$ , an image can be eight-bit, sixteen-bit, grayscale, colour, etc.

*Physical pixel* (the smallest displayable element on a given output device)  $\mathbf{F}_{\mathbf{i}}^{(2)}$ ,  $\mathbf{i} = [i_1, i_2]$ , is a physical 2-D domain (see Definition 4.5) and an element of the physical plane  $\mathfrak{F}^{(2)}$ .

*Logical pixel*  $\mathbf{L}_{\mathbf{i}}^{(2)}$ ,  $\mathbf{i} = [i_1, i_2]$ , is a logical 2-D domain (see Definition 4.7) and an element of the logical plane  $\mathfrak{L}^{(2)}$ . For each  $\mathbf{i} \in \mathbf{I}^{(2)}$ , the logical pixel  $\mathbf{L}_{\mathbf{i}}^{(2)}$  is uniquely assigned to the physical pixel  $\mathbf{F}_{\mathbf{i}}^{(2)}$  using the central mapping  $\mathbf{s}\varphi$  (see Definition 4.9).

Let us note that the simplest algorithm for constructing curves given by the equation  $f(x, y) = 0$  uses the difference between physical and logical pixels and the vertex mapping, see [7], pg. 72, or [6]. For the purposes of this thesis, the algorithm of the line segment construction using the central mapping will be sufficient; see [10].

**Remark.** If it is clear which digital space we are working with, the specification of its dimension will be omitted. If the coordinates of the physical (logical) pixel  $\mathbf{F}_{\mathbf{i}}^{(2)}$  ( $\mathbf{L}_{\mathbf{i}}^{(2)}$ ) are

unknown and/or not decisive (i.e., we are only interested if a given pixel does or does not belong to a certain set), a notation  $\mathbf{F}_i(\mathbf{L}_i)$ ,  $i \in \mathbb{N}_0$ , will be used.

*Image matrix* is an  $M$ -by- $N$  matrix,  $(M, N)$  is the resolution of the digital plane  $\mathfrak{D}^{(2)}$  (see Definition 4.4), whose elements are the values  $\beta(\mathbf{F}_{\mathbf{i}}^{(2)})$ ,  $\beta(\mathbf{L}_{\mathbf{i}}^{(2)})$  of individual physical (logical) pixels  $\mathbf{F}_{\mathbf{i}}^{(2)}$ ,  $\mathbf{L}_{\mathbf{i}}^{(2)}$ ,  $\mathbf{i} = [i_1, i_2]$ .

## 5.2 Adaptive Histogram Equalization

Histogram equalization provides a possibility of modifying the dynamic range and contrast of an image by altering the image in the way that the output image contains a more uniform distribution of intensities (the intensity histogram is flatter and covers the whole range of the intensities). The transformation function is given by the cumulative histogram of the original image.

Let us consider an image  $I$ , i.e., a valuation  $\beta : \mathfrak{D}^{(2)} \rightarrow \mathbb{N}$ . The histogram equalization is implemented as follows:

- The histogram of intensities is taken. Let us denote  $h(i)$  the number of pixels with the intensity equal to  $i$ ,  $i = 0, \dots, L$ , where  $L$  represents the maximum possible intensity value, given by the codomain of the valuation  $\beta$  (most often,  $L = 255$  or  $L = 65,535$ ).
- The cumulative histogram of intensities is calculated. Let us denote  $c(i)$  the number of pixels with the intensity less or equal to  $i$ ,  $i = 0, \dots, L$ , i.e.,  $c(i) = \sum_{j=0}^i h(j)$ .
- Now the histogram equalization assigns to the pixel  $\mathbf{F}$  with the original intensity  $\beta(\mathbf{F}) = i$  the output value  $\mathbf{F}_{\text{out}}(i)$  given by the formula

$$\beta(\mathbf{F})_{\text{out}} = \text{Round} \left( \frac{c(i) - c_{\min}}{M \cdot N - c_{\min}} \cdot L \right), \quad i = 0, \dots, L$$

where  $c_{\min}$  is the minimum non-zero value of  $c$ , and  $M, N$  represent the width and height of the input image  $I$ .

However, this histogram processing method is global (in the sense that it applies a transformation function based on the intensity level distribution of the entire image). Although this method can enhance the overall contrast and dynamic range of the image (and thereby making certain details more visible), there are many cases in which enhancement of details over small areas (i.e., the areas with a negligible influence on the global transformation function because their number of pixels is very small comparing to the number of pixels of the entire image) is desired.

The solution is to derive the transformation function based on the intensity distribution within the local neighbourhood of the processed pixel in order to simulate the adaptivity of the human eye.

**Definition 5.1** (local neighbourhood). *Let  $I$  be a digital image in the spatial domain,  $\mathbf{s}\varphi : \mathfrak{F}^{(2)} \rightarrow \mathfrak{L}^{(2)}$  the central mapping,  $\mathbf{F}_0 \in I$  a processed pixel, and  $r \in \mathbb{N}$ . The set  $O^{(2)} = \left\{ \mathbf{F} \in I : \mathcal{C}_{\mathfrak{L}}^{(2)}(\mathbf{s}\varphi(\mathbf{F}), \mathbf{s}\varphi(\mathbf{F}_0)) \leq r \right\}$  is called a local neighbourhood of the pixel  $\mathbf{F}_0$  of the size (radius)  $r$ .*

Note that the maximum metric with  $c_1 = c_2 = 1$  was used (see Definition 4.12).

### 5.3 Adaptive Histogram Equalization with Adaptive Neighbourhood

The previous method works well only if strong interfaces do not appear in the original image. If they do, it is necessary to use the adaptive histogram equalization with an adaptive neighbourhood. The neighbourhood of the processed pixel has a variable shape respecting the interfaces that appear in the original image.

In the literature available, there is very little information on these methods. Let us explore now the adaptive neighbourhoods used most often, together with a suggestion of a few improvements.

**Definition 5.2** ( $V(\alpha)$ -neighbourhood). *Let  $O^{(2)}$  be a (non-adaptive) local neighbourhood (according to Definition 5.1) of a processed pixel  $\mathbf{F}_0$ , let  $\beta : \mathfrak{D}^{(2)} \rightarrow \mathbb{N}$  be a natural valuation of the digital plane  $\mathfrak{D}^{(2)}$  (according to Definition 4.19), and  $\alpha \in \mathbb{N}$ . The set  $O_V^{(2)} = \left\{ \mathbf{F} \in O^{(2)} : |\beta(\mathbf{F}) - \beta(\mathbf{F}_0)| \leq \alpha \right\}$  is called a  $V(\alpha)$ -neighbourhood of the pixel  $\mathbf{F}_0$ .*

**Definition 5.3** ( $A(k)$ -neighbourhood). *Let  $O^{(2)}$  be a (non-adaptive) local neighbourhood (according to Definition 5.1) with  $n$  elements of a processed pixel  $\mathbf{F}_0$ , let  $\beta : \mathfrak{D}^{(2)} \rightarrow \mathbb{N}$  be a natural valuation of the digital plane  $\mathfrak{D}^{(2)}$  (according to Definition 4.19), and  $k \in \mathbb{N}$ ,  $k \leq n$ . Let us consider a sequence  $\{\mathbf{F}_i\}_{i=0}^n$  such that  $\mathbf{F}_i \in O^{(2)}$  and  $|\beta(\mathbf{F}_i) - \beta(\mathbf{F}_0)| \leq |\beta(\mathbf{F}_{i+1}) - \beta(\mathbf{F}_0)|$ ,  $i = 1, \dots, n-1$ . The set  $O_A^{(2)} = \{\mathbf{F}_i : i = 1, \dots, k\}$  is called an  $A(k)$ -neighbourhood of the pixel  $\mathbf{F}_0$ .*

The software developed as a part of this thesis employs a combined neighbourhood, according to the following definition.

**Definition 5.4** (combined  $S(\alpha, k, q_V, q_A)$ -neighbourhood). *Let  $O^{(2)}$  be a (non-adaptive) local neighbourhood (according to Definition 5.1) with  $n$  elements of a processed pixel  $\mathbf{F}_0$ , let  $\beta : \mathfrak{D}^{(2)} \rightarrow \mathbb{N}$  be a natural valuation of the digital plane  $\mathfrak{D}^{(2)}$  (according to Definition 4.19),  $\alpha, k \in \mathbb{N}$ ,  $k \leq n$ , and  $q_V, q_A \in \mathbb{R}$ ,  $q_V, q_A \geq 1$ . Let us consider a sequence  $\{\mathbf{F}_i\}_{i=0}^n$  such*

that  $\mathbf{F}_i \in O^{(2)}$  and  $|\beta(\mathbf{F}_i) - \beta(\mathbf{F}_0)| \leq |\beta(\mathbf{F}_{i+1}) - \beta(\mathbf{F}_0)|$ ,  $i = 1, \dots, n-1$ . Let  $O_V^{(2)}$  be the  $V(\alpha)$ -neighbourhood (according to Definition 5.2) of  $\mathbf{F}_0$ , and  $O_A^{(2)}$  the  $A(k)$ -neighbourhood (according to Definition 5.3) of  $\mathbf{F}_0$ . The set  $O_S^{(2)} = (O_V^{(2)} \cap O_A^{(2)}) \cup O'_S$  where

$$O'_S = \begin{cases} \left\{ \mathbf{F}_i \in O_A^{(2)} \setminus O_V^{(2)} : |\beta(\mathbf{F}_i) - \beta(\mathbf{F}_0)| \leq q_V \cdot \alpha \right\} & \text{for } O_V^{(2)} \subset O_A^{(2)}, \\ \left\{ \mathbf{F}_i \in O_V^{(2)} \setminus O_A^{(2)} : i = k+1, \dots, \min\{\lfloor q_A \cdot k \rfloor, |O_V^{(2)}|\} \right\} & \text{for } O_A^{(2)} \subset O_V^{(2)}, \\ \emptyset & \text{for } O_V^{(2)} = O_A^{(2)}, \end{cases}$$

is called an  $S(\alpha, k, q_V, q_A)$ -neighbourhood of the pixel  $\mathbf{F}_0$ . If  $O_S^{(2)} = \emptyset$ , the equalization is not performed and the processed pixel  $\mathbf{F}_0$  keeps its original value.

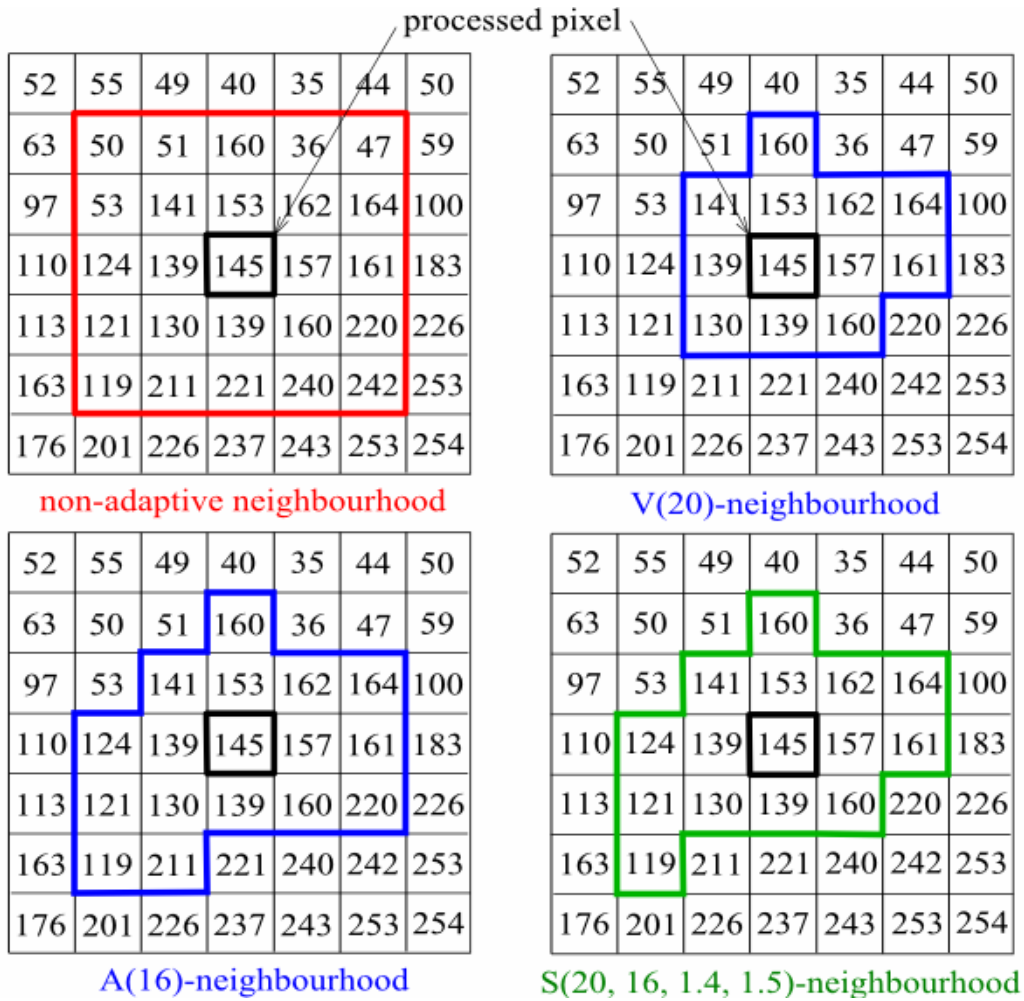


Figure 1:  $S(20, 16, 1.4, 1.5)$ -neighbourhood with  $r=2$  compared to the non-adaptive neighbourhood,  $V(20)$ -neighbourhood and  $A(16)$ -neighbourhood

Figure 1 shows a comparison of all the neighbourhoods discussed so far. The non-adaptive neighbourhood contains all the pixels within the chosen radius ( $r=2$ ), including the values that are significantly smaller or bigger than the value of the processed pixel  $\mathbf{F}_0$ . The  $V(20)$ -neighbourhood correctly denies the outlying values but also three values (124, 121 and 119

in the bottom-left corner) that could be used for the equalization because they differ from the processed pixel much less than from the surroundings. On the contrary, the A(16)-neighbourhood includes all the correct values but also two values (211 and 220) that should not belong to the other ones. The combined S(20, 16, 1.4, 1.5)-neighbourhood removes the imperfections of the previous cases.

Less formally, as a corollary of Definitions 5.2 and 5.3, one of the conditions given by  $\alpha$  and  $k$  is always „stronger“ than the other one, i.e., either  $O_V^{(2)} \subseteq O_A^{(2)}$  or  $O_A^{(2)} \subseteq O_V^{(2)}$  holds. The combined neighbourhood contains the pixels that fulfill both conditions simultaneously ( $O_V^{(2)} \cap O_A^{(2)}$ ), together with the pixels that fulfill only the weaker condition and do not break the stronger condition „too much“ (measured by  $q_V$  and  $q_A$ ).

Let us note that if  $O_V^{(2)} = O_A^{(2)}$ , then also  $O_V^{(2)} = O_A^{(2)} = O_S^{(2)}$ . If  $q_V = q_A = 1$ , then  $O_S^{(2)} = O_V^{(2)} \cap O_A^{(2)}$ .

Again, the application of this method to HDR images is shown in Section 6.4.

## Fuzzy Neighbourhood

The previous sections introduced various kinds of local neighbourhoods. The question was whether a certain pixel does or does not belong to a neighbourhood of the processed pixel, i.e., to a set of pixels constructed according to the given conditions.

This section is concerned with another aspect: we suppose that the pixels lying closer to the processed pixel (in the sense of the selected metric) are more reliable and more meaningful than the pixels lying further away (within the selected radius  $r$ ). It means that some local neighbourhood is constructed, as before, and then the membership degree is assigned to each pixel from this neighbourhood, according to its distance (again, in the sense of the selected metric) from the processed pixel.

**Definition 5.5** (fuzzy local neighbourhood). *Let  $I$  be a digital image in the spatial domain,  $\mathbf{s}\varphi : \mathfrak{F}^{(2)} \rightarrow \mathfrak{L}^{(2)}$  the central mapping,  $\mathbf{F}_0 \in I$  a processed pixel,  $r \in \mathbb{N}$ , and  $\mu_r \in \mathbb{R}$ ,  $0 < \mu_r \leq 1$ . Let us consider a function  $\mu : \mathfrak{D}^{(2)} \rightarrow \mathbb{R}^+$  defined as*

$$\mu(\mathbf{F}) = 1 - \frac{1 - \mu_r}{r} \cdot \mathcal{C}_{\mathfrak{L}}^{(2)}(\mathbf{s}\varphi(\mathbf{F}), \mathbf{s}\varphi(\mathbf{F}_0)).$$

*Then the set  $\tilde{O}^{(2)} = \left\{ [\mathbf{F}, \mu(\mathbf{F})] : \mathbf{F} \in I, \mathcal{C}_{\mathfrak{L}}^{(2)}(\mathbf{s}\varphi(\mathbf{F}), \mathbf{s}\varphi(\mathbf{F}_0)) \leq r \right\}$  is called a fuzzy local neighbourhood of the pixel  $\mathbf{F}_0$ .*

## 6 Digital Geometry Approach to 3-D Image Enhancement

All the methods described in Chapter 5 work in 2-D, i.e., they are always performed within a single image. However, with suitable data, which can be understood as a 3-D image, e.g., with a series of optical cuts obtained by means of a confocal microscope, an idea on a generalization of these 2-D procedures into 3-D arises. In 3-D image enhancement procedures, each pixel would be processed with regards not only to adjacent pixels in the given optical cut, but also with regards to adjacent pixels in the previous and following optical cuts.

These 3-D procedures have not been described in literature so far. Since they use more input information, they are supposed to contribute to the quality of displaying cells and other structures significantly (e.g., the visualization of tiny corpuscles, originally almost imperceptible due to a bad contrast). The description and implementation of the 3-D image enhancement procedures (especially various kinds of adaptive histogram equalization) constitutes the next task and target of this thesis.

When dealing with 3-D image enhancement procedures, some problems have to be solved. For example, it is necessary to consider the different scaling of the vertical axis, i.e., the step of a confocal microscope that took the input images. This step varies and that is why the parameters of adaptive neighbourhoods have to be adjusted accordingly. Another challenging task is an efficient software implementation of these algorithms because of their very high computational costs.

### 6.1 Basic Concepts

*Digital 3-D image*  $I$  is an  $m$ -ary valuation  $\beta$  of the digital space  $\mathfrak{D}^{(3)}$  (see Definition 4.19). Depending on  $m$ , an image can be eight-bit, sixteen-bit, grayscale, colour, etc.

*Physical voxel* (the smallest volume element)  $\mathbf{F}_{\mathbf{i}}^{(3)}$ ,  $\mathbf{i} = [i_1, i_2, i_3]$ , is a physical 3-D domain (see Definition 4.5) and an element of the physical space  $\mathfrak{F}^{(3)}$ .

*Logical voxel*  $\mathbf{L}_{\mathbf{i}}^{(3)}$ ,  $\mathbf{i} = [i_1, i_2, i_3]$ , is a logical 3-D domain (see Definition 4.7) and an element of the logical plane  $\mathfrak{L}^{(3)}$ . For each  $\mathbf{i} \in \mathbf{I}^{(3)}$ , the logical voxel  $\mathbf{L}_{\mathbf{i}}^{(3)}$  is uniquely assigned to the physical voxel  $\mathbf{F}_{\mathbf{i}}^{(3)}$  using the central mapping  $\mathbf{s}\varphi$  (see Definition 4.9).

**Remark.** If it is clear which digital space we are working with, the specification of its dimension will be omitted. If the coordinates of the physical (logical) voxel  $\mathbf{F}_{\mathbf{i}}^{(3)}$  ( $\mathbf{L}_{\mathbf{i}}^{(3)}$ ) are unknown and/or not decisive (i.e., we are only interested if a given voxel does or does not belong to a certain set), a notation  $\mathbf{F}_i$  ( $\mathbf{L}_i$ ),  $i \in \mathbb{N}_0$ , will be used.

*Image matrix* is an  $M_1$ -by- $M_2$ -by- $M_3$  matrix,  $(M_1, M_2, M_3)$  is the resolution of the digital space  $\mathfrak{D}^{(3)}$  (see Definition 4.4), whose elements are the values  $\beta(\mathbf{F}_{\mathbf{i}}^{(3)})$ ,  $\beta(\mathbf{L}_{\mathbf{i}}^{(3)})$  of individual physical (logical) voxels  $\mathbf{F}_{\mathbf{i}}^{(3)}$ ,  $\mathbf{L}_{\mathbf{i}}^{(3)}$ ,  $\mathbf{i} = [i_1, i_2, i_3]$ .

## 6.2 Adaptive Histogram Equalization in 3-D

Chapter 5 describes adaptive histogram equalization in 2-D with various kinds of adaptive neighbourhoods. Let us now extend this theory and introduce adaptive histogram equalization that uses 3-D neighbourhoods.

**Definition 6.1** (local 3-D neighbourhood). *Let  $I$  be a 3-D digital image in the spatial domain,  $\mathbf{s}\varphi : \mathfrak{F}^{(3)} \rightarrow \mathfrak{L}^{(3)}$  the central mapping,  $\mathbf{F}_0 \in I$  a processed voxel, and  $r \in \mathbb{N}$ . The set  $O^{(3)} = \left\{ \mathbf{F} \in I : \mathcal{C}_{\mathfrak{L}}^{(3)}(\mathbf{s}\varphi(\mathbf{F}), \mathbf{s}\varphi(\mathbf{F}_0)) \leq r \right\}$  is called a local 3-D neighbourhood of the voxel  $\mathbf{F}_0$  of the size (radius)  $r$ .*

**Remark.** Introduction of the 3-D neighbourhoods causes only a modification of the input data set in the histogram equalization algorithm (see Section 5.2); the principle and implementation of the algorithm itself remains unchanged.

The main difference here (compared to a 2-D neighbourhood) is a different scaling of the third axis (given by the step of a microscope that took the input data). This fact has to be considered when the distance of two logical voxels is calculated (as in Definition 6.1). The definition of the weighted maximum metric

$$\mathcal{C}_{\mathfrak{L}}^{(3)}(\mathbf{L}_{\mathbf{i}}^{(3)}, \mathbf{L}_{\mathbf{j}}^{(3)}) = \max \{c_k |i_k - j_k|\}_{k=1}^3$$

(or another metric of a logical space – see Definition 4.12) enables us to use different weights;  $c_1$  and  $c_2$  are chosen identically (usually  $c_1 = c_2 = 1$ ), while  $c_3$  is chosen according to the step of a microscope that took the original data (usually  $c_3 \in \mathbb{N}$  but any  $c_3 \in \mathbb{R}^+$  is feasible). An example of a (non-adaptive) local 3-D neighbourhood is in Figure 2.

This basic local neighbourhood is non-adaptive, i.e., it has a fixed shape given by the metric  $\mathcal{C}_{\mathfrak{L}}^{(3)}$  and the parameter  $r$ , thus not respecting any interfaces that appear in the processed image.

For this reason it is necessary to define and implement adaptive 3-D neighbourhoods (analogously to Section 5.3). These neighbourhoods have a variable shape and are able to adjust themselves to local features of the processed image.

**Remark.** While the extension into 3-D is quite straightforward in theory, the practical implementation of the algorithm is another matter; especially due to very high computational costs and usually a very high volume of data to be handled.

### 6.3 Adaptive Histogram Equalization with Adaptive Neighbourhood in 3-D

This section introduces three kinds of adaptive 3-D neighbourhoods and a fuzzy 3-D neighbourhood, as an extension of the theory built in Section 5.3.

#### 3-D $V(\alpha)$ -neighbourhood

**Definition 6.2** (3-D  $V(\alpha)$ -neighbourhood). *Let  $O^{(3)}$  be a (non-adaptive) local 3-D neighbourhood (according to Definition 6.1) of a processed voxel  $\mathbf{F}_0$ , let  $\beta : \mathfrak{D}^{(3)} \rightarrow \mathbb{N}$  be a natural valuation of the digital space  $\mathfrak{D}^{(3)}$  (according to Definition 4.19), and  $\alpha \in \mathbb{N}$ . The set  $O_V^{(3)} = \{\mathbf{F} \in O^{(3)} : |\beta(\mathbf{F}) - \beta(\mathbf{F}_0)| \leq \alpha\}$  is called a 3-D  $V(\alpha)$ -neighbourhood of  $\mathbf{F}_0$ .*

Figure 2 (right) illustrates the benefit of the 3-D  $V(30)$ -neighbourhood. As you can see, all the outlying values (compared to the processed voxel) that belong to the non-adaptive neighbourhood are correctly denied now; however, a few values that lie just outside the boundaries given by the parameter  $\alpha$  (e.g., 161 in the top right part of the middle layer) and that could be used for the processing are denied as well.

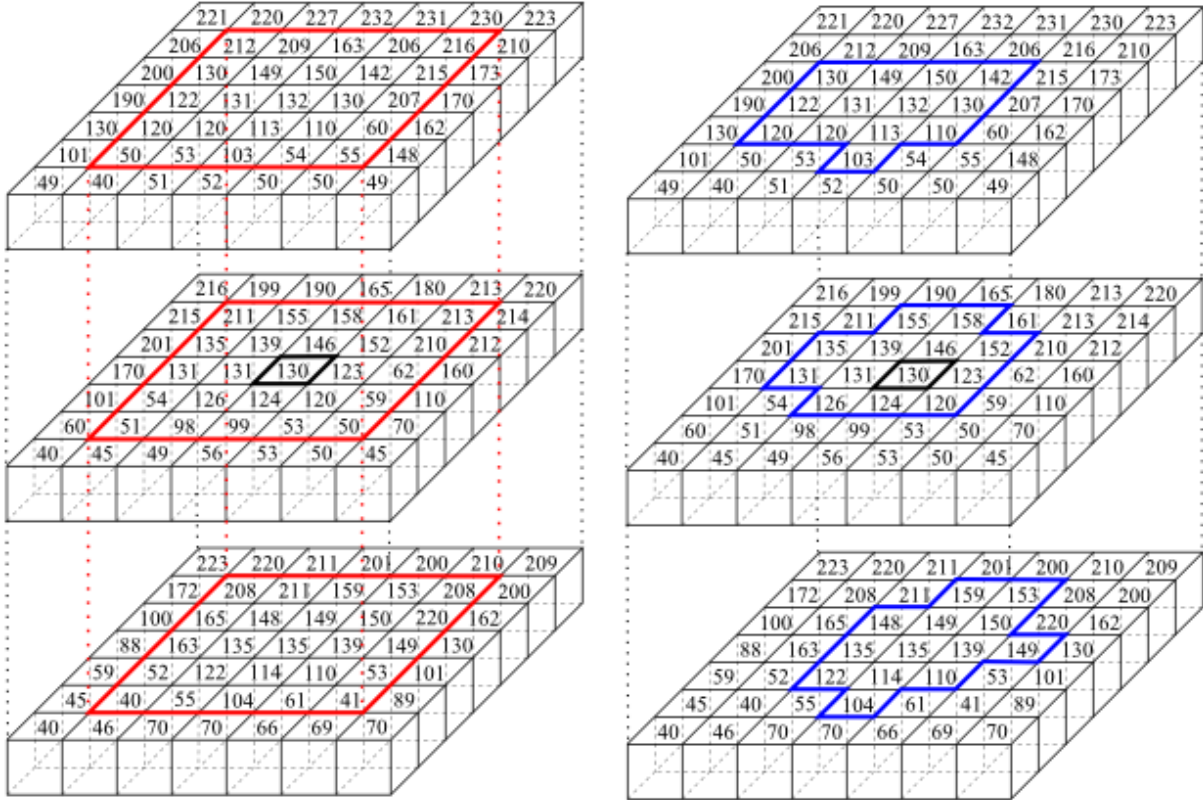


Figure 2: Non-adaptive local 3-D neighbourhood (left) and 3-D  $V(30)$ -neighbourhood (right), both with  $r=2$  and  $c_1 = c_2 = 1$ ,  $c_3 = 2$  (the gap between layers is made just for the displaying purposes)

### 3-D $A(k)$ -neighbourhood

**Definition 6.3** (3-D  $A(k)$ -neighbourhood). Let  $O^{(3)}$  be a (non-adaptive) local 3-D neighbourhood (according to Definition 6.1) with  $n$  elements of a processed voxel  $\mathbf{F}_0$ , let  $\beta : \mathfrak{D}^{(3)} \rightarrow \mathbb{N}$  be a natural valuation of the digital space  $\mathfrak{D}^{(3)}$  (according to Definition 4.19), and  $k \in \mathbb{N}$ ,  $k \leq n$ . Let us consider a sequence  $\{\mathbf{F}_i\}_{i=0}^n$  such that  $\mathbf{F}_i \in O^{(3)}$  and  $|\beta(\mathbf{F}_i) - \beta(\mathbf{F}_0)| \leq |\beta(\mathbf{F}_{i+1}) - \beta(\mathbf{F}_0)|$ ,  $i = 1, \dots, n-1$ . The set  $O_A^{(3)} = \{\mathbf{F}_i : i = 1, \dots, k\}$  is called a 3-D  $A(k)$ -neighbourhood of  $\mathbf{F}_0$ .

Figure 3 (left) illustrates the benefit of the 3-D  $A(50)$ -neighbourhood. Most of the outlying values (compared to the processed voxel) is correctly denied; however, due to a fixed number of elements (given by the parameter  $k$ ), a few of the outlying values is still included (e.g., 55 and 60 in the bottom right part of the upper layer).

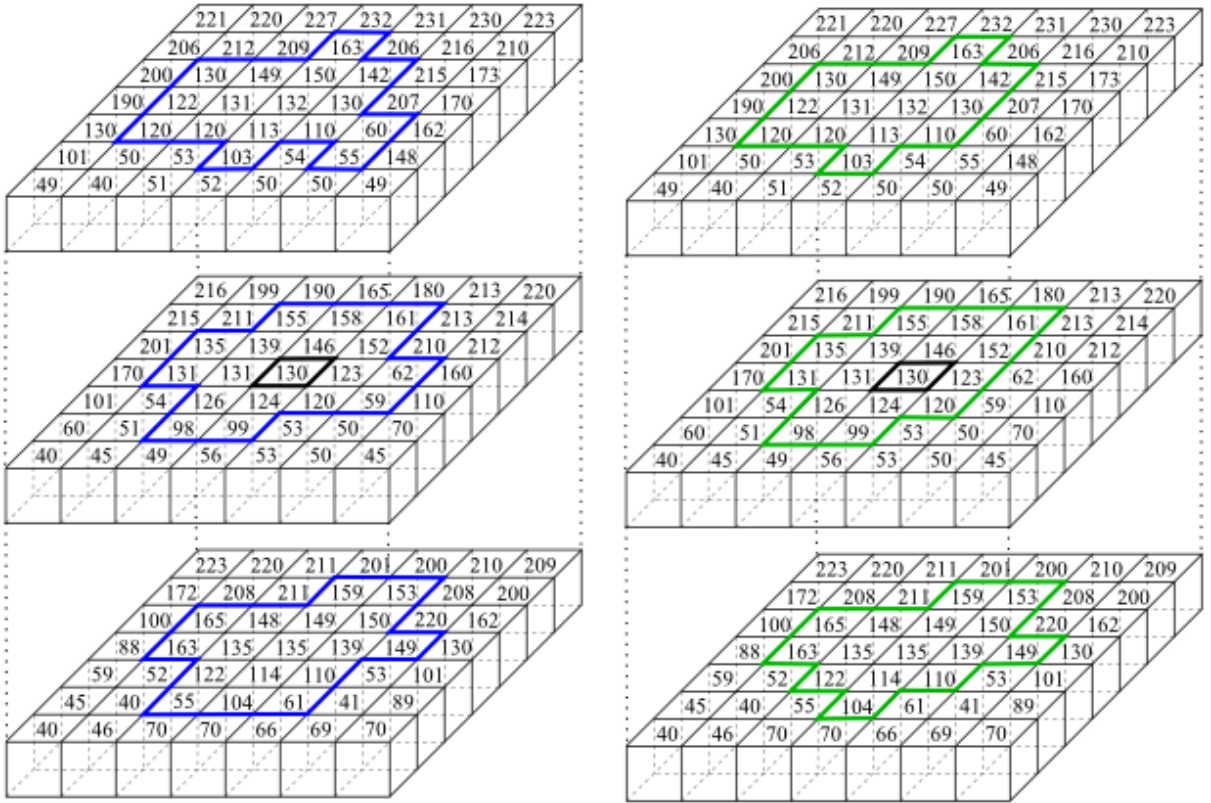


Figure 3: 3-D  $A(50)$ -neighbourhood (left) and 3-D  $S(30, 50, 1.3, 1.3)$ -neighbourhood (right), both with  $r=2$  and  $c_1 = c_2 = 1$ ,  $c_3 = 2$  (the gap between layers is made just for the displaying purposes)

### Combined 3-D $S(\alpha, k, q_V, q_A)$ -neighbourhood

**Definition 6.4** (combined 3-D  $S(\alpha, k, q_V, q_A)$ -neighbourhood). Let  $O^{(3)}$  be a (non-adaptive) local 3-D neighbourhood (according to Definition 6.1) with  $n$  elements of a processed voxel

$\mathbf{F}_0$ , let  $\beta : \mathfrak{D}^{(3)} \rightarrow \mathbb{N}$  be a natural valuation of the digital space  $\mathfrak{D}^{(3)}$  (according to Definition 4.19),  $\alpha, k \in \mathbb{N}$ ,  $k \leq n$ , and  $q_V, q_A \in \mathbb{R}$ ,  $q_V, q_A \geq 1$ . Let us consider a sequence  $\{\mathbf{F}_i\}_{i=0}^n$  such that  $\mathbf{F}_i \in O^{(3)}$  and  $|\beta(\mathbf{F}_i) - \beta(\mathbf{F}_0)| \leq |\beta(\mathbf{F}_{i+1}) - \beta(\mathbf{F}_0)|$ ,  $i = 1, \dots, n-1$ . Let  $O_V^{(3)}$  be the 3-D  $V(\alpha)$ -neighbourhood (according to Definition 6.2) of  $\mathbf{F}_0$ , and  $O_A^{(3)}$  the 3-D  $A(k)$ -neighbourhood (according to Definition 6.3) of  $\mathbf{F}_0$ . The set  $O_S^{(3)} = \left(O_V^{(3)} \cap O_A^{(3)}\right) \cup O'_S$  where

$$O'_S = \begin{cases} \left\{ \mathbf{F}_i \in O_A^{(3)} \setminus O_V^{(3)} : |\beta(\mathbf{F}_i) - \beta(\mathbf{F}_0)| \leq q_V \cdot \alpha \right\} & \text{for } O_V^{(3)} \subset O_A^{(3)}, \\ \left\{ \mathbf{F}_i \in O_V^{(3)} \setminus O_A^{(3)} : i = k+1, \dots, \min\{\lfloor q_A \cdot k \rfloor, |O_V^{(3)}|\} \right\} & \text{for } O_A^{(3)} \subset O_V^{(3)}, \\ \emptyset & \text{for } O_V^{(3)} = O_A^{(3)}, \end{cases}$$

is called a 3-D  $S(\alpha, k, q_V, q_A)$ -neighbourhood of  $\mathbf{F}_0$ . If  $O_S^{(3)} = \emptyset$ , the equalization is not performed and the processed voxel  $\mathbf{F}_0$  keeps its original value.

The contribution of this combined neighbourhood is demonstrated in Figure 3 (right); the imperfections described in the two previous examples are now rectified, given by the properties of the combined neighbourhood.

The practical application of these methods to HDR images is shown in Section 6.4.

### 3-D Fuzzy neighbourhood

When constructing the adaptive 3-D neighbourhoods discussed so far, we are only interested if a given voxel does or does not belong to the neighbourhood. However, we can also take into consideration the actual distance of the given voxel from the processed one, and, according to this distance, assign the membership degree to each voxel. The main point is that the smaller the distance, the bigger reliability and relevance of the corresponding voxel.

**Definition 6.5** (fuzzy 3-D local neighbourhood). *Let  $I$  be a 3-D digital image in the spatial domain,  $\mathbf{s}\varphi : \mathfrak{F}^{(3)} \rightarrow \mathfrak{L}^{(3)}$  the central mapping,  $\mathbf{F}_0 \in I$  a processed voxel,  $r \in \mathbb{N}$ , and  $\mu_r \in \mathbb{R}$ ,  $0 < \mu_r \leq 1$ . Let us consider a function  $\mu : \mathfrak{D}^{(3)} \rightarrow \mathbb{R}^+$  defined as*

$$\mu(\mathbf{F}) = 1 - \frac{1 - \mu_r}{r} \cdot \mathcal{C}_{\mathfrak{L}}^{(3)}(\mathbf{s}\varphi(\mathbf{F}), \mathbf{s}\varphi(\mathbf{F}_0)).$$

*Then the set  $\tilde{O}^{(3)} = \left\{ [\mathbf{F}, \mu(\mathbf{F})] : \mathbf{F} \in I, \mathcal{C}_{\mathfrak{L}}^{(3)}(\mathbf{s}\varphi(\mathbf{F}), \mathbf{s}\varphi(\mathbf{F}_0)) \leq r \right\}$  is called a fuzzy 3-D local neighbourhood of  $\mathbf{F}_0$ .*

Again, the main difference here, compared to the 2-D case, is a different scaling of the third axis, i.e., the decrease of the membership degree is different in this direction (usually faster).

Analogously to Definition 6.5, 3-D  $V(\alpha)$ , 3-D  $A(k)$  and 3-D  $S(\alpha, k, q_V, q_A)$ -fuzzy neighbourhoods can be defined as well.

## 6.4 Comparison of Results

The practical implementation and comparison of the methods described in Sections 5.3 and 6.3 is illustrated in Figures 4– 6. The precise specification is given underneath. All computations were run on a PC with AMD Athlon 64 3500+ (2.2 GHz), RAM 1 GB.

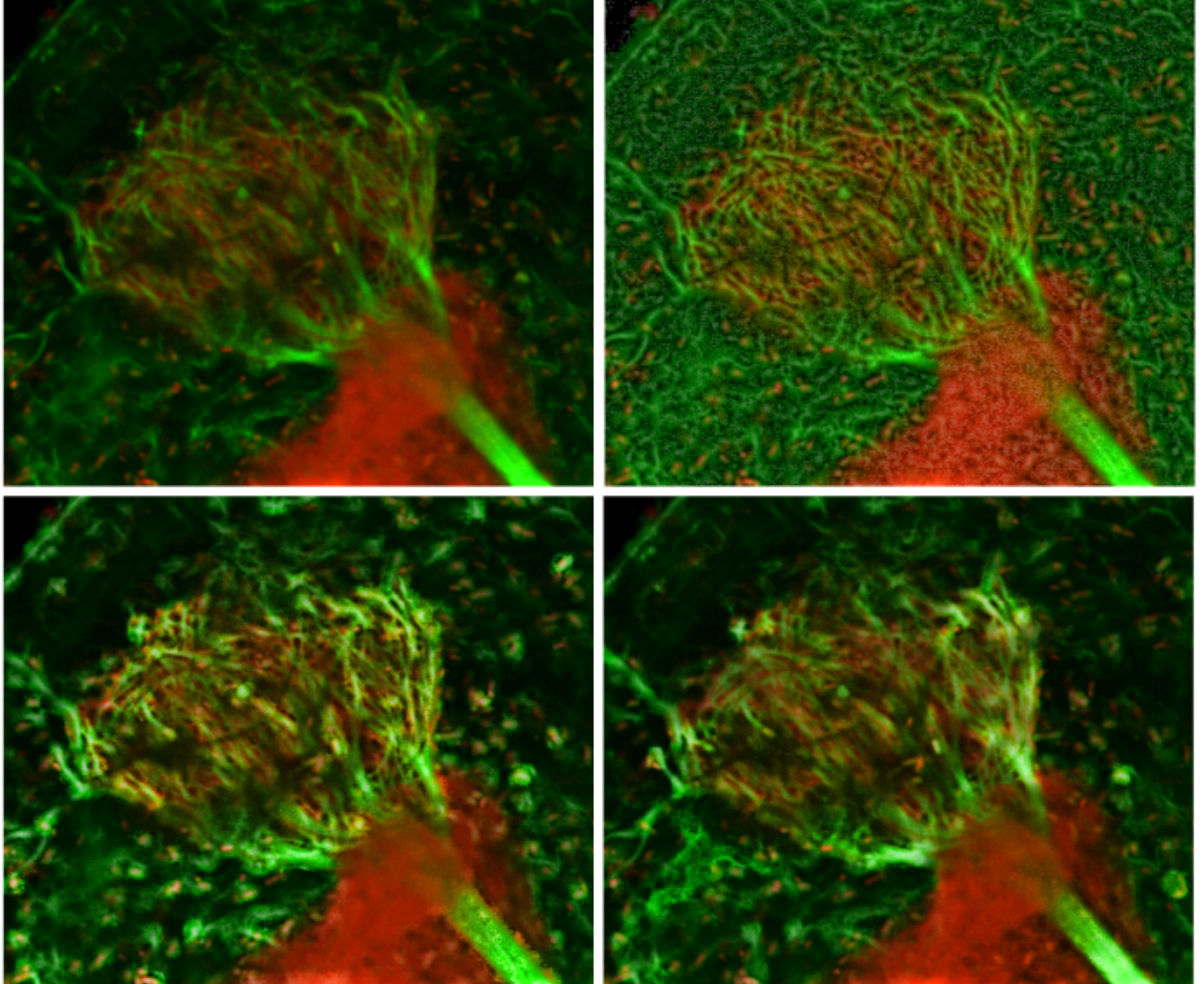
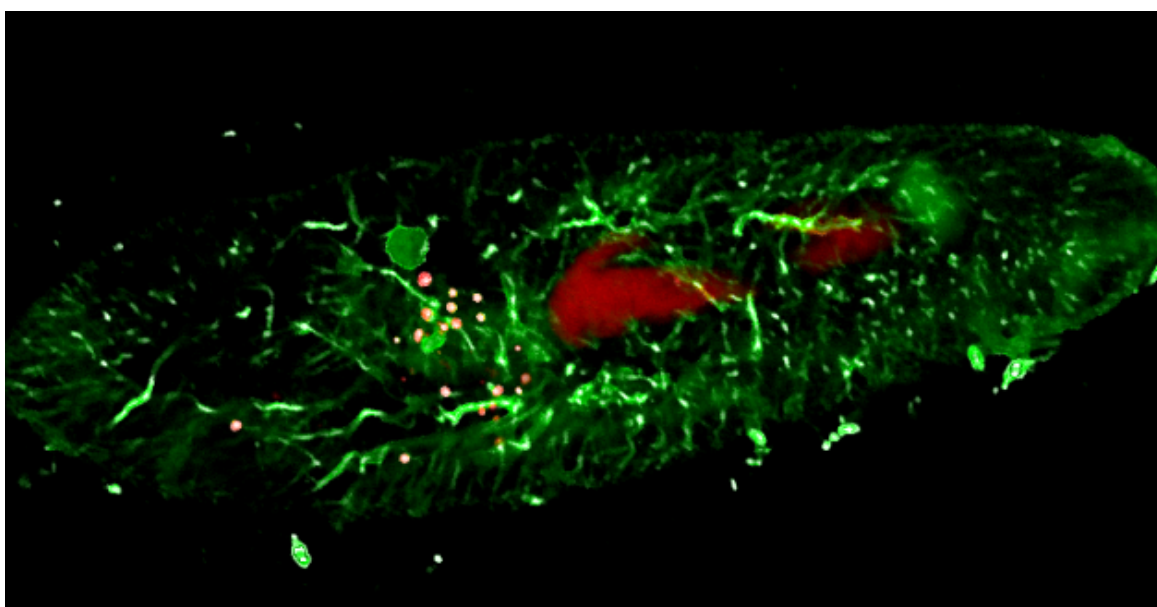
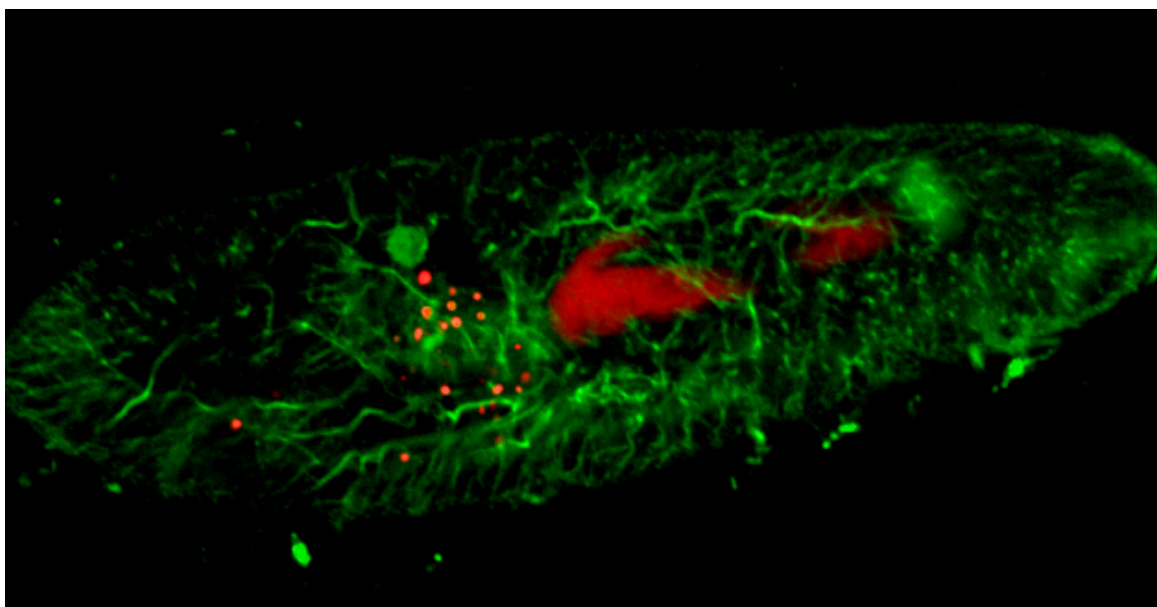
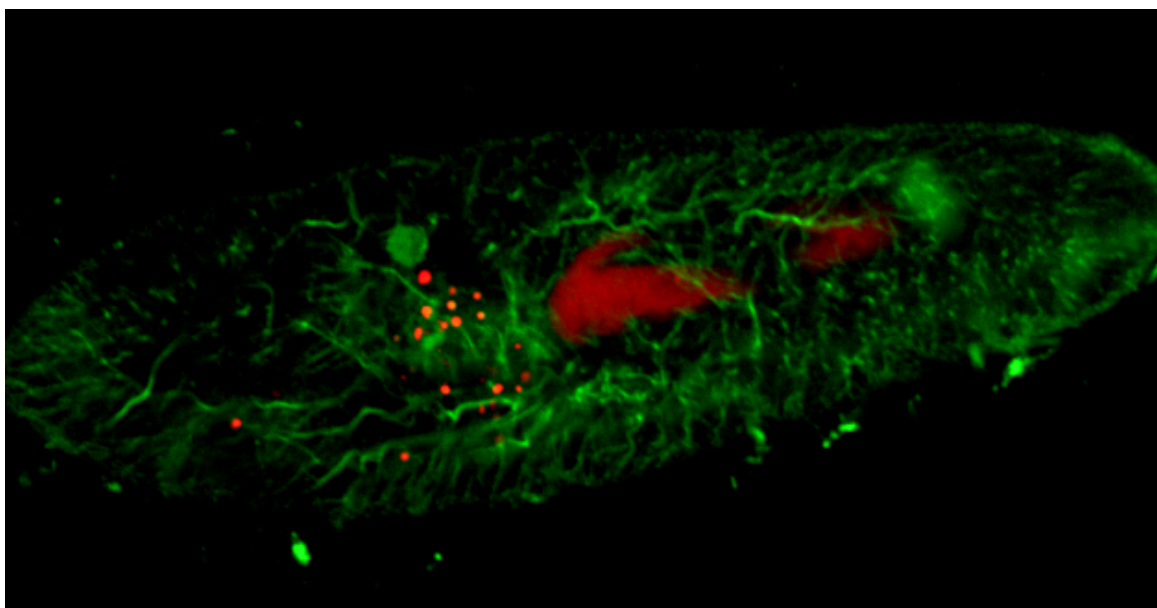


Figure 4: Original TIF file „0006jan.tif“ contains  $2 \times 8$  images, resolution  $425 \times 352$  px. (DVD: 3-D Equalization\01); all the cuts were processed, the 8<sup>th</sup> one is shown here. The original image (top left); equalized image by means of a 2-D (non-adaptive) local neighbourhood with  $r = 4$  (top right), computational time 38 sec. (4.8 sec. per image); equalized image by means of a 2-D (non-adaptive) local neighbourhood with  $r = 6$ , adaptive to additive noise (bottom left), computational time 39 sec. (4.9 sec. per image); and equalized image by means of a 3-D (non-adaptive) local neighbourhood with  $r = 6$ , adaptive to additive noise (bottom right), computational time 50 sec.

From this example we can see that the adaptivity to additive noise is a must. The contribution of 3-D processing is also illustrated here – a lot of undesirable artifacts is removed, in exchange for a modest increase of the computational time.



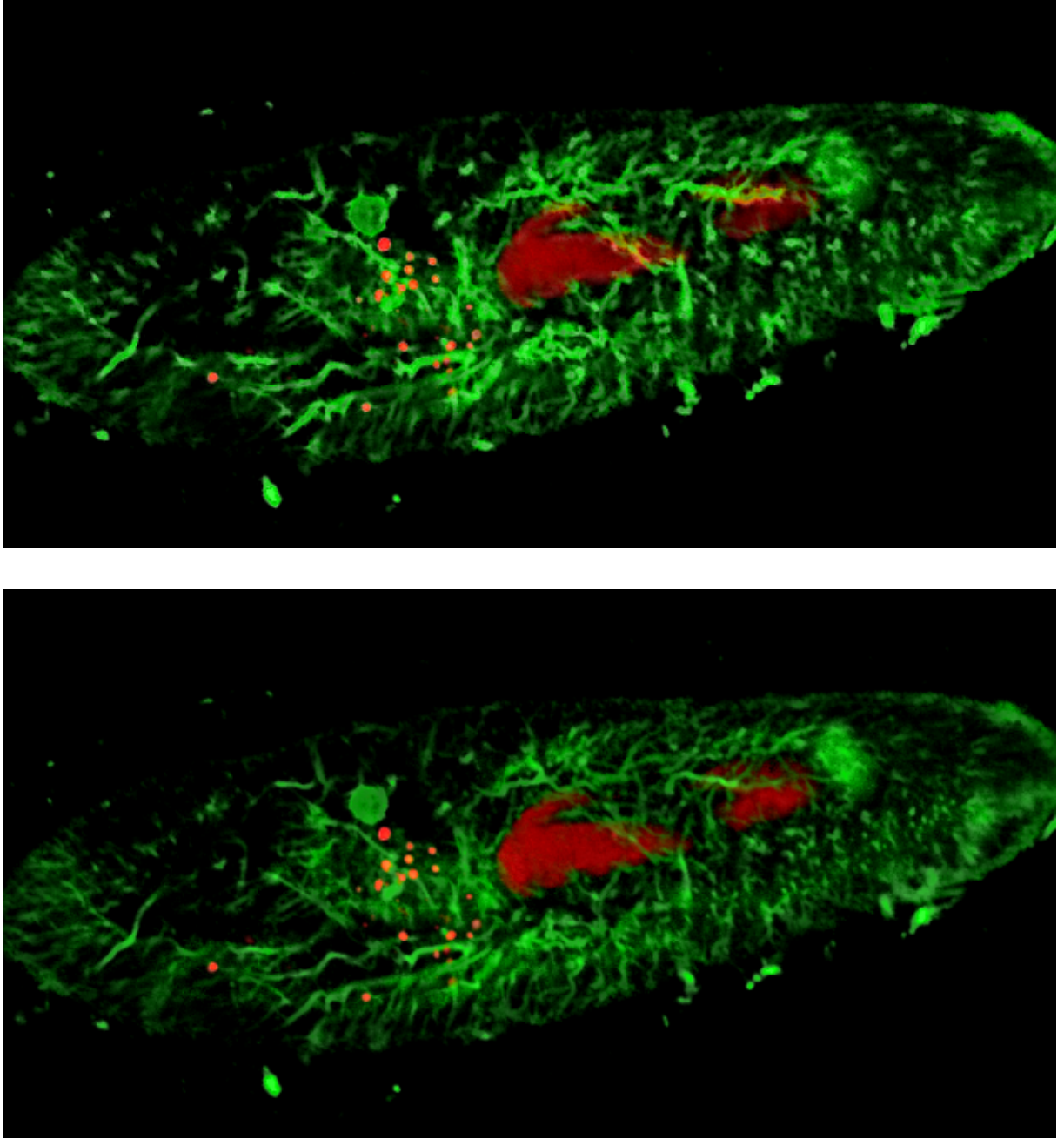


Figure 5: Original TIF file „Pc201.tif“ contains  $2 \times 24$  images, resolution  $784 \times 407$  px. (DVD: 3-D Equalization\02); all the cuts were processed, the 22<sup>nd</sup> one is shown here. The original image (previous page, top); image with expanded contrast (previous page, middle), computational time 2 sec. (0.1 sec. per image); equalized image by means of a 2-D  $A(k)$ -neighbourhood with  $r = 6$ ,  $k = 50$ , adaptive to additive noise (previous page, bottom), computational time 2 min. 55 sec. (7.3 sec. per image); equalized image by means of a 2-D  $V(\alpha)$ -neighbourhood with  $r = 6$ ,  $\alpha = 30 \cdot 256$ , adaptive to additive noise (this page, top), computational time 6 min. 35 sec. (15.2 sec. per image); and equalized image by means of a 3-D  $V(\alpha)$ -neighbourhood with  $r = 6$ ,  $\alpha = 30 \cdot 256$ , adaptive to additive noise (this page, bottom), computational time 8 min. 3 sec.

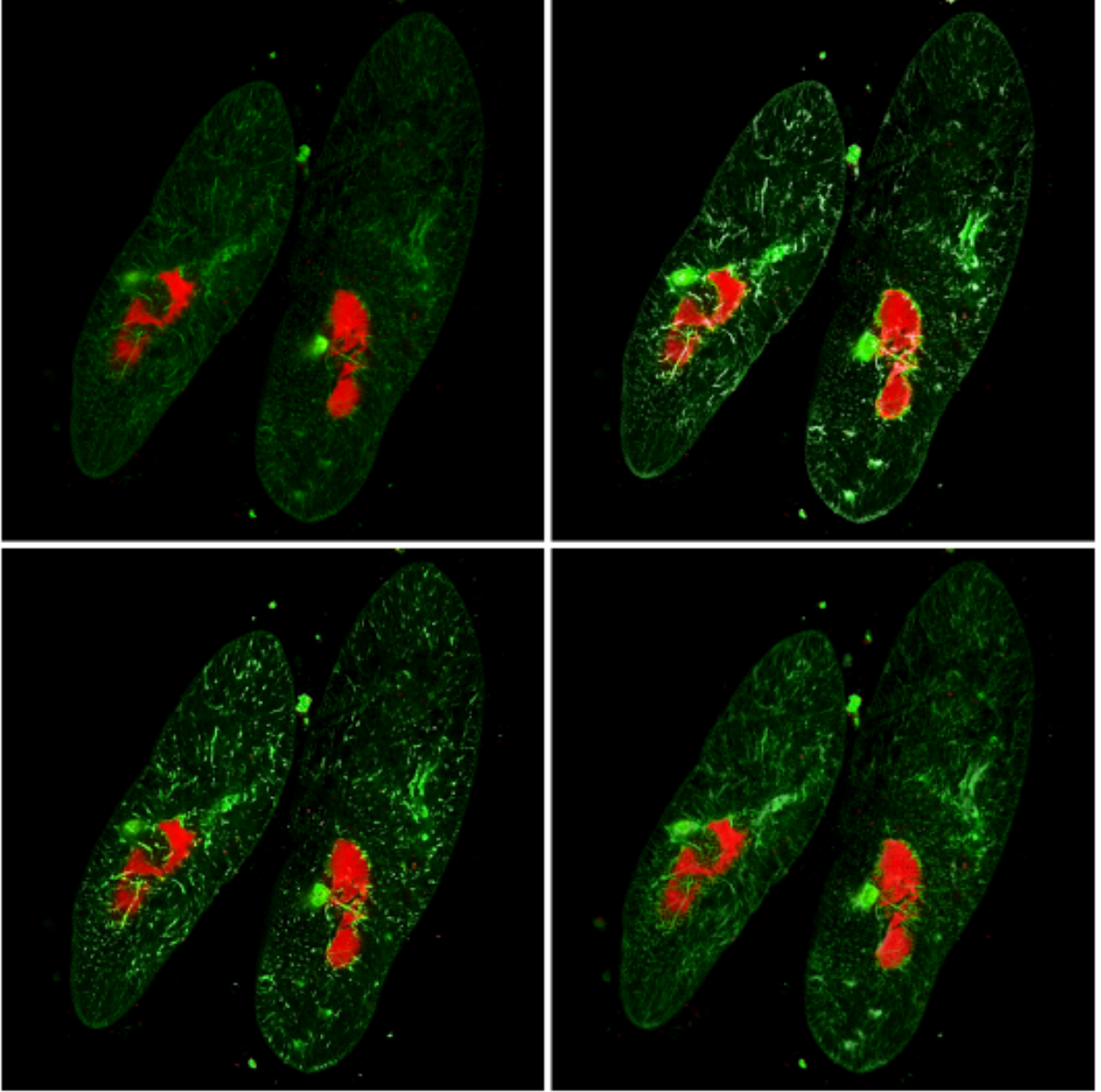


Figure 6: Original TIF file „BParam6.tif“ contains  $2 \times 13$  images, resolution  $1024 \times 1024$  px. (DVD: 3-D Equalization\03); all the cuts were processed, the 11<sup>th</sup> one is shown here. The original image (top left); equalized image by means of a 2-D  $S(\alpha, k, q_V, q_A)$ -neighbourhood with  $r = 14$ ,  $\alpha = 35 \cdot 256$ ,  $k = 50$ ,  $q_V = q_A = 1.4$ , adaptive to additive noise (top right), computational time 7 min. 8 sec. (32.9 sec. per image); equalized image by means of a 2-D  $S(\alpha, k, q_V, q_A)$ -neighbourhood with  $r = 4$ ,  $\alpha = 35 \cdot 256$ ,  $k = 50$ ,  $q_V = q_A = 1.4$ , adaptive to additive noise (bottom left), computational time 4 min. 42 sec. (21.7 sec. per image); and equalized image by means of a 3-D  $S(\alpha, k, q_V, q_A)$ -neighbourhood with  $r = 4$ ,  $\alpha = 35 \cdot 256$ ,  $k = 50$ ,  $q_V = q_A = 1.4$ , adaptive to additive noise (bottom right), computational time 6 min. 32 sec.

## 7 Three-Dimensional Object Reconstruction

Three-dimensional reconstruction of an explored object from a series of its optical cuts was introduced in author's diploma thesis [10] and published in author's publication [9].

In this chapter, let us take a brief look at the basic ideas because this algorithm is also included in the software developed as a part of this thesis. Moreover, new HDR data are used now and the new software enables to perform various kinds of 2-D and 3-D image enhancement procedures, according to the previous chapters, thus yielding better resulting reconstructions.

### Principle of 3-D Object Reconstruction

A series of optical cuts can be represented as a three-dimensional data grid. Let us choose the Cartesian coordinate system  $\langle O, \mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3 \rangle$  in the following way: the point  $O$  is identified with the centre of the last cut. The vectors  $\mathbf{e}_1, \mathbf{e}_2$  are the direction vectors of the axes of symmetry of the optical cuts and their size is equal to the dimensions of the physical pixels of the input data, which are assumed to be unit. The vector  $\mathbf{e}_3$  is orthogonal to  $\mathbf{e}_1, \mathbf{e}_2$  and is unit as well (see Figure 7). However, the distance of the optical cuts is generally not unit. This distance is set by the user according to the step of the microscope that provided the input data.

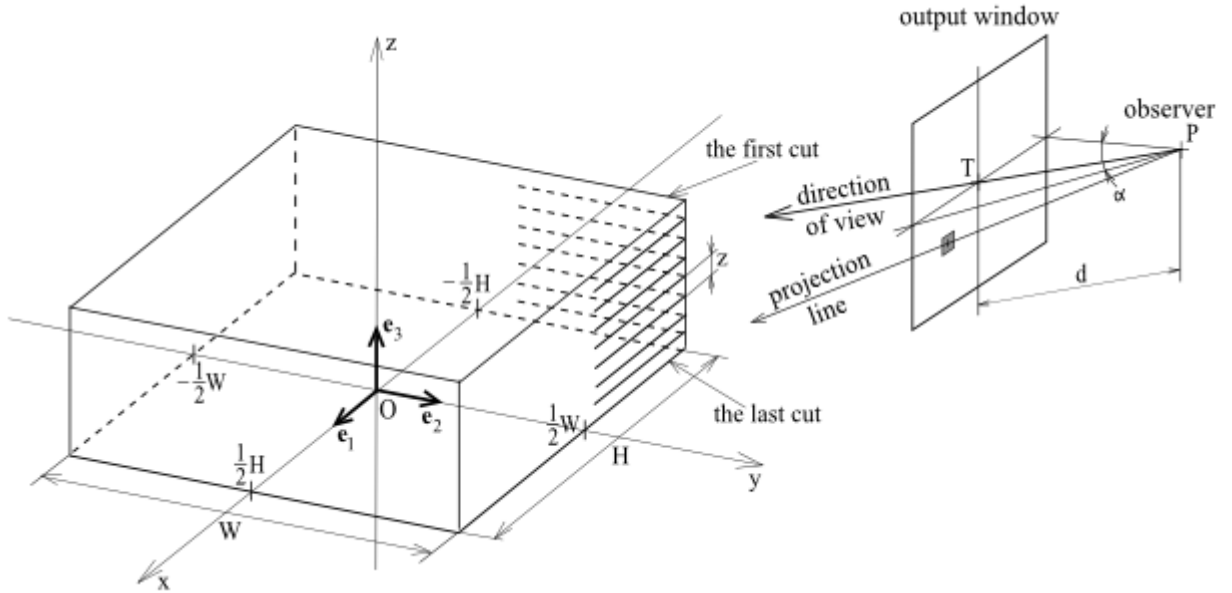


Figure 7: Principle of three-dimensional object reconstruction

Now it is necessary to place a screen (output window) on which the reconstructed object will be displayed. The position of the observer is given by the point  $P$  and the central

projection with the center at  $P$  is used. The vector  $\mathbf{d}$  of the main ray (direction of view) is set by the user. Hereby, the coordinates of the vector  $\mathbf{d}$  in the base  $\{\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3\}$  are given.

The plane of the screen is given by the point  $T$  and two direction vectors  $\mathbf{r}, \mathbf{s}$ . The point  $T$  is determined as  $T = P + \mathbf{d}$ . The vector  $\mathbf{r}$  is orthogonal to the vector  $\mathbf{d}$  and coplanar with the vectors  $\mathbf{e}_1, \mathbf{e}_2$ . Its size is determined by the vector  $\mathbf{d}$  and the angle of view  $\alpha$ , which is set by the user. The vector  $\mathbf{s}$  is orthogonal to the vectors  $\mathbf{d}, \mathbf{r}$  and the ratio  $\frac{\|\mathbf{s}\|}{\|\mathbf{r}\|}$  is equal to the height-to-width ratio of the output window; see Figure 8.

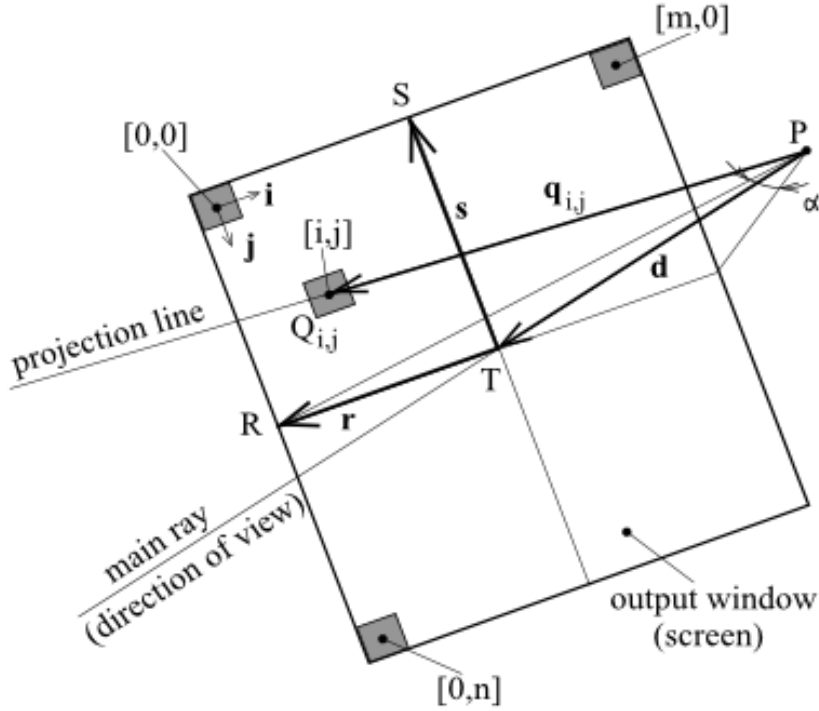


Figure 8: Output window (screen) serving for displaying of a reconstructed object

The base vectors  $\mathbf{i}, \mathbf{j}$  of the global coordinate system satisfy

$$\mathbf{i} = -\frac{2\mathbf{r}}{M} \quad \text{and} \quad \mathbf{j} = -\frac{2\mathbf{s}}{N}$$

where  $(M, N)$  is the resolution of the output window. Thus, the logical pixel of the output window with global coordinates  $[i, j]$  is also a point  $Q_{i,j}$  satisfying

$$Q_{i,j} - P = \mathbf{q}_{i,j} = \mathbf{d} + \mathbf{r} + \mathbf{s} + i \cdot \mathbf{i} + j \cdot \mathbf{j}.$$

From the programmer's point of view, the output window is an image created in the following way: a projection line

$$X = P + t \cdot \mathbf{q}_{i,j}$$

passes step by step through each logical pixel  $[i, j]$  of the output window. In the software solution, this procedure is realized in the following way:

$$\begin{aligned} \mathbf{q}_{0,0} &= \mathbf{d} + \mathbf{r} + \mathbf{s} \\ \mathbf{q}_{i+1,j} &= \mathbf{q}_{i,j} + \mathbf{i} \quad (\text{movement inside a row of the output window}) \\ \mathbf{q}_{0,j+1} &= \mathbf{q}_{0,j} + \mathbf{j} \quad (\text{movement to the next row in the output window}). \end{aligned}$$

Now we have to find the intersections of the projection line with the system of cuts and to save the values of individual colour components of these intersections. Then, all the colours acquired are mixed (see [10]) and the resulting colour of the currently processed pixel is obtained.

Quick and efficient computation of the intersections of the projection line with the system of cuts constitutes the biggest challenge. For this purpose, the so-called line segment voxelization procedure (using the Bresenham's algorithm and fast integer arithmetic only; see [10] and [9]) was developed.

### Line Segment Voxelization

Using the coordinates of the input and output voxel, we find out on which axis the biggest difference between the original and the terminal coordinates appears. Let, for example, the axis  $x$  be this axis (see Figure 9).

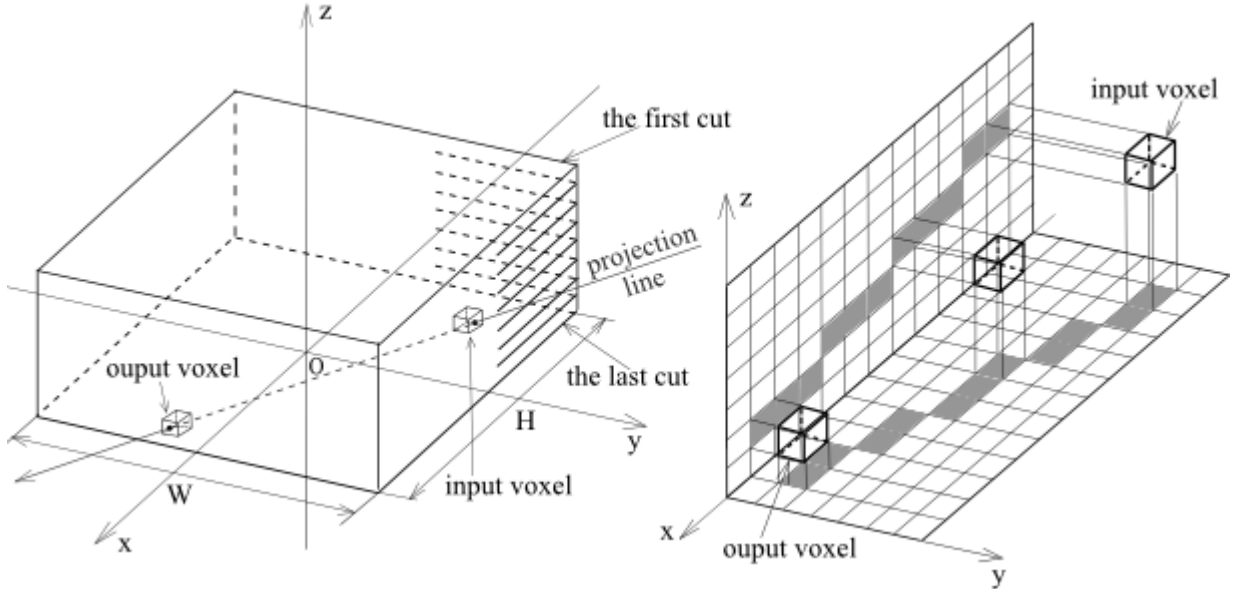


Figure 9: Principle of line segment voxelization

Now let us project the input and output voxel into the plane  $xy$  and let us use the Bresenham's algorithm for the first time. It will proceed along the axis  $x$  and generate the

$y$ -coordinates of the pixels approaching the line segment connecting the original and terminal pixel. Now, by analogy, let us project the input and output voxel into the plane  $xz$  and let us use the Bresenham's algorithm for the second time. In the final phase, these partial results are combined and we obtain the logical coordinates of individual voxels. This algorithm works with integer values only, which makes it very fast.

Two examples of reconstructed objects are given in Figures 10 and 11.

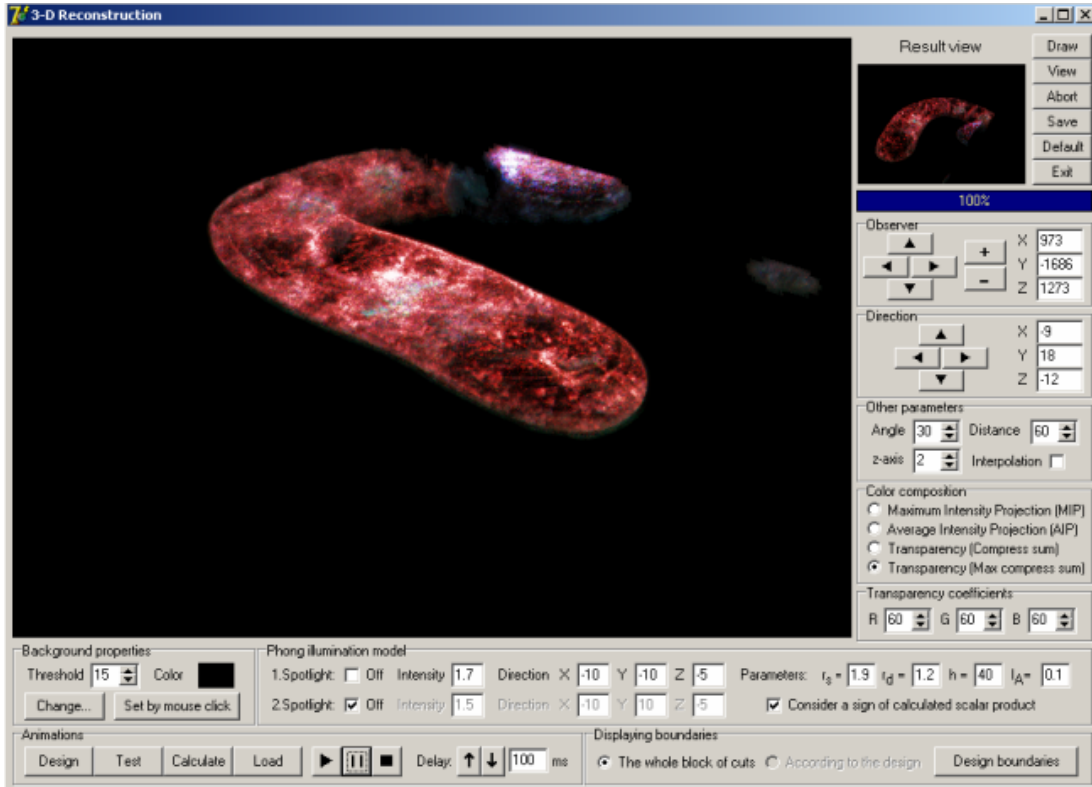


Figure 10: 3-D reconstruction of Tobacco cell; beginning of strangulation

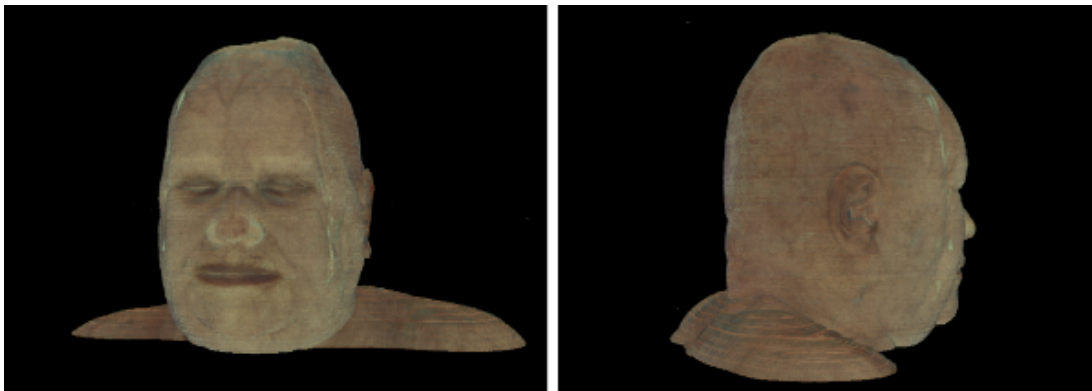


Figure 11: Visible Human Project: 3-D reconstruction of a human head

## 8 Applications

The broad range of applications available to laser scanning confocal microscopy includes a wide variety of studies in neuroanatomy and neurophysiology, as well as morphological studies of a wide spectrum of cells and tissues. In addition, the growing use of new fluorescent proteins is rapidly expanding the number of original research reports coupling these useful tools to modern microscopic investigations. Other applications include resonance energy transfer, stem cell research, photobleaching studies, lifetime imaging, multiphoton microscopy, total internal reflection, DNA hybridization, membrane and ion probes, bioluminescent proteins, and epitope tagging. Some of these techniques are described in the paragraphs below.

Another application that often works with HDR images is displaying of the solar corona. Due to an enormous contrast between the brightest part of the corona and its parts lying further away, it is necessary to use mathematical algorithms to visualize coronal structures.

Figure 12 shows an example of visualization of a solar corona image. The original image (left) was equalized by means of 2-D adaptive histogram equalization with adaptive neighbourhood (right); as you can see, a lot of coronal structures, almost invisible in the original image, is clearly visible in the resulting image.

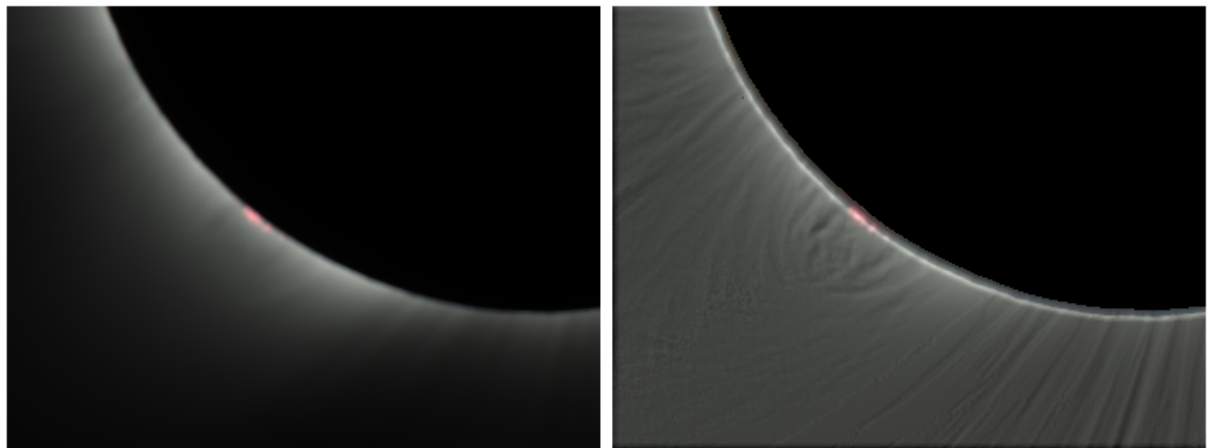


Figure 12: Original image (left) and equalized image (right) by means of a 2-D (non-adaptive) local neighbourhood with  $r = 4$ , computational time 3 sec.

## 9 Conclusions

The thesis has been concerned with 2-D and 3-D adaptive filters for visualization of high-dynamic range (HDR) images. The first part of the full thesis contains an introduction of HDR images, a description of the principle of confocal microscopy and its comparison with classic optical microscopy, a discussion about the point spread function, and a mathematically correct definition of digital image and other related terms (physical and logical domains, vertex and central mapping, digital space metrics, valuation, etc.) that provide the necessary theoretical background.

Author's main results are contained in the second part of the thesis that explains the frequency domain approach (using discrete Fourier transform and frequency filters) and digital geometry approach (using adaptive histogram equalization) to 2-D and 3-D image enhancement, and the principle of the 3-D object reconstruction.

The contribution of this thesis consists in the following points:

- Formalization of the known theory of adaptive histogram equalization; the algorithm of histogram equalization itself is described, and the definitions of (non-adaptive) local neighbourhoods, adaptive  $V(\alpha)$ -neighbourhood and adaptive  $A(k)$ -neighbourhood are given, based on the theory built in the first part of the thesis.
- Introduction of the combined  $S(\alpha, k, q_V, q_A)$ -neighbourhood; this adaptive neighbourhood connects advantages of simpler methods in order to obtain a more broadly-usable neighbourhood.
- Introduction of the fuzzy neighbourhood; this approach takes into account the distance of individual logical pixels/voxels from the processed one and assigns their membership degree accordingly.
- Generalization of the previous theory into 3-D; with suitable data, e.g., with a series of optical cuts that can be represented as a 3-D digital image, it is possible to employ adaptive histogram equalization in 3-D in order to obtain better-looking outputs.
- Illustration of the benefits of the 3-D processing; by means of new HDR data provided by the Olympus FluoView<sup>TM</sup> FV1000 Laser Scanning Confocal Microscope, several examples are given in order to prove the contribution of the 3-D adaptive histogram equalization with adaptive neighbourhood.
- Introduction and description of 3-D frequency filters and 3-D processing in the frequency domain.

All the results discussed above are also implemented in the software developed as a part of this thesis. The software employs opening of images of various formats, 2-D and 3-D adaptive histogram equalization with adaptive neighbourhood and adaptive to additive noise, 2-D composition of optical cuts, 3-D object reconstruction, linear and non-linear filters, 2-D discrete Fourier transform and inverse discrete Fourier transform, frequency filters, etc.

Since this thesis targets primarily the visualization of HDR images provided by means of confocal microscopes, the results can be applied especially in biological and medical research. The broad range of applications includes studies of the complicated architecture of neural networks, morphological studies of a wide spectrum of cells and tissues, studies in neuroanatomy and neurophysiology, CT scans, stem cell research, resonance energy transfer, etc. However, any application that needs to visualize HDR images can utilize the results of the thesis.

## References

- [1] BEZVODA, V., et al. *Dvojměrná diskrétní Fourierova transformace a její použití – I.: Teorie a obecné užití*. 1st edition. Praha: Státní pedagogické nakladatelství, 1988.
- [2] CAPODIFERRO, L., et al. *Two-channel technique for high dynamic range image visualization*. 8<sup>th</sup> International Conference on Information Visualization, London, 2004, pp. 269-273.
- [3] ČÍŽEK, V. *Diskrétní Fourierova transformace a její použití*. 1st edition. Praha: SNTL – Nakladatelství technické literatury, 1981.
- [4] DRUCKMÜLLER, M. – HERIBAN, P. *Digital Image Processing System for Windows*, version 5.0. SOFO, 1996.
- [5] DRUCKMÜLLER, M. *Phase Correlation Method for the Alignment of Total Solar Eclipse Images*. The Astrophysical Journal, 2009, vol. 706, no. 2, pp. 1605–1608.
- [6] MARTIŠEK, D. *Počítačová grafika pro matematické inženýry*. PC-Dir Real, s.r.o., Brno, 1999.
- [7] MARTIŠEK, D. *Matematické principy grafických systémů*. Littera, Brno, 2002.
- [8] MARTIŠEK, D. *The 2-D Processing of Images Provided by Confocal Microscopes*. Mendel, Brno, 2002.
- [9] MARTIŠEK, D. – MARTIŠEK, K.: *Direct Volume Rendering Methods for Cell Structures*. Scanning: The Journal of Scanning Microscopies, accepted on 1<sup>st</sup> February 2012.

- [10] MARTIŠEK, K. *Numerical Methods of Multispectral Confocal Microscopy*. Diploma thesis. Brno University of Technology, 2007.
- [11] PRATT, W. K. *Digital Image Processing*. 4th edition. John Wiley & Sons, Inc. Hoboken, New Jersey., 2007.
- [12] SRINIVASA REDDY, B. – CHATTERJI, B. N. *An FFT-Based Technique for Translation, Rotation, and Scale-Invariant Image Registration*. IEEE Transactions on Image Processing, August 1996, vol. 5, no. 8, pp. 1266-1271.
- [13] SRIPATHI, D. *Efficient Implementations of Discrete Wavelet Transforms Using FPGAs*. Diploma thesis. The Florida State University, 2003.
- [14] STARCK, J. – MURTAGH, F. – BIAOUI, A. *Image Processing and Data Analysis*. Cambridge University Press, 1998.
- [15] UHLÍŘ, V., et al. *Current-induced motion and pinning of domain walls in spin-valve nanowires studied by XMCD-PEEM*. Physical Review B, 22., Vol.81, 2010.
- [16] ŽENÍŠEK, A. *Lebesgueův integrál a základy funkcionální analýzy*. PC-Dir Real, s.r.o., Brno, 1999.
- [17] Image Processing Learning Resources [online] [cit. 4 December 2011].  
<<http://homepages.inf.ed.ac.uk/rbf/HIPR2>>.
- [18] Olympus FluoView Resource Center [online] [cit. 28 November 2011].  
<<http://www.olympusfluoview.com>>.
- [19] P3DFFT – Highly Scalable Parallel 3D Fast Fourier Transforms Library [online] [cit. 27 March 2012].  
<<http://www.sdsc.edu/us/resources/p3dfft>>.
- [20] The Visible Human Project® [online] [cit. 11 December 2011].  
<[http://www.nlm.nih.gov/research/visible/visible\\_human.html](http://www.nlm.nih.gov/research/visible/visible_human.html)>.

## Author's Publications

1. MARTIŠEK, D. – MARTIŠEK, K. – PROCHÁZKOVÁ, J. *New Methods for Space Reconstruction of Inside Cell Structures*. Journal of Applied Biomedecine, Vol.2007, (2007), No.5, pp.151-155.
2. KARPÍŠEK, Z. – MARTIŠEK, K. *Fuzzy Reliability of a System*. Proceedings East West Fuzzy Colloquium 2010, pp.147-153.
3. KARPÍŠEK, Z. – MARTIŠEK, K. *Software pro výpočet fuzzy spolehlivosti soustav s využitím FJK-algoritmu*. Informační bulletin České statistické společnosti, Vol.22, (2011), No.2, pp.71-79.
4. MARTIŠEK, K. – DRUCKMÜLLEROVÁ, H. *A Numerical Method for the Visualization of the Fe XIV Emission in the Solar Corona Using Broadband Filters*. The Astrophysical Journal Supplement Series, Vol.197, (2011), No.2, pp.23-29.
5. MARTIŠEK, D. – MARTIŠEK, K.: *Direct Volume Rendering Methods for Cell Structures*. Scanning: The Journal of Scanning Microscopies, accepted on 1<sup>st</sup> February 2012.

## Other Author's Products

1. ŠENBERGER, J. – ČECH, J. – ZÁDĚRA, A. – KAŇA, V. – KARPÍŠEK, Z. – MARTIŠEK, K. Výpočet licí teploty litin Ni-resist (software).
2. MARTIŠEK, K. – KARPÍŠEK, Z. Fuzzy spolehlivost systému (software).
3. ZÁDĚRA, A. – LAŠTOVICA, J. – ŠENBERGER, J. – MARTIŠEK, K. Termodynamické výpočty aktivity kyslíku a síry v litinách (software).
4. MARTIŠEK, K. – NOVOTNÝ, J. – NOVOTNÝ, J. Software for 2-D and 3-D Digital Images Processing.

## Author's CV

**Name:** Ing. Karel Martišek

**Born:** 2nd February 1984, Brno

**Nationality:** Czech

**Address:** Vlčnovská 3, Brno, 628 00

### **Education:**

**from 2007** doctoral study - Brno University of Technology, Faculty of Mechanical Engineering, study branch: Applied Mathematics

**2002 – 2007** master study - Brno University of Technology, Faculty of Mechanical Engineering, study branch: Mathematical Engineering

### **Pedagogic activities:**

**2007 – 2011** tutorials from Mathematics I, II, III and IV in FME BUT

### **Projects:**

**2010** grant project FSI-J-10-76 of the Specific Research 2010 „Statistical and Optimization Methods in Image Analysis“

**2009** grant project 2E08017 of the Ministry of Education, Youth and Sports of the Czech Republic „Numerical Methods of Visualization of Fe XIV Emission in Solar Corona“

participation on the grant project FSI-S-10-14 of the Specific Research 2011 „Modern Methods of Mathematical Simulation of Engineer Sciences Problems“

## Abstract

The thesis is concerned with filters for visualization of high dynamic range images. In the theoretical part of the full thesis, the principle of confocal microscopy is described and the term digital image is defined in a mathematically correct way. Both frequency approach (using 2-D and 3-D discrete Fourier transform and frequency filters) and digital geometry approach (using adaptive histogram equalization with adaptive neighbourhood) are chosen for the processing of images. Necessary adjustments when working with non-ideal images containing additive and impulse noise are described as well. The last part of the thesis is interested in 3-D reconstruction from optical cuts of an object. All the procedures and algorithms are also implemented in the software developed as a part of this thesis.