Image reconstruction from multiscale top points

Kanters, F.M.W.

Award date:
2002

TU/e Eindhoven University of Technology

Image reconstruction from multiscale top points

by F.M.W. Kanters

Master of Science thesis

Project Period: 3/2002-9/2002
Report number: 02A/05
Commissioned by:

Supervisor:
prof. dr. ir. P.P.J. v.d. Bosch (Dep. of Electrical Engineering, TU/e)

Additional Commission members:
prof. dr. ir. B.M. ter Haar Romeny (Dep. of Biomedical Engineering, TU/e)
dr. L.M.J. Florack (Dep. of Biomedical Engineering, TU/e)

/ Department of Electrical Engineering

Abstract

For many image processing tasks, hierarchical and topological methods are required. New approaches to image analysis involving the *deep structure* of images are promising, but many questions about points in scale space are still unanswered. In this thesis, a second order reconstruction algorithm for multiscale points is presented in order to extract information about these points. It is tested with random points and spatially equidistant points in scale space. Using the equidistant points, an optimal distance between reconstruction points is measured, taking the limited machine precision into account. For this measurement, the condition number of the correlation matrix is used. The algorithm is also tested with multiscale critical points and multiscale top points. Two possible applications for the reconstruction from multiscale top points are discussed: data compression for images using reconstructions and content based image retrieval using coefficients of the reconstruction algorithm. The results of both feasibility studies are promising.

keywords: reconstruction, scale space, deep structure, critical points, top points, compression, image retrieval

Contents

Chapter 1

Introduction

Image analysis is an important field of research these days. It is used for a wide variety of applications, e.g. robot vision, video steered animation, security and medical imaging, to name a few. Especially in the medical field, imaging is more and more inspired by the human front end visual system.

One important example is the scale space theory, in the western world introduced by Koenderink [10] and Witkin [25] in the eighties. All our observations are done by integrating some physical property with a measurement device. This can be, for example, integration (over a spatial area) of light on a CCD detector element or receptive field in your eye. The size of the sensitive area of such "devices" is called the *aperture*. The size of the aperture determines the sharpness of the image. The need for different apertures for different tasks can be seen by looking at an image-mosaic (Fig 1.1). To see the larger structure, an other aperture is needed as to see the small images (you slightly blur the image for the larger structure). The human front end visual system is designed to extract information from multiple scales by applying sampling apertures, at a wide range of sizes simultaneously. In image analysis this notion of scale can be introduced by using a stack of images taken at a range of resolutions which is called a scale-space. Koenderink introduced in 1984 the concept of *deep structure* in image analysis as the exploitation of multiple scales of the image simultaneously [10].

Another example of the exploitation of knowledge about the human visual system in image analysis is the differential structure of images. The front end visual system extracts derivatives in space and time to high order from the visual input pattern on the retina. In modern image analysis this can be done by convolving the image with derivatives of a gaussian kernel, regularizing the image. In this differential scale space of an image, a number of interesting points can be found with special differential characteristics, e.g. critical points and top points [4, 2]. These points will be discussed later in this thesis. An interesting work on the scale space theory and the human front end visual system is the monograph by ter Haar Romeny [6].

There are still many open questions about the deep structure of images, and critical points in particular. One of the questions is how much information is contained in these critical points. Much research about reconstruction algorithms is done to get a hold on features which contain crucial image information. For

Figure 1.1: Image mosaic of the Mona Lisa.

example image reconstruction from sign information [20], reconstruction from zero crossings of a wavelet transform [19, 21] and reconstruction from zero crossings in scale space [7]. Nielsen and Lillholm look at the image information of different features [15].

In this thesis, we look at the information contained in scale space top points, by proposing an algorithm for image reconstruction from multiscale points. It is based on the work of Florack [3] and on the work of Nielsen and Lillholm [15]. After the theory is described in chapter 2, the algorithm is tested with different points in scale space, including multiscale top points, in chapter 3. Some possible applications of the scale space top points are discussed in chapter 4 and 5. These include image compression and content based image retrieval. Of course, there are more applications possible, and new applications will arise from further research about the deep structure of images.

Chapter 2

The reconstruction algorithm

2.1 Definitions

- $i = 1, ..., N$: enumeration index for scale space points
- $\phi(x, y, t)$: standard Gaussian at scale t centered at the origin, thus

$$\phi(x, y, t) = \frac{1}{4\pi t} e^{-\frac{x^2+y^2}{4t}}$$

- $f(x, y)$: arbitrary high resolution image
- $\hat{f}(x, y)$: approximation of $f(x, y)$
- $\langle f|g \rangle$: scalar product of $f(x, y)$ and $g(x, y)$,

$$\langle f|g \rangle = \int \int f(x, y) g(x, y)\, dx\, dy$$

2.2 Theory

Consider the basic function $\phi(x, y, t)$. Derivatives of the basic function can be defined as:

$$\phi_{,\nu_1 \ldots \nu_k}(x, y) = \nabla_{\nu_1 \ldots \nu_k} \phi \tag{2.1}$$

Now let us define the basic function:

$$\phi_{i,\nu_1 \ldots \nu_k}(x, y) = \sqrt{2t_i}^{\,k} \phi_{,\nu_1 \ldots \nu_k}(x - x_i, y - y_i, t_i) \tag{2.2}$$

as a normalized Gaussian of scale t_i, centered at x_i, y_i and differentiated to the k-th order with respect to $x^{\nu_1}, \ldots, x^{\nu_k}$ in which we identify $x^1 \equiv x$ and $x^2 \equiv y$. We may call (x_i, y_i) the spatial base point of $\phi_{i,\nu_1 \ldots \nu_k}(x, y)$, and (x_i, y_i, t_i) the scale space base point.
Furthermore, define $L_{i,\nu_1 \ldots \nu_k}$ as the features obtained by taking the scalar product of the original image $f(x, y)$ with the basic function $\phi_{i,\nu_1 \ldots \nu_k}(x, y)$:

$$L_{i,\nu_1 \ldots \nu_k} = \langle\, f \,|\, \phi_{i,\nu_1 \ldots \nu_k} \,\rangle \tag{2.3}$$

$L_{i,\nu_1 \ldots \nu_k}$ is called the Gaussian blurred derivative of the image with respect to $x^{\nu_1}, \ldots, x^{\nu_k}$ at point i. In this thesis, only orders $0 \leq k \leq 2$ are used. The indices $\nu_1, \ldots, \nu_k$ are referred to as spatial indices. A spatial index (in two dimensions) can take only two possible values, interchangeably denoted as "x"and "y", or as "1" resp. "2". The label i is sometimes referred to as an enumeration index. An enumeration index can take arbitrarily many values in principle, say $i = 1, \ldots, N$.

Let us consider a number of scale space base points (x_i, y_i, t_i) with $i = 1, ..., N$ in scale space. For every point i, the features $L_{i,\nu_1 \ldots \nu_k}$ can be calculated. Given this feature space, a second order reconstruction $\hat{f}$ is proposed, as follows:

$$\hat{f}(x,y) = \sum_{i=1}^{N} a_i \phi_i(x,y) + b_i^x \phi_{i,x}(x,y) + b_i^y \phi_{i,y}(x,y) + c_i^{xx} \phi_{i,xx}(x,y) + c_i^{xy} \phi_{i,xy}(x,y) + c_i^{yy} \phi_{i,yy}(x,y) \tag{2.4}$$

which can be shortened using summation convention for the repeated spatial indices to:

$$\hat{f}(x,y) = \sum_{i=1}^{N} a_i \phi_i(x,y) + b_i^\mu \phi_{i,\mu}(x,y) + c_i^{\mu\rho} \phi_{i,\mu\rho}(x,y) \tag{2.5}$$

It can be shown, cf. Appendix A, that the coefficients a_i, b_i^ν and $c_i^{\nu\rho}$ can be determined such that this reconstruction is optimal in L^2-sense. As a constraint on the reconstruction, all features in every point $i = 1, ..., N$ of the reconstruction must be the same as those in the original image, thus in case we adopt the full set of second order constraints,

$$\langle f - \hat{f} \,|\, \phi_i \rangle = 0, \;\; \langle f - \hat{f} \,|\, \phi_{i,\mu} \rangle = 0 \quad \text{and} \quad \langle f - \hat{f} \,|\, \phi_{i,\mu\rho} \rangle = 0 \tag{2.6}$$

for all $i = 1, ..., N$ and for $\mu = x, y$ and $\rho = x, y$, subject to the constraints:

$$\langle f \,|\, \phi_i \rangle = L_i, \;\; \langle f \,|\, \phi_{i,\mu} \rangle = L_{i,\mu} \quad \text{and} \quad \langle f \,|\, \phi_{i,\mu\rho} \rangle = L_{i,\mu\rho} \tag{2.7}$$

for all $i = 1...N$ and for $\mu = x, y$ and $\rho = x, y$.

If (2.5) is substituted in (2.6) using (2.7), the missing coefficients can be calculated from the following linear system of equations:

$$\langle \sum_{i=1}^{N} a_i \phi_i + b_i^\mu \phi_{i,\mu} + c_i^{\mu\rho} \phi_{i,\mu\rho} \,|\, \phi_j \rangle = L_j \tag{2.8}$$

$$\langle \sum_{i=1}^{N} a_i\phi_i + b_i^{\mu}\phi_{i,\mu} + c_i^{\mu\rho}\phi_{i,\mu\rho} \,|\, \phi_{j,\nu} \rangle = L_{j,\nu} \tag{2.9}$$

$$\langle \sum_{i=1}^{N} a_i\phi_i + b_i^{\mu}\phi_{i,\mu} + c_i^{\mu\rho}\phi_{i,\mu\rho} \,|\, \phi_{j,\nu\eta} \rangle = L_{j,\nu\eta} \tag{2.10}$$

with $\mu = x, y$, $\rho = x, y$, $\nu = x, y$ and $\eta = x, y$, which can be rewritten as:

$$\sum_{i=1}^{N} a_i\langle \phi_i \,|\, \phi_j \rangle + b_i^{\mu}\langle \phi_{i,\mu} \,|\, \phi_j \rangle + c_i^{\mu\rho}\langle \phi_{i,\mu\rho} \,|\, \phi_j \rangle = L_j \tag{2.11}$$

$$\sum_{i=1}^{N} a_i\langle \phi_i | \phi_{j,\nu} \rangle + b_i^{\mu}\langle \phi_{i,\mu} | \phi_{j,\nu} \rangle + c_i^{\mu\rho}\langle \phi_{i,\mu\rho} \,|\, \phi_{j,\nu} \rangle = L_{j,\nu} \tag{2.12}$$

$$\sum_{i=1}^{N} a_i\langle \phi_i | \phi_{j,\nu\eta} \rangle + b_i^{\mu}\langle \phi_{i,\mu} | \phi_{j,\nu\eta} \rangle + c_i^{\mu\rho}\langle \phi_{i,\mu\rho} \,|\, \phi_{j,\nu\eta} \rangle = L_{j,\nu\eta} \tag{2.13}$$

By transferring the derivatives to one side, using partial integration, (2.11–2.13) can be written as:

$$\sum_{i=1}^{N} a_i\langle \phi_i \,|\, \phi_j \rangle + b_i^{\mu}\langle \phi_{i,\mu} \,|\, \phi_j \rangle + c_i^{\mu\rho}\langle \phi_{i,\mu\rho} \,|\, \phi_j \rangle = L_j \tag{2.14}$$

$$\sum_{i=1}^{N} -a_i\langle \phi_{i,\nu} | \phi_j \rangle - b_i^{\mu}\langle \phi_{i,\mu\nu} | \phi_j \rangle - c_i^{\mu\rho}\langle \phi_{i,\mu\rho\nu} \,|\, \phi_j \rangle = L_{j,\nu} \tag{2.15}$$

$$\sum_{i=1}^{N} a_i\langle \phi_{i,\nu\eta} | \phi_j \rangle + b_i^{\mu}\langle \phi_{i,\mu\nu\eta} | \phi_j \rangle + c_i^{\mu\rho}\langle \phi_{i,\mu\rho\nu\eta} \,|\, \phi_j \rangle = L_{j,\nu\eta} \tag{2.16}$$

To simplify this we define a generalized correlation matrix as is done by Florack [3]:

Definition 1 *For each combination of spatial indices $\mu_1, ..., \mu_k$ the generalized correlation matrix $\mathbf{\Phi}_{\mu_1,...,\mu_k}$ is the $N \times N$-matrix with components $\Phi_{ij,\mu_1...\mu_k} = \langle \phi_{i,\mu_1...\mu_k} \,|\, \phi_j \rangle$, resulting in:*

$$\Phi_{ij,\mu_1...\mu_k} = \phi_{,\mu_1...\mu_k}(\bar{x}, t)|_{\bar{x}=\bar{x}_{ij}, t=t_{ij}} \quad \text{with } \bar{x}_{ij} = \bar{x}_i - \bar{x}_j \quad \text{and } t_{ij} = t_i + t_j$$

With Definition 1, (2.14–2.16) can be written in matrix form:

$$\overbrace{\begin{pmatrix} \Phi & \Phi_x & \Phi_y & \Phi_{xx} & \Phi_{xy} & \Phi_{yy} \\ -\Phi_x & -\Phi_{xx} & -\Phi_{xy} & -\Phi_{xxx} & -\Phi_{xxy} & -\Phi_{xyy} \\ -\Phi_y & -\Phi_{xy} & -\Phi_{yy} & -\Phi_{xxy} & -\Phi_{xyy} & -\Phi_{yyy} \\ \Phi_{xx} & \Phi_{xxx} & \Phi_{xxy} & \Phi_{xxxx} & \Phi_{xxxy} & \Phi_{xxyy} \\ \Phi_{xy} & \Phi_{xxy} & \Phi_{xyy} & \Phi_{xxxy} & \Phi_{xxyy} & \Phi_{xyyy} \\ \Phi_{yy} & \Phi_{xyy} & \Phi_{yyy} & \Phi_{xxyy} & \Phi_{xyyy} & \Phi_{yyyy} \end{pmatrix}}^{\text{Mixed correlation matrix M}} \times \overbrace{\begin{pmatrix} a_1 \\ \dots \\ a_N \\ b_1^x \\ \dots \\ b_N^x \\ b_1^y \\ \dots \\ b_N^y \\ c_1^{xx} \\ \dots \\ c_N^{xx} \\ c_1^{xy} \\ \dots \\ c_N^{xy} \\ c_1^{yy} \\ \dots \\ c_N^{yy} \end{pmatrix}}^{\text{coefficient vector}} = \overbrace{\begin{pmatrix} L_1 \\ \dots \\ L_N \\ L_1^x \\ \dots \\ L_N^x \\ L_1^y \\ \dots \\ L_N^y \\ L_1^{xx} \\ \dots \\ L_N^{xx} \\ L_1^{xy} \\ \dots \\ L_N^{xy} \\ L_1^{yy} \\ \dots \\ L_N^{yy} \end{pmatrix}}^{\text{feature vector}} \tag{2.17}$$

By solving (2.17), all necessary coefficients of (2.4) can be calculated to make the reconstruction $\hat{f}$. Note that the full system has $6N$ equations and $6N$ unknowns. If not all features of order $0 \leq k \leq 2$ are needed, just remove the corresponding row and column in the matrix M, as well as the corresponding entries in the coefficient vector on the l.h.s. and the feature vector on the r.h.s. For example: If only the second order features are used, remove row 1-3 and column 1-3 from M and the first $3N$ elements of the coefficient vector and feature vector.

2.3 Implementation

The algorithm described in the previous section is first implemented in Mathematica 4 [26]. This program is chosen as the design environment, because it is easy to convert mathematical formulas to a working Mathematica program. The drawback of this implementation is the speed and memory usage. A reconstruction using 500 points takes about 15 minutes on a P4 1.7 GHz and uses about 1.0 GB of memory!

Using the flexibility of the MathLink add-on, it is possible to make a C++ program, which exchanges data with a Mathematica program and interacts with the Mathematica interface. In the current implementation, we use a C++ Mathlink program to calculate the coefficients of the linear system. To calculate the inverse of the matrix, the Intel Math Kernel Library is used, which is optimized for both speed and memory usage. This reduces the calculation

time for reconstruction from 500 points to 5-10 seconds, using about 250 MB of memory.

Note that with the speed improvement, the maximum precision has decreased, due to the usage of double precision floats in the C++ program, instead of the more precise Mathematica numbers. In the next chapters the fast Mathlink implementation is used.

Chapter 3

Image reconstruction from multiscale points

3.1 Reconstruction from one point

As a first example for the reconstruction algorithm described in the previous chapter, a reconstruction is made from one point, using only L, L_x and L_y as features, which is shown in Figure 3.1. At any reconstruction point, the features of the reconstruction *must*, by construction, be the same as the features of the original. However, visually the reconstruction does not have to resemble the original! One expects that resemblance will be enforced if "enough" independent points and features are used. Note that the reconstruction is a combination of zero and first order derivatives of a Gaussian with scale $t = 6$, according to (2.5). Since the y component is very strong and the x component is almost zero, the blob looks like a first derivative of a Gaussian in the y direction, with a larger positive area, to get a correct value of L, as expected.

3.2 Reconstruction from random points

In this section, random points are used to make a reconstruction of two different images. The points are selected with random spatial location according to a uniform distribution, and with decreasing probability as scale increases, such that:

$$N(\tau) = N_0 \, e^{-n\tau} \tag{3.1}$$

with N_0 the number of points for $\tau = 0$ and n the dimension. Here τ is used as the "natural" scale parameter, instead of t, with $t = \frac{1}{2}e^{2\tau}$. Figure 3.2 shows reconstructions of a part of the famous Lena image (64×64 pixels), using 400 random points at scale $\tau = 0.0$, with different sets of features. Note the "holes" in the image, especially at reconstructions from points with little features. Figure 3.3 shows reconstructions of the same image, this time only using L as a feature, again using 400 points, but varying the maximum scale τ_{max}. Here $\tau_{max} = 3$ corresponds with $\sigma \approx 40$ pixels. Note that the difference between the reconstructions with $\tau_{max} = 3$ and $\tau_{max} = 2$ is small. From Figure 3.2 and Figure 3.3 one can conclude that image information increases if the number of

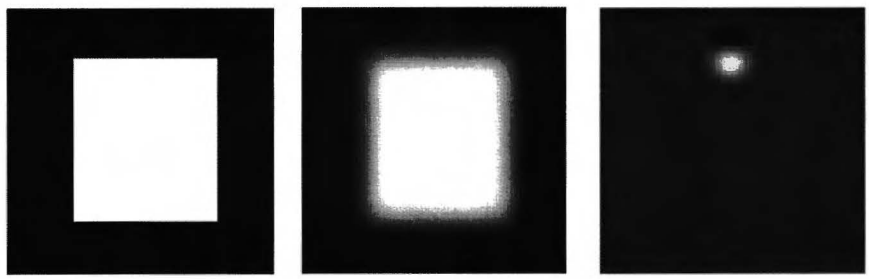

Figure 3.1: Reconstruction from a single point, using only L, L_x and L_y as features. From left to right: original image, blurred original image with $t = 6$, reconstruction using one point. Features of original image at point 1: $L = 132.308$, $L_x = 0.0454528$ and $L_y = -87.1682$, features of reconstructed image at point 1: $L = 132.308$, $L_x = 0.0454528$ and $L_y = -87.1682$. Error of features in point 1: $\Delta L = 6.79 \times 10^{-7}$, $\Delta L_x = 6.18 \times 10^{-11}$ and $\Delta L_y = 3.47 \times 10^{-6}$.

features increases and that if little information is available, higher scales have to be used to get a visually correct reconstruction. Increasing the maximum scale even more does not make the reconstruction visually any better.

Using the previous results, another image is reconstructed. This time, 800 points are used, again with all combinations of features. The maximum scale τ_{max} is chosen in such a way that there are no more "holes" in the reconstructed image. This is done by choosing a τ_{max} and visually check the reconstruction for holes. A high value for τ_{max} will reduce the risk of holes, so this is a reason to choose τ_{max} as high as possible. One reason to choose τ_{max} as low as possible is the fact that for a constant number of points, N_0 is maximal if τ_{max} is minimal, which is important for small details in the image. The optimal τ_{max} will thus be the lowest value with no holes present in the reconstruction. Figure 3.4 shows the result of reconstructions from random points, where for every feature set, the optimal τ_{max} is visually determined. The original image is one slice of a MR brain scan (128×128 pixels). Note that in the MR image, part of the skull is artificially removed, for 3D visualization (ray-tracing) of the brains under the skull.

Surprising is the fact that there is no clear relation between the optimal τ_{max} and the features used for reconstruction! However, there might be a relation between the amount of information contained in the used features and the optimal τ_{max}.

3.3 Reconstruction from equidistant points

As can be seen in Figure 3.2 through Figure 3.4, the visual quality of the reconstruction depends not only on the features used, but also on the interaction between the separate blobs. Therefore, the distance between blobs plays an important role in the quality of the reconstruction. In this section we use

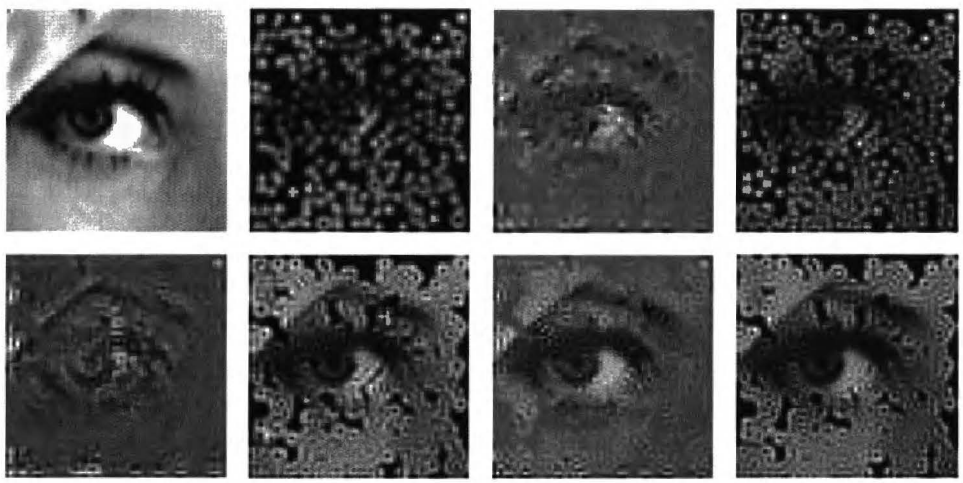

Figure 3.2: Reconstructions of Lena's eye (64×64 pixels) from 400 random points at scale $\tau = 0.0$. From left to right, top to bottom: original image, reconstructions using $\{L\}$, $\{L_x, L_y\}$, $\{L, L_x, L_y\}$, $\{L_{xx}, L_{xy}, L_{yy}\}$, $\{L, L_{xx}, L_{xy}, L_{yy}\}$, $\{L_x, L_y, L_{xx}, L_{xy}, L_{yy}\}$ and $\{L, L_x, L_y, L_{xx}, L_{xy}, L_{yy}\}$ as features.

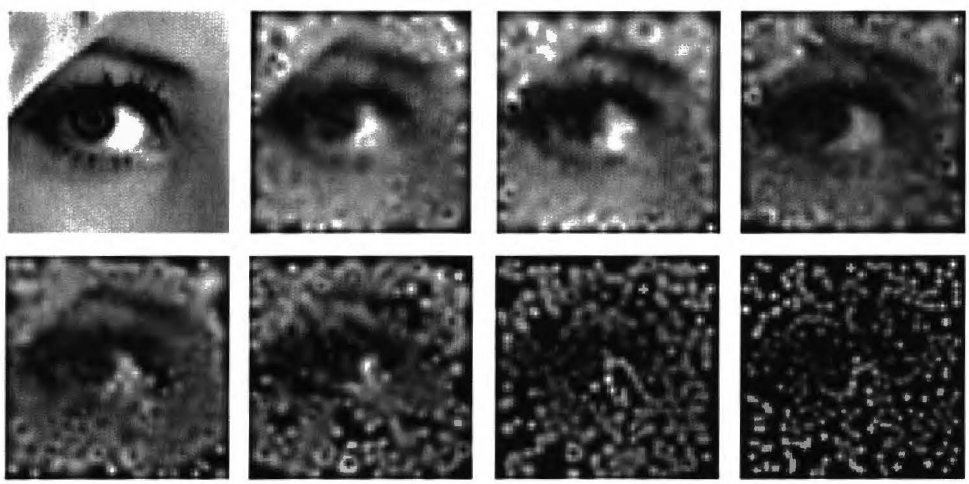

Figure 3.3: Reconstructions of Lena's eye (64×64 pixels) from 400 random points using only L as feature. From left to right, top to bottom: original image, reconstructions using $\tau_{max} = 3.0$, $\tau_{max} = 2.5$, $\tau_{max} = 2.0$, $\tau_{max} = 1.5$, $\tau_{max} = 1.0$, $\tau_{max} = 0.5$, $\tau_{max} = 0.0$.

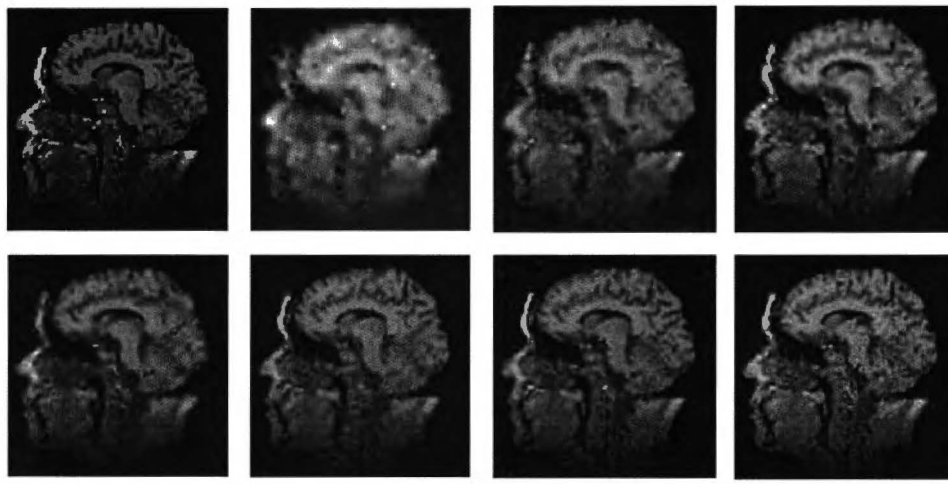

Figure 3.4: Reconstructions of MR brain scan (128×128 pixels) from 800 random points. From left to right, top to bottom: original image, reconstructions using $\{L, \tau_{max} = 3.0, N_0 = 80\}$, $\{L_x, L_y, \tau_{max} = 3.0, N_0 = 80\}$, $\{L, L_x, L_y, \tau_{max} = 2.0, N_0 = 87\}$, $\{L_{xx}, L_{xy}, L_{yy}, \tau_{max} = 3.0, N_0 = 80\}$, $\{L, L_{xx}, L_{xy}, L_{yy}, \tau_{max} = 1.5, N_0 = 96\}$, $\{L_x, L_y, L_{xx}, L_{xy}, L_{yy}, \tau_{max} = 1.8, N_0 = 89\}$ and $\{L, L_x, L_y, L_{xx}, L_{xy}, L_{yy}, \tau_{max} = 1.1, N_0 = 110\}$.

equidistant points for the reconstruction to easily measure the influence of the distance between blobs for the reconstruction quality. For measurement of the reconstruction quality the Root Mean Square error is often used. Although it is proven not to be a very good measurement for the visual quality, it is still good enough for our purpose of quantitative reconstruction. The RMS error is given by:

$$||f - \hat{f}||_{L^2} = \sqrt{\frac{1}{NM} \sum_{i=1}^{N} \sum_{j=1}^{M} \epsilon_{ij}^2} \tag{3.2}$$

with ϵ_{ij} the pixel-wise difference between the reconstruction and original and M and N the dimensions of the image.

The RMS error is caused by the lack of completeness of the feature set and by the error made due to the limited precision of the system. Especially the calculation of the inverse of the matrix in (2.17) can introduce errors due to limited machine precision. In our case, we use the Intel Math Kernel Library to calculate the inverse, which has the following properties:

Definition 2 *For a Matrix* $\mathbf{A}$ *with dimensions* $n \times n$ *the norm* $\|\mathbf{A}\|_\infty$ *is defined by:*

$$\|\mathbf{A}\|_\infty \hat{=} \max_i \sum_{j=1}^{n} |a_{ij}|$$

Data perturbations. If $\bar{x}$ is the exact solution of $\mathbf{A}\bar{\mathrm{x}} = \bar{\mathrm{b}}$, and $\bar{x} + \delta\bar{x}$ is the exact solution of a perturbated problem $(\mathbf{A} + \delta\mathbf{A})\bar{\mathrm{x}} = (\bar{\mathrm{b}} + \delta\bar{\mathrm{b}})$, then

$$\frac{\|\delta\bar{x}\|}{\|\bar{x}\|} \leq \kappa_\infty(\mathbf{A})\left(\frac{\|\delta\mathbf{A}\|}{\|\mathbf{A}\|} + \frac{\|\delta\bar{\mathrm{b}}\|}{\|\bar{\mathrm{b}}\|}\right), \text{ where } \kappa_\infty(\mathbf{A}) = \|\mathbf{A}\|_\infty \|\mathbf{A}^{-1}\|_\infty \tag{3.3}$$

The amplification factor $\kappa_\infty(\mathbf{A})$ is called the condition number of matrix A. Note that the norm $\|.\|$ is the standard quadratic norm, while $\|\mathbf{A}\|_\infty$ is as in Definition 2.

Rounding errors. If ϵ is the *machine precision*, and $c(n)$ is a modest function of the matrix order n, then

$$\frac{\|\delta\bar{x}\|}{\|\bar{x}\|} \leq c(n)\,\kappa_\infty(\mathbf{A})\,\epsilon \tag{3.4}$$

So if the condition number is very large, the error in the inverse due to rounding errors is also very large. In practice, $c(n) = \mathcal{O}(n)$.

Now let us consider a reconstruction from a number of points at a fixed scale τ, with $t = \frac{1}{2}e^{2\tau}$, which are equally distributed over the spatial domain. The points lie on a grid with distance D. Note that if D decreases, the number of points N increases. Figure 3.5 shows the RMS error and the condition number κ versus the distance D for different scales for the full feature set with order $k \leq 2$. The original image is a white rectangle in a black background. Some results of the reconstruction using different D can be seen in Figure 3.6. Note that the optimal fit yields $D \approx 0.9\sigma$ with $\sigma = \sqrt{2t}$, in agreement with the expected linear scaling behavior.

The same measurement of the RMS error versus the distance D is done for the MR image of the brain, for some higher values of τ. Figure 3.7 shows the results, and some of the reconstructions. Note the odd result for $D = 4.8$, where the error is much higher than for $D = 4.4$, probably due to some special interaction between two pixels, such as very high derivative values in opposite directions, which causes the condition number to explode. More research about this problem is needed. Note that again the optimal $D \approx 0.9\sigma$ with $\sigma = \sqrt{2t}$, except for higher values of σ, where it is closer to $D \approx 0.8\sigma$.

The remarkable thing about the RMS graphs in Figure 3.5 and Figure 3.7 is the asymptotic behavior towards $D \to 0$. This indicates a numerical problem due to increasing mutual dependencies of features. This is in agreement with the fact that if less features are taken into account, for example if only L_{xx}, L_{xy} and L_{yy} or if only L is taken as a feature, the same results are found, but the curves are shifted to the left. Less features at a fixed distance D, means less pixel interaction, so the condition number will be lower. Conclusion is that points too close together will give problems due to machine limitations.

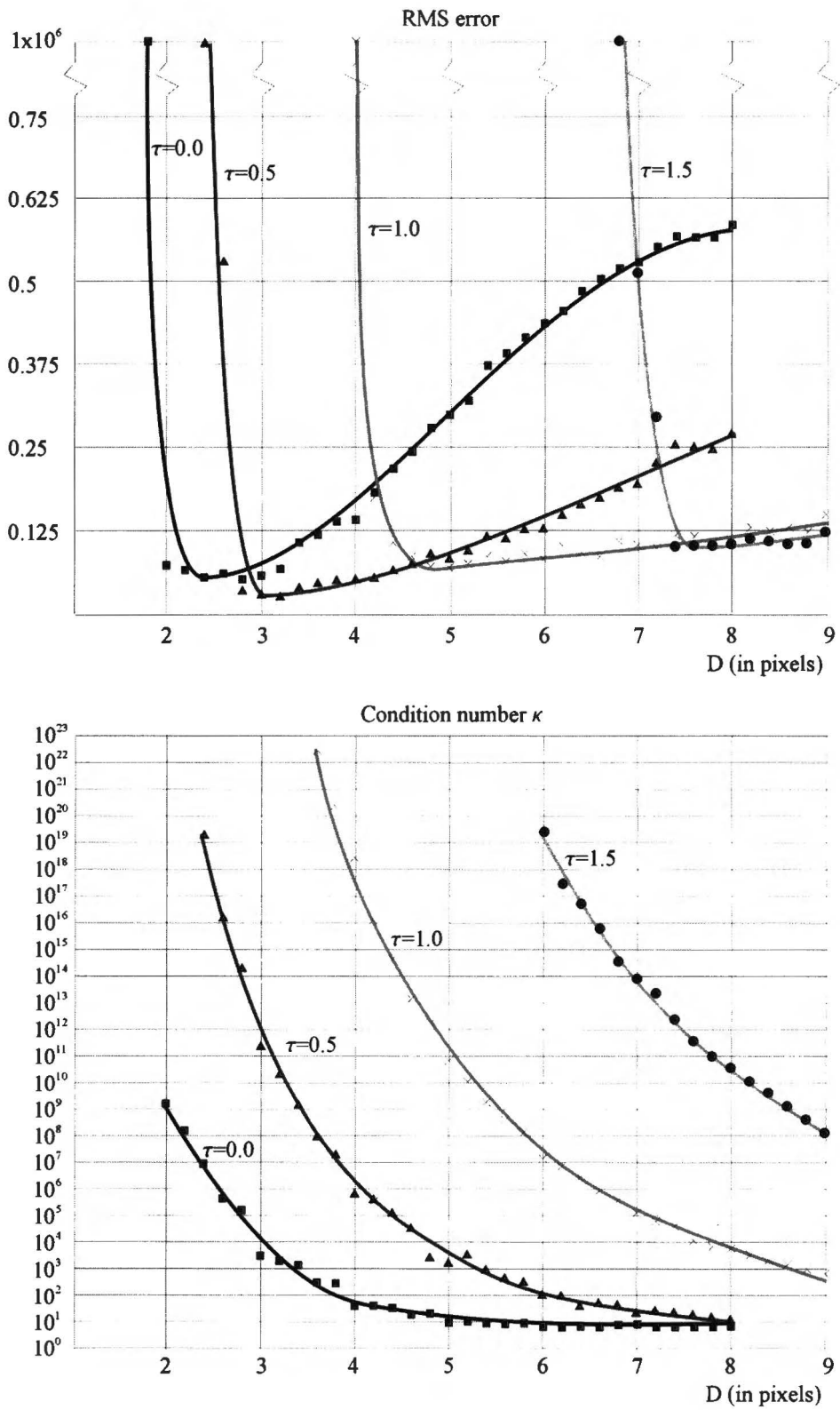

Figure 3.5: Root Mean Square error and Condition number κ versus pixel distance D of image block.tif, using $L, L_x, L_y, L_{xx}, L_{xy}$ *and* L_{yy} as features. Scale parameter τ is used instead of t ($t = \frac{1}{2}e^{2\tau}$)

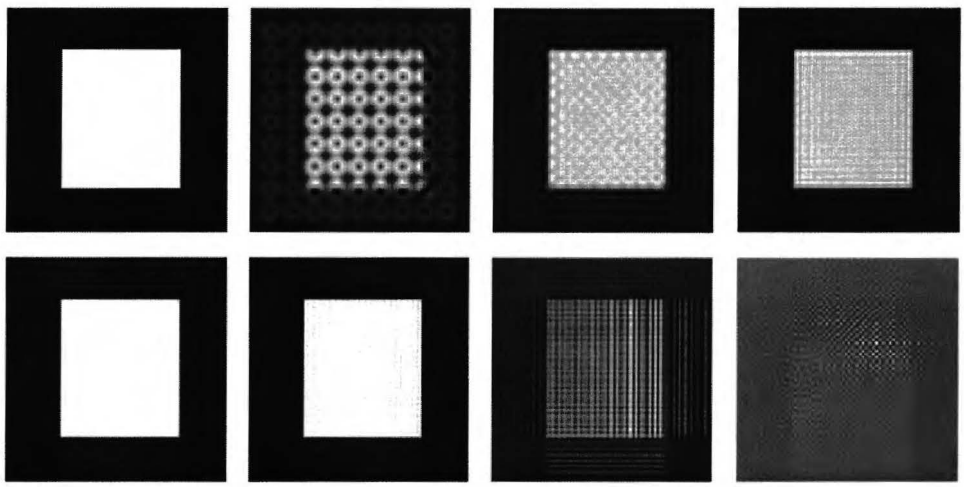

Figure 3.6: Reconstructions of block.tif, using $L, L_x, L_y, L_{xx}, L_{xy}$ and L_{yy} as features. Points are equidistant at fixed scale $\tau = 0.5$. From left to right: original, reconstructions with distance D=6, 5, 4, 3, 2.8, 2.6, 2.4.

3.4 Reconstruction from multiscale critical points

The equidistant points of the previous section are suitable for a good reconstruction, but can contain much redundant information. According to Nielsen and Lillholm [15], there are points in scale space which contain more image information than others, for example edge points and blobs. They also look which features are best suited for different points.

In this section, an experiment[1] is done with reconstructions from multiscale top points as described by Florack et al. [3]. The approach is the same as the reconstructions by Nielsen and Lillholm [15], but with different points and features. First we define (spatial) critical points and top points:

Definition 3 *spatial critical points are points where the spatial gradient is zero. For 2D images these points are maxima, minima or saddles.*

Definition 4 *top points are critical points where the Hessian degenerates (det H=0). For generic 2D images, these points are annihilations or creations of saddles with maxima or minima.*

where the Hessian of a 2D image f is given by:

$$Hf = \nabla \nabla f = \begin{pmatrix} \partial_x^2 f & \partial_x \partial_y f \\ \partial_y \partial_x f & \partial_y^2 f \end{pmatrix} \tag{3.5}$$

1. Note that due to time limitations, it is only a first try to reconstruct images from top points and much further research is needed to get a proper overview of the possibilities and limitations.

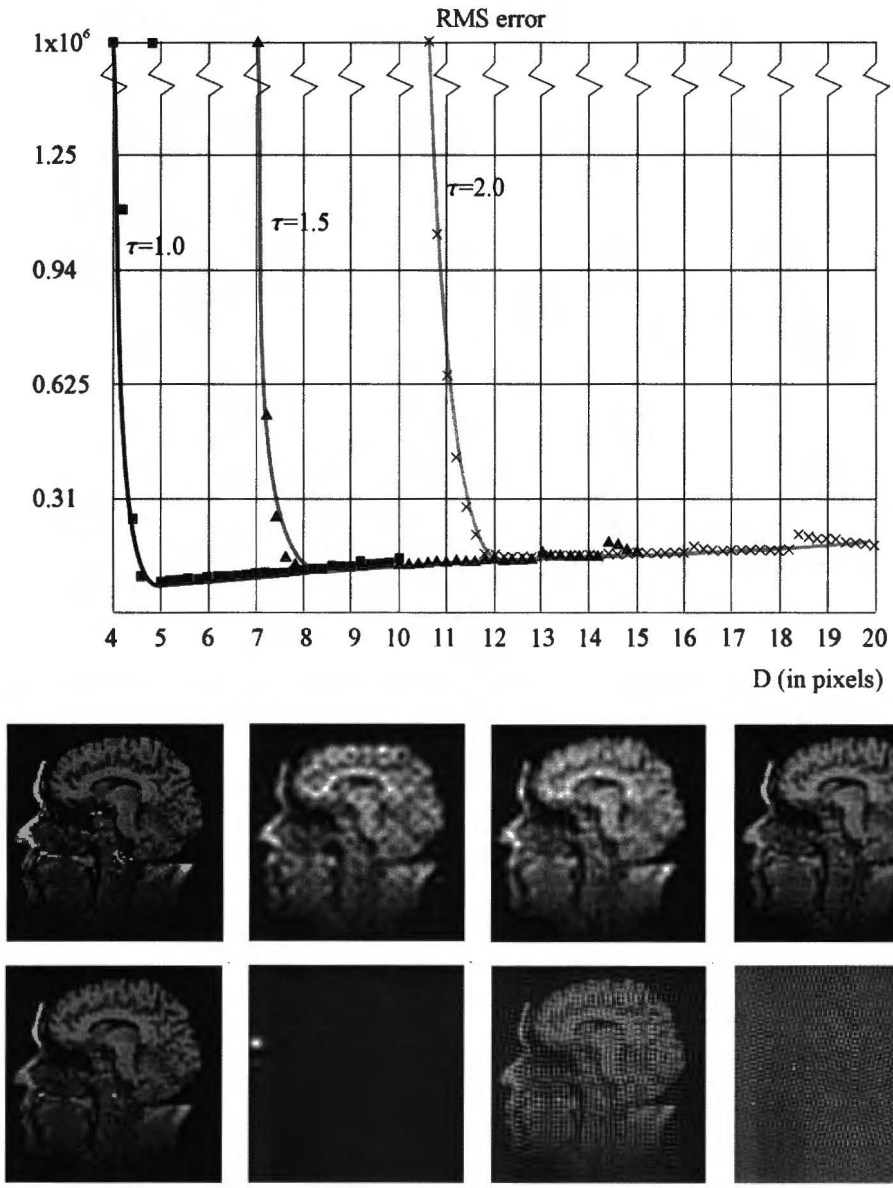

Figure 3.7: Top: Root Mean Square error versus pixel distance D of image mrbrain.tif, using $L, L_x, L_y, L_{xx}, L_{xy}$ *and* L_{yy} as features. Scale parameter τ is used instead of t ($t = \frac{1}{2}e^{2\tau}$) Bottom: Reconstructions of mrbrain.tif, using $L, L_x, L_y, L_{xx}, L_{xy}$ and L_{yy} as features. Points are equidistant at fixed scale $\tau = 1.0$. From left to right: original, reconstructions with distance D=10, 8, 6, 5, 4.8, 4.4, 4.0.

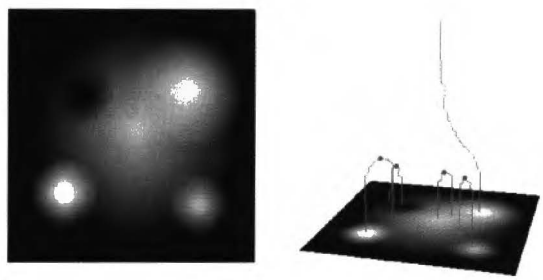

Figure 3.8: Simple example top points. Left: original image, right: 3D view of critical paths and top points.

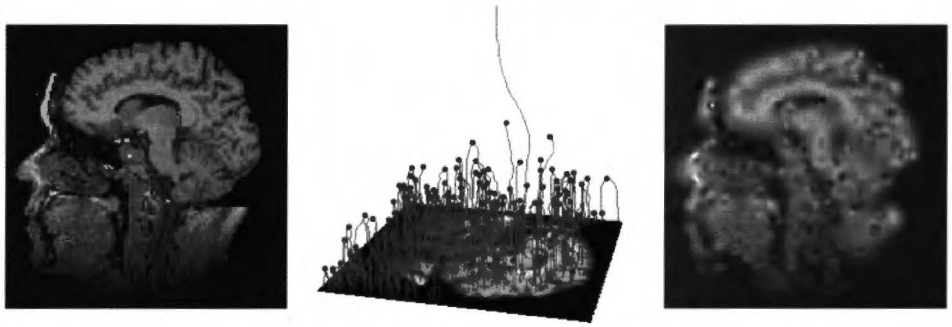

Figure 3.9: Reconstruction from top points of mrbrain.tif. Left: original image, Center: 3D view of critical paths and top points, Right: Reconstruction from top points, using all features with order $k \leq 2$

Given an image, top points can be found by tracking all maxima, minima and saddles in scale, and finding those points where pairs of saddles annihilate with maxima or minima and points where pairs of saddles and maxima or minima are created. Figure 3.8 shows a simple example of top points of an image. Note that according to Loog et al. [12], always one critical path from a maximum or minimum will be left at the highest scale. More about critical points and top points can be found in [2],[4],[8] and [11].

In our experiment, we use these top points for image reconstruction. Again we use the mrbrain.tif image. Figure 3.9 shows the original image, with the 3D view of the critical paths and top points. It also shows a reconstruction from all top points, using the full feature set with order $k \leq 2$. The number of top points for this image is 211. This seems to be insufficient, at least for the reconstruction order $k \leq 2$ actually used. Critical points are tracked in scale from $\tau = 0.0$ to $\tau = 4.0$. As can be seen in figure 3.9, the reconstruction does not contain much detail and resembles only very coarse the original. This is partly due to the fact that no top points can be found at scales $\tau \leq 0.0$ (which is equal to $t \leq \frac{1}{2}$), because we start tracking at that scale.

A better reconstruction can be made if all critical points at scale $\tau = 0$ are added to the top points. To prevent calculation problems, points closer than D_{opt} (calculated from figure 3.7) are deleted. The number of points increases

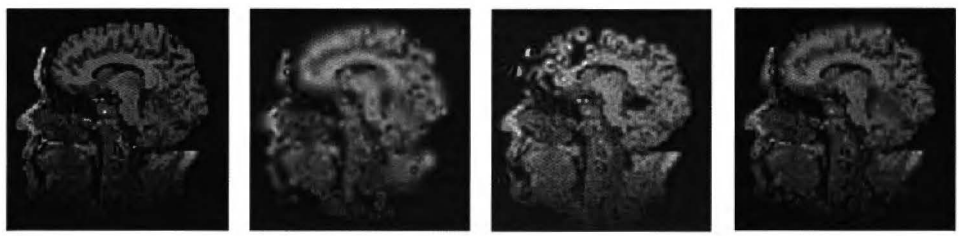

Figure 3.10: Reconstruction from top points and critical points at scale $\tau = 0$ of mrbrain.tif. From left to right: original image, reconstructions from: top points, critical points at scale $\tau = 0$, top points and critical points at scale $\tau = 0$. All reconstructions use all features of order $k \leq 2$

to 779 if these points are added. In figure 3.10, the result of the reconstruction with top points combined with critical points at scale $\tau = 0$ is shown.

The reconstruction using top points and critical points at scale $\tau = 0$ looks reasonable, but compared to the reconstruction from equidistant points shown in figure 3.7 it is not very good, also considering the fact that in the latter case only 576 points are used, whereas here 779 points are used. Note that for reconstruction from critical points, features L_x and L_y are known (gradient zero) and therefore the total amount of information needed for the reconstruction is reduced.

Probably, better results can be achieved using top points at lower scales, or using more information from the critical paths, e.g. scale space saddles (points on the critical paths where the laplacian is also zero).

Chapter 4

Application example 1: Image compression

A possible application of the reconstruction algorithm is data compression, due to the limited number of points needed for a visually good reconstruction. In this chapter, a quick comparison is made between different reconstructions and some different jpeg compressed images. Again, we used the MR-brain image (128×128 pixels). For the reconstructions we used random points, equidistant points and top-points with critical points as described in chapter 3. The jpeg images are all made in Corel PhotoPaint 7 by applying baseline optimized jpeg compression to the original uncompressed image. One image is compressed with quality 17 (lower number means higher image quality), one image is compressed with quality 65 and one image is compressed with quality 74. Figure 4.1 shows the results of the different methods of compression. Although it is proven that the RMS error is not the best way to compare compression algorithms, it is still widely used for this purpose. In Table 4.1, all the methods with their properties and RMS error are shown.

Looking at Figure 4.1 and Table 4.1, some observations can be made. Compression using reconstructions from random points (Figure 4.1e) does not make much sense. Due to the random distribution, it is possible to have "holes" in the image, which is undesirable. Reconstructions from top points and critical points (Figure 4.1b) can be used for compression, but are outperformed by jpeg (Figure 4.1f) at the same compression ratio, visually as well as regarding the RMS error. Reconstructions using equidistant points (Figure 4.1c and d) can be used better for compression, but they are still outperformed by jpeg (Figure 4.1g and h). The reconstructions look more blurred. Note that the different jpeg images do not differ much in visual quality, while the RMS error does increase as the compression ratio increases.

The jpeg compression outperforms the reconstruction methods for all compression ratios, but at lower compression ratio's (e.g. 1:2.1) the visual difference between jpeg en reconstruction from equidistant points is not very large. For images with less dense information, however, the difference will be much larger, because reconstruction from equidistant points does not take the image structure into account (and thus large homogeneous areas will be badly compressed). Note that most lossless compression algorithms can gain the same compression ratio (order 1:2). For high compression ratios (order of 1:5-1:10) the reconstruction from top points and critical points is used. The difference between jpeg

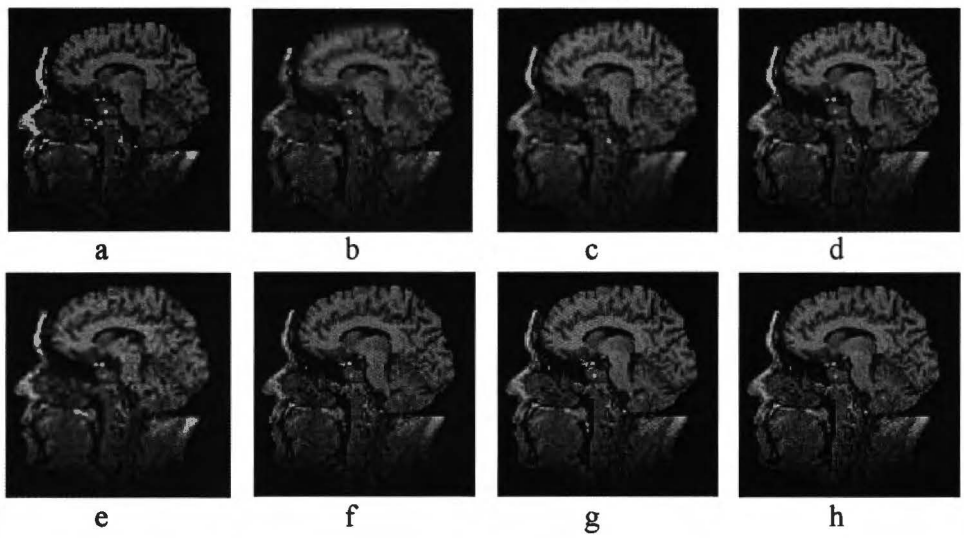

Figure 4.1: Image compression examples. Top, from left to right: Original (128 × 128), reconstruction from top points and critical points, reconstruction from equidistant points with $\tau = 1.0$, reconstruction from equidistant points with $\tau = 0.5$. Bottom, from left to right: Reconstruction from random points, jpeg with quality 74, jpeg with quality 65, jpeg with quality 17, using Corel PhotoPaint 7.

and reconstruction from top points and critical points is large, but the reconstruction from these points is not yet optimal. The advantage is that image structure *is* taken into account here, in contrast with the reconstructions from equidistant points. While this was only a first test of reconstructions from top points and critical points, a lot of further research is needed to get optimal results.

Note that the ratios for the reconstructions are all based on double precision floating point numbers (64 bits), while the ratios for the jpeg images are based on 8 bit gray values. Reconstruction can thus be used for compression of high precision images, while most standard compression implementations can't handle these images. Many medical applications these days such as MR and CT use 12 bits or 14 bits gray scale images.

Also note that the number of features used for reconstruction from top points and critical points is 6. Because the fact that by definition the first derivatives are (close to) zero, we only need to transfer or store 4 features instead of 6, as can be seen in table 4.1. This improves the compression ratio, while quality is maintained.

Another interesting difference between jpeg compression and compression using top points and critical points is the fact that the number of points, and therefore the compression ratio, cannot be controlled, because it depends heavily on the image structure. For some applications this is a drawback (e.g. real time applications with fixed bandwidth) but for other applications this is a nice property (e.g. automatic selection of optimal compression ratio for storage of images).

The calculation time needed is still a major drawback. Where jpeg compression

takes only little time, reconstruction using 700 points with 6 features each takes several seconds. For larger images this can give problems for many applications.

Note that there are a lot of different implementations of the jpeg compression algorithm, which will all yield different results. Running the same tests as described here with for example Adobe Photoshop 5 will result in a completely different comparison.

While only one image is compared, this is not a very thorough research about the possibilities of image compression using reconstruction from multiscale points, but recall that it was merely intended as a quick feasibility study. This chapter shows that at this moment it does not appear feasible to use our method for image compression. Compression using equidistant points can be used but has no advantages over standard jpeg. Compression using reconstructions from top points and critical points could be useful for some applications in the future, but at this point, the quality of the reconstruction is not sufficient [1]. Also many questions are still unanswered. For example, it is not clear how the reconstruction performs with larger images or color images. Future research must show whether using reconstructions from multiscale points is really useful for image compression or not.

1. It is not unlikely that reconstruction quality can be improved by taking into account different (notably higher order) features

Table 4.1: Reconstruction as compression tool

Method	Number of Points	Number of features	Compression Ratio [1)]	RMS error
Original	16384	n/a	n/a	n/a
Random points	800	6	1:3.4	19.8
top points and critical points	779	4	1:5.3	13.0
jpeg quality setting 74	n/a	n/a	1:5.3	6.4
Equidistant points $\tau = 1.0$	576	6	1:4.7	10.8
jpeg quality setting 65	n/a	n/a	1:4.7	5.9
Equidistant points $\tau = 0.5$	1296	6	1:2.1	7.2
jpeg quality setting 17	n/a	n/a	1:2.1	1.8

1) For reconstructions, the compression ratio is calculated as $1:\frac{16384}{\text{No. of points} \times \text{No. of features}}$, for jpeg it is calculated as $1:\frac{16384}{\text{image size} - 330 \text{ bytes (from header)}}$. Header size determined by single pixel jpeg compressed image.

Chapter 5

Application example 2: Content based image retrieval using top points

5.1 Introduction

In this chapter a feasibility study is presented for using multiscale top points in a content based image retrieval system. The goal of such a system is to find, given an image, the closest matches to that image in a large image database, looking at the image content. Many of such systems exist, but IBM's Query by Image Content (QBIC)[14] is probably the best-known one. The user has to specify a number of parameters prior to the searching, so much labor is still needed here. Some examples of more automated systems are Virage [5] and VisualSeek [22], which are based on texture and material structure. Color histogram based systems are also commonly used [23, 24]. These systems, however, do not take into account the spatial distribution of the features used. Some examples of systems which do use spatial information are systems which use segmented image regions [13, 1]. In the following sections, we propose a completely new approach, based on multiscale top points.

5.2 Proportional Transportation Distance (PTD)

In chapter 3 we tried to reconstruct images using multiscale top points. These top points could be used to describe an image. This way, we can translate the content based image retrieval problem to comparing "distances" between two sets of (top) points in a multidimensional space. For this purpose we use the Proportional Transportation Distance (PTD) for weighted point sets from Giannopoulos and Veltkamp [16]. First, we define the point sets.

Definition 5 *Let* $A = \{a_1, a_2, \ldots, a_m\}$ *be a weighted point set such that* $a_i = \{(x_i, w_i)\}, i = 1, \ldots, m$, *where* $x_i \in \mathbb{R}^k$ *with* $w_i \in \mathbb{R}^+ \cup \{0\}$ *being its corresponding weight. Let also* $W = \sum_{i=1}^{m} w_i$ *be the total weight of set A.*

Definition 6 *Let* $B = \{b_1, b_2, \ldots, b_n\}$ *be a weighted point set such that* $b_j = \{(x_j, u_j)\}, j = 1, \ldots, n$, *where* $x_j \in \mathbb{R}^k$ *with* $u_j \in \mathbb{R}^+ \cup \{0\}$ *being its corresponding weight. Let also* $U = \sum_{j=1}^{n} u_j$ *be the total weight of set B.*

The PTD can then be defined as follows:

Definition 7 *Let A and B be two weighted point sets and d_{ij} a ground distance between point a_i and b_j. The set of all feasible flows $\mathcal{F} = [f_{ij}]$ from A to B, is now defined by the following constraints:*

$$\begin{array}{ll} \text{(i)} & f_{ij} \geq 0, i = 1, \ldots, m, j = 1, \ldots, m \\ \text{(ii)} & \sum_{j=1}^{n} f_{ij} = w_i, i = 1, \ldots, m \\ \text{(iii)} & \sum_{i=1}^{m} f_{ij} = \frac{u_j W}{U}, j = 1, \ldots, n \\ \text{(iv)} & \sum_{i=1}^{m} \sum_{j=1}^{n} f_{ij} = W \end{array}$$

then the PTD can be defined as:

$$PTD(A, B) = \frac{min_{F \in \mathcal{F}} \sum_{i=1}^{m} \sum_{j=1}^{n} f_{ij} d_{ij}}{W}$$

The PTD can be seen as the minimum amount of work needed to transform A into a new set A' that resembles B. In particular, we redistribute A's total weight from the position of its points, to the position of B's points leaving the old percentages of weights in B the same. For calculation of the PTD, we used a fast implementation of the Earth Movers Distance (EMD) which is publicly available [27, 17]. In our case, we used normalized weights, such that $U = W = 1$.

There are two major parameters with which we can experiment. The first one is the ground distance d_{ij}. All top points have the following properties: $a_i = \{(x_i, y_i, t_i, C_i, C_{i,x}, C_{i,y}, C_{i,xx}, C_{i,xy}, C_{i,yy})\}$, with $C_{i,\mu\rho}$ the reconstruction coefficient of the feature $L_{i,\mu\rho}$ (see chapter 3). One possible ground distance is the Euclidian distance for only the x and y coordinates, discarding all other features. Another possibility is to include the reconstruction coefficients $C_{i,\mu\rho}$ of each point in the ground distance. The second tuneable parameter is the weight w_i for each point. We can use equal weights for all points, but we can also use different weights for each point, using the reconstruction coefficients of chapter 3. In the next section a number of different settings for these parameters are used for content based image retrieval.

5.3 Experimental results

In the previous section we described a method to calculate a distance between two weighted point sets, the PTD. We can use this PTD for content based image retrieval in a large database. In the experiments described in this section, we used a subset of the Face database from the Olivetti Research Laboratory, made by Samaria and Harter [18]. The subset consists of 200 images of faces from 20 people (10 images each, with different deviations such as pose, glasses, distortion). From every image of this set, the top points and the reconstruction coefficients were calculated. For the content based image retrieval experiment we used the first image of each person as a query and looked at the 9 images with the smallest PTD to this image, using different parameters.

Figure 5.1: Experiment 1: $d_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + (t_i - t_j)^2}$ and weights are equally distributed. Leftmost images are queries, neighboring images are closest matches, with increasing PTD.

5.3.1 Experiment 1

In our first experiment the ground distance is defined as:

$$d_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + (t_i - t_j)^2} \tag{5.1}$$

and the weights are equally distributed, thus:

$$w_i = \frac{1}{m} \tag{5.2}$$

$$u_j = \frac{1}{n} \tag{5.3}$$

While the ground distance is a little ad-hoc and the weights do not contain any information, the results are surprising. As can be seen in Figure 5.1, the images in the database with the smallest PTD are mostly from the same person. For the full set of test results, see Appendix 2. Note that in the second row, the results seem to be independent of the glasses! Also note that the third row is one of the worst query results from the database for this experiment (in this case, the hair line seems to have strong features). Somehow, the *structure* of the faces is more important than the pose of the person for the PTD algorithm.

5.3.2 Experiment 2

In our second experiment the ground distance is defined as in (5.1), but the weights are different. Our first try was to take the mean of the absolute values of the reconstruction coefficients (again admittedly an ad hoc choice):

$$w_i = \frac{|C_i| + |C_{i,x}| + |C_{i,y}| + |C_{i,xx}| + |C_{i,xy}| + |C_{i,yy}|}{6} \tag{5.4}$$

$$u_j = \frac{|C_j| + |C_{j,x}| + |C_{j,y}| + |C_{j,xx}| + |C_{j,xy}| + |C_{j,yy}|}{6} \tag{5.5}$$

The results of this first try can be seen in Figure 5.2. As can be seen, the results are absolutely not close to the results from the first experiment. It seems that some points have a much too large weight. Looking at the range of the reconstruction coefficients, the problem becomes more clear. Sometimes the reconstruction coefficients can be as high as 10^6, resulting in very strong points, where distance is not very important anymore. As a second try we take the log of the weights from (5.4):

$$w_i = Log(\frac{|C_i| + |C_{i,x}| + |C_{i,y}| + |C_{i,xx}| + |C_{i,xy}| + |C_{i,yy}|}{6}) \tag{5.6}$$

$$u_j = Log(\frac{|C_j| + |C_{j,x}| + |C_{j,y}| + |C_{j,xx}| + |C_{j,xy}| + |C_{j,yy}|}{6}) \tag{5.7}$$

The results are shown in Figure 5.3. The results are much better now, but still not better than with equally distributed weights! It is clear that one has to think more carefully about the way in which the incommensurable features are to be incorporated.

5.3.3 Experiment 3

For image retrieval, it is desired to be independent of translations and rotations of the query image. When looking at the top point structure of an image, one question is how much information is contained in the position of the top points. As a final experiment we tried a ground distance which is translation invariant, because the x and y coordinates are left out. We propose the ground distance as:

$$d_{ij} = \sqrt{(t_i - t_j)^2 + (C_i - C_j)^2 + (C_{i,\nu} - C_{j,\nu})^2 + (C_{i,\mu\rho} - C_{j,\mu\rho})^2} \tag{5.8}$$

For all spatial indices ν, μ, ρ. The weights are taken the same as in (5.6). The results for this experiment are shown in Figure 5.4. It is very surprising to see that the results are better without the position of the top points in the ground distance! Still there are some queries which give problems, as can be seen in the fourth row of Figure 5.4. The big advantage with this ground distance is the invariance regarding translation. If the reconstruction would be based on rotation invariant filters, e.g. absolute value of the gradient for the first order and the laplacian for the second order, this approach for image retrieval would also be rotation invariant. Maybe also scale invariance could be obtained if t is left out from the ground distance.

Figure 5.2: Experiment 2a: $d_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + (t_i - t_j)^2}$ and mean of absolute reconstruction coefficients as weights. Leftmost images are queries, neighboring images are closest matches, with increasing PTD.

Figure 5.3: Experiment 2b: $d_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + (t_i - t_j)^2}$ and Log of mean of absolute reconstruction coefficients as weights. Leftmost images are queries, neighboring images are closest matches, with increasing PTD.

Figure 5.4: Experiment 3a: $d_{ij} = \sqrt{(t_i - t_j)^2 + (C_i - C_j)^2 + (C_{i,\nu} - C_{j,\nu})^2 + (C_{i,\mu\rho} - C_{j,\mu\rho})^2}$ and Log of mean of absolute reconstruction coefficients as weights. Leftmost images are queries, neighboring images are closest matches, with increasing PTD.

Note that for all described methods, some odd results can show up. For example the fifth image on the third row of Figure 5.4. These results can not be explained at this moment and have to be further investigated.

Chapter 6

Conclusions and discussion

The goal of the research was to find out how much image information is contained in multiscale points (critical and top points in particular), by making a reconstruction of the original image, using only these points and some local jet attributes in scale space. Also a few possible applications using these reconstructions are discussed. From the experiments described in this thesis, we can conclude that only a part of this goal is achieved. We found some interesting results, supporting the idea that some points contain much image information, but more research is required to fully understand the amount of information contained in those points. Therefore, a number of topics for new research have come up, which will be described here.

Regarding the reconstruction algorithm, a few conclusions can be drawn. For the reconstruction, a large matrix has to be inverted, which can introduce errors if points are close together. This can be seen in the condition number of the matrix. A pseudo-inverse algorithm can be the solution, which has to be investigated. The experiments regarding equidistant points are only at one scale. Points at different scales, which are close together, sometimes lead to a high condition number, but sometimes they do not. The reason is still an open question, which also has to be investigated. Nielsen and Lillholm used in [15] an extra constraint to retain total image energy. The effect of this extra constraint must be explored. The reconstruction as described in this thesis uses only features of order $k \leq 2$. The effect of using higher order derivatives as features is yet still unknown (But it is unlikely that orders higher than 4 are of much influence, since the correlation between gaussian derivatives of higher orders is high). The possibility to use different orders of features at different scales has to be implemented and examined, at this moment only one set of features for all points can be used. A coarse to fine approach of the reconstruction algorithm should also be explored. If it exists it may solve memory problems as well as ill-conditioning.

The reconstruction from random points is useful for a first test, and can be used to show the influence of different features and scales for the reconstruction. Reconstruction from equidistant points is useful for testing the possibilities and limitations of the algorithm. Problems due to limited machine precision can be pointed out clearly using equidistant points, which gives us an optimal distance between points. Regarding the reconstruction from top points, only a rather ad-

hoc experiment is done due to time limitation. Our first results show that the top points only are not sufficient for a high quality second order reconstruction. One possible reason is the fact that the detection of those top points in an image is not yet perfect. At this moment, no top points at scale $\tau \leq 0$ are found. It also might be that higher order features are needed here. Using also the critical points at scale $\tau = 0$ resulted in much better reconstructions. Maybe, the top points themselves do not contain enough information, but the critical paths do. Further research about reconstruction from top points and critical paths must be done to determine the true image information contained in these points.

The usage of reconstructions from several different multiscale points for data compression of images does not look very promising at this moment. Compression using reconstructions from equidistant points yields reasonable results, but they are outperformed by jpeg at the same compression ratio. Using reconstructions from top points and critical points could be useful for some applications in the future, but the visual quality of the reconstructions is too low at this moment. An advantage could be the fixed compression ratio of this method, which is dependant on the image structure. Note that in this thesis, only one image is tested for image compression, and many questions have to be answered before we can actually decide whether this method is useful for image compression or not.

For content based image retrieval, an approach using the Proportional Transportation Distance between sets of top points weighted with reconstruction coefficients looks promising. The two main parameters of the used PTD algorithm have to be chosen carefully to work correctly. If a ground distance is chosen without any information about the spatial position of the top points, a system which is invariant to translation can be achieved, with very good results. Invariance regarding translation, rotation *and* scaling could be obtained by using a different ground distance and a different reconstruction base, but was not tested.

The results in this thesis do not answer the question about the amount of information contained in critical points and top points. It rather gives an idea about the potential of these points and directions for further research about this topic.

Acknowledgements.

We thank Panos Giannopoulos and Remco Veltkamp for their implementation of the Proportional Transportation Distance (PTD) algorithm and Bram Platel for the Mathematica code for the detection of the scale space critical points. Further I want to thank the Biomedical Image Analysis Group of the Eindhoven University of Technology, especially my supervisors Luc Florack and Bart ter Haar Romeny, for their help and useful discussions.

Bibliography

[1] C. Carson, M. Thomas, S. Belongie, J. M. Hellerstein, and J. Malik. Blobworld: A system for region-based image indexing and retrieval. In *Third International Conference on Visual Information Systems*. Springer, 1999.

[2] L. Florack and A. Kuijper. The topological structure of scale-space images. *Journal of Mathematical Imaging and Vision*, 12(1):65–79, February 2000.

[3] L. M. J. Florack. Reconstruction from scale space critical points. Internal report.

[4] L. D. Griffin and A. C. F. Colchester. Superficial and deep structure in linear diffusion scale space: Isophotes, critical points and separatrices. *Image and Vision Computing*, 13(7):543–557, September 1995.

[5] Amarnath Gupta and Ramesh Jain. Visual information retrieval. *Communications of the ACM*, 40(5):70–79, 1997.

[6] B. M. ter Haar Romeny. *Front-End Vision and multiscale image analysis*. Kluwer Academic Publishers, 2002. Leerboek over schaalruimtemodellen voor digitale beeldverwerking en de eerste stadia van het visuele systeem, geschreven in Mathematica.

[7] R. Hummel and R. Moniot. Reconstruction from zero crossings in scale space. In *IEEE Transactions on Acoustics, Speech, and Signal Processing*, volume 37, pages 2111–2130, 1989.

[8] S. N. Kalitzin, B. M. ter Haar Romeny, A. H. Salden, P. F. M. Nacken, and M. A. Viergever. Topological numbers and singularities in scalar images: Scale-space evolution properties. *Journal of Mathematical Imaging and Vision*, 9(3), November 1998.

[9] M. Kerckhove, editor. *Scale-Space and Morphology in Computer Vision: Proceedings of the Third International Conference, Scale-Space 2001, Vancouver, Canada*, volume 2106 of *Lecture Notes in Computer Science*. Springer-Verlag, Berlin, July 2001.

[10] J. J. Koenderink. The structure of images. *Biological Cybernetics*, 50:363–370, 1984.

[11] A. Kuijper and L.M.J. Florack. The application of catastrophe theory to image analysis. Submitted to Image and Vision Computing.

[12] M. Loog, J. J. Duistermaat, and L. M. J. Florack. On the behavior of spatial critical points under Gaussian blurring. a folklore theorem and scale-space constraints. In Kerckhove [9], pages 183–192.

[13] Wei-Ying Ma and B. S. Manjunath. Netra: A toolbox for navigating large image databases. *Multimedia Systems*, 7(3):184–198, 1999.

[14] et.al. M.Flickner, H.Sawhney. Query by image and video content: the qbic system. *IEEE Computer*, 28(9):23–32, 1995.

[15] M. Nielsen and M. Lillholm. What do features tell about images? In Kerckhove [9], pages 39–50.

[16] R. Veltkamp P. Giannopoulos. A pseudo-metric for weighted point sets. In *ECCV 2002, LNCS 2352*, pages 715–730. Springer, 2002.

[17] Y. Rubner. Code for earth movers distance (emd). http://vision.stanford.edu/~rubner/emd/default.htm.

[18] F. Samaria and A. Harter. Parameterisation of a stochastic model for human face identification, 1994.

[19] J.L. Sanz. Multidimensional signal representation by zero crossings: an algebraic study. *Society for Industrial and Applied Mathematics, Philadelphia, PA, USA*, 49:281–295, 1989.

[20] J.L.C. Sanz and T.T. Huang. Image representation by sign information. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1(7):729–738, July 1989.

[21] M. Shmouely and Y.Y. Zeevi. Image representation by level crossings of the wavelet transform. *IEEE Transactions on Image Processing*, 2:565–568, 1996.

[22] J. Smith and S. Chang. Single color extraction and image query, 1995.

[23] Markus Stricker and Michael Swain. The capacity and the sensitivity of color histogram indexing. Technical Report TR-94-05, University of Chicago, 3 1994.

[24] M. Swain and D. Ballard. Color indexing. *International Journal on Computer Vision*, 7(1):11–32, 1991.

[25] A. P. Witkin. Scale-space filtering. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 1019–1022, Karlsruhe, Germany, 1983.

[26] S. Wolfram. *The Mathematica Book*. Wolfram Media/Cambridge University Press, fourth edition, 1999.

[27] L.J. Guibas Y. Rubner, C. Tomasi. A metric for distributions with applications to image databases. In *IEEE International Conference on Computer Vision, Bombay, India*, pages 59–66, 1998.

Appendix A

Derivation of the reconstruction formula

For reconstruction we want to minimize $||f - \hat{f}||_{L^2}$ and minimize the error in the features. Using Euler-Lagrange, we obtain:

$$S[\hat{f}] \stackrel{def}{=} \frac{1}{2}|| \hat{f} ||^2_{L^2} + \sum_i \lambda_i \langle f - \hat{f} \mid \phi_i \rangle \tag{A.1}$$

Using the functional derivative:

$$\frac{\delta S[\hat{f}]}{\delta \hat{f}} \stackrel{def}{=} \hat{f} - \sum_i \lambda_i \phi_i \tag{A.2}$$

we can determine the unique solution of $\frac{\delta S[\hat{f}]}{\delta \hat{f}} = 0$:

$$\hat{f} = \sum_i \lambda_i \phi_i \tag{A.3}$$

which is the $\hat{f}$ that minimizes the L^2-norm if the coefficients λ_i are calculated by substitution of A.3 in:

$$\langle f - \hat{f} \mid \phi_j \rangle = 0 \tag{A.4}$$

Using (A.1)-(A.3) for our second order case, with filters $\phi_{i,\nu_1 \ldots \nu_k}$, we obtain (2.5) for the reconstruction formula $\hat{f}$. The coefficients a_i, b_i^μ and $c_i^{\mu\rho}$ can be calculated using (A.4) as constraints with the correct filters substituted, resulting in solving (2.17).

Appendix B

Results of the content based image retrieval experiments

Figure B.1: Experiment 1: $d_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + (t_i - t_j)^2}$ and weights are equally distributed. Leftmost images are queries, neighboring images are closest matches, with increasing PTD.

Figure B.2: Experiment 2a: $d_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + (t_i - t_j)^2}$ and mean of absolute reconstruction coefficients as weights. Leftmost images are queries, neighboring images are closest matches, with increasing PTD.

Figure B.3: Experiment 2b: $d_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + (t_i - t_j)^2}$ and Log of mean of absolute reconstruction coefficients as weights. Leftmost images are queries, neighboring images are closest matches, with increasing PTD.

Figure B.4: Experiment 3a: $d_{ij} = \sqrt{(t_i - t_j)^2 + (C_i - C_j)^2 + (C_{i,\nu} - C_{j,\nu})^2 + (C_{i,\mu\rho} - C_{j,\mu\rho})^2}$ and Log of mean of absolute reconstruction coefficients as weights. Leftmost images are queries, neighboring images are closest matches, with increasing PTD.