Adaptive Morphological Filtering of Incomplete Data

Anders Landström, Matthew J. Thurley, and Håkan Jonsson
Department of Computer Science, Electrical and Space Engineering
Luleå University of Technology
Luleå, Sweden
{anders.p.landstrom, matthew.thurley, hakan.jonsson}@ltu.se

Abstract—We demonstrate how known convolution techniques for uncertain data can be used to set the shapes of structuring elements in adaptive mathematical morphology, enabling robust morphological processing of partially occluded or otherwise incomplete data. Results are presented for filtering of both gray-scale images containing missing data and 3D profile data where information is missing due to occlusion effects. The latter demonstrates the intended use of the method: enhancement of crack signatures in a surface inspection system for casted steel.

The presented method is able to disregard unreliable data in a systematic and robust way, enabling adaptive morphological processing of the available information while avoiding any false edges or other unwanted features introduced by the values of faulty pixels.

Index Terms—Mathematical morphology, Adaptive morphology, Normalized convolution, Occluded data.

I. INTRODUCTION

A. Background

Mathematical morphology is a non-linear framework for image processing, originally developed by Matheron [1] and Serra [2]. The methods are based on set theory, focusing on geometrical structure. In mathematical morphology filtering is performed by probing an image with functions known as structuring elements (SEs), considering where in the data the SEs do or do not fit. In standard mathematical morphology, the same SE (defined by the user) is used to filter the whole image. This is often not ideal, however, and therefore there is an ongoing interest in adaptive mathematical morphology where SEs are allowed to vary from pixel to pixel.

Previous work [3] has demonstrated how the Local Structure Tensor (LST) can be used to control the shapes of elliptical SEs for adaptive mathematical morphology. These elliptical SEs range from lines to disks depending on the degree of orientation dominancy (anisotropy) around the pixel for which each of the individual SEs is defined. We here present how the methods can be extended by established methods for handling partly uncertain data, enabling processing of incomplete (i.e. partly missing) data directly without need for additional pre-processing steps such as inpainting or other types of reconstruction of missing information.

Separating between known and unknown pixel values, i.e. considering pixel certainty, is highly important whenever input data is incomplete. In particular, pixel certainty should be used to handle missing values in our intended application: crack detection in 3D steel surface profile data captured by laser triangulation. In this type of data, pixel values may be missing due to occlusion, which occurs when the surface itself blocks the line-of-sight between the camera sensor and the laser line projected onto the measured surface. Such missing pixels may be located within cracks but also appear around scales, which constitute a brittle oxidized top layer resulting from the casting process itself.

Many different methods for reconstruction of images exist, including inpainting [4] and kernel methods [5]. Inpainting strives to fill holes of missing values with statistically likely data, similar to what an artist would do when reconstructing a missing piece of a painting. Kernel methods relies on regression or interpolation of missing data based on pixel values in the surrounding regions, as defined by a kernel function penalizing distance away from the considered pixel. Our goal, however, is not to achieve reconstruction of missing values, but to prohibit missing pixels from affecting the filtered image. Hence, rather than first reconstructing data from some given model, we use a kernel method known as normalized convolution [6] to define a method for adaptive morphological filtering which is unaffected by missing data. The resulting method ignores missing pixels and adjusts affected quantities accordingly. This is achieved using only a few required user-set parameters, simplifying filter optimization for a given task.

B. Contribution

This work combines existing techniques, and demonstrates how a robust method for adaptive morphological processing which can handle missing pixel values without additional pre- or post-processing can be achieved by adopting existing computationally effective methods intended for processing data of signal/certainty type. The contribution of this work thereby lies not in the individual parts, but in how they are combined into a robust technique for adaptive morphological processing of partly occluded or otherwise incomplete data.

We emphasize the difference towards reconstruction methods: reconstruction methods could certainly be considered as a pre-processing step, but this work demonstrates how missing data can be merged directly into the technique without any need for additional pre-processing.

The presented method is not limited to use for 3D profile data, but can be applied for adaptive morphological filtering...
of any type of incomplete images where not all pixel values are considered reliable. This could be a result of faulty image sensors, surface reflectance properties, or image pixels being lost or distorted during data transmission.

II. THEORY

A. Adaptive Morphology using Elliptical Structuring Elements

The LST is a mathematical entity holding information about orientational structure in the data. Originally introduced by Knutsson [7], it constitutes a commonly used method for estimating directional features in 2D images as well as in higher dimensions.

As demonstrated in Ref. [3], adaptive elliptical SEs for mathematical morphology can be set for each pixel \( x = (x_1, x_2)^T \) from its local structure tensor

\[
T(x) = G_\sigma \ast \left( \nabla f(x) \nabla^T f(x) \right). \tag{1}
\]

Here \( f(x) \) denotes the image value for each pixel \( x \), \( \nabla = \left( \frac{\partial}{\partial x_1}, \frac{\partial}{\partial x_2} \right)^T \) is the gradient (nabla) operator, and \( G_\sigma \) is a Gaussian kernel with standard deviation \( \sigma \) which regularizes the matrix. For a more thorough background on image processing using the LST, the reader is referred to Ref. [8].

In Ref. [3] the eigenvalues and eigenvectors of \( T(x) \), which contain the dominant orientation in the data around each pixel, are used to set an elliptical SE for each pixel. More specifically, the semi-major axis length \( a(x) \) and the semi-minor axis length \( b(x) \) are defined as

\[
a(x) = \frac{\lambda_1(x) + \epsilon}{\lambda_1(x) + \lambda_2(x) + 2\epsilon} M, \tag{2}
\]

\[
b(x) = \frac{\lambda_2(x) + \epsilon}{\lambda_1(x) + \lambda_2(x) + 2\epsilon} M, \tag{3}
\]

where \( \lambda_1(x) \) and \( \lambda_2(x) \) are the (positive) eigenvalues of the local structure tensor \( T(x) \), \( \epsilon \) is a small positive number (i.e. machine epsilon) added to handle situations where the eigenvalues are zero, and \( M \) is the user-defined maximum length of the semi-major axis. The result is an adaptive method for mathematical morphology, in which elliptical SEs \( E(x) \) align to directional structures of the data but remains disk-shaped where no such dominant direction exists. These SEs are so-called flat, meaning that they simply describe a neighborhood in which the morphological operations are performed (as opposed to SEs in the form of functions).

Using flat SEs \( E(x) \) the two basic morphological operations erosion, \( \varepsilon_E(f) \), and dilation, \( \delta_E(f) \), of an image \( f(x) \) are defined as

\[
\varepsilon_E(f) = \Lambda \{ f(u) : u \in E(x) \}, \tag{4}
\]

\[
\delta_E(f) = \nabla \{ f(u) : u \in E(x) \}, \tag{5}
\]

\( \Lambda \) and \( \nabla \) denotes the minimum and maximum operators, respectively. The morphological opening \( \gamma_E \) and closing \( \phi_E \) are then defined as

\[
\gamma_E(f) = (\delta_E \circ \varepsilon_E)(f), \tag{6}
\]

\[
\psi_E(f) = (\varepsilon_E \circ \delta_E)(f). \tag{7}
\]

The above formulation does not, however, produce satisfactory results for data where pixel information is uncertain or even missing, since all pixels are allowed to affect the final outcome although some may be completely irrelevant due to low certainty but can hold extreme values which has serious consequences for the minimum/maximum operations.

B. Convolving Data of Signal/Certainty Type

Normalized convolution was introduced by Knutsson and Westin [6] and provides a mean for processing of data where pixel values are not considered equally reliable. This is done by assigning to each pixel a corresponding certainty weight representing our trust in the (scalar or higher tensor) value of that pixel.

Following the notation of Knutsson and Westin [6] (slightly adapted for our terminology), a generalized convolution can be defined as

\[
\{ a B \ast c f_T \} (x) = \sum_{u \in N_x} a(u-x) B(u-x) \ast c(u) f_T(u), \tag{8}
\]

where \( a \) and \( c \) denotes the applicability and certainty weight functions, respectively, \( B \) contains the set of basis functions used, and \( f_T \) here represents the tensor-valued signal (in our case either the gray-level function \( f \) or the LST \( T \)). The \( \ast \) symbol denotes a multi-linear operation, and the \( \ast \) symbol marks the operation over which the convolution is performed. \( N_x \) is the neighborhood of pixels \( u \) around \( x \) on which the applicability function \( a(u-x) \) is non-zero. Note that we define \( u \) and \( x \) in a global coordinate system, while the difference \( \xi = u - x \) is a local coordinate, relative to \( x \), on which the applicability function and the basis functions depend.

Normalized convolution can then be formulated as

\[
C_N(x | a, B, c, f_T) = N^{-1}(x) D(x), \tag{9}
\]

where

\[
D(x) = \{ a B \ast c f_T \} (x), \tag{10}
\]

\[
N(x) = \{ a B^* \ast c \} (x), \tag{11}
\]

and provides a tool for performing convolution operations on signals where our trust in the pixel (tensor) values varies over the image. In order to simplify notation, we have omitted to denote the dependency of \( D \) and \( N \) on \( a, B, c, \) and \( f_T \).

The output, \( C_N \), contains a set of basis coefficients which are optimal in a least-squares sense. By using polynomial basis functions for approximating the signal within a certain neighborhood (including the constant basis \( b_0 = 1 \)), i.e. by calculating the coefficients \( \beta_k, k = 0, 1, 2 \), for the three polynomial basis functions \( \{ b_k \} = \{ 1, \xi_1, \xi_2 \} \) in

\[
f(x + \xi) \approx \sum_{k=0}^3 \beta_k b_k(\xi) = \beta_0 \cdot 1 + \beta_1 \cdot \xi_1 + \beta_2 \cdot \xi_2, \tag{12}
\]

we can estimate the gradient as the coefficients of the two bases \( b_1 = \xi_1 \) and \( b_2 = \xi_2 \), given by the local pixel coordinate
\( \xi = [\xi_1, \xi_2]^T \). To see this, simply consider the first order Taylor expansion around a pixel \( x \) given by

\[
    f(x + \xi) \approx f(x) + \frac{\partial f(x)}{\partial x_1} \cdot \xi_1 + \frac{\partial f(x)}{\partial x_2} \cdot \xi_2, \tag{13}
\]

where \( \nabla \) again is the gradient operator. From (12) and (13), the estimated coefficients \( \beta_1 \) and \( \beta_2 \) are identified as the approximated components of the function gradient.

In practice, this can be done by another related operation: normalized differential convolution. The mathematical expression of this operation, given by

\[
    C_{N\Delta}(x | a, B, c, f_T) = N^{-1}_{\Delta}(x)D_{\Delta}(x), \tag{14}
\]

where

\[
    D_{\Delta}(x) = \left\{ [a \circ c] \{ aB \odot c f_T \} \right\}(x), \tag{15}
\]

\[
    N_{\Delta}(x) = \left\{ [a \circ c] \{ aB \odot c^* \} \right\}(x). \tag{16}
\]

is more complex, and for more details the reader is referred to Knutsson and Westin [6], [9]. Most importantly for our purposes, it removes the need for a constant basis and therefore reduces the dimensionality of the involved matrices. It has been shown that the result of normalized differential convolution is equivalent to that of normalized convolution with the constant basis added [10].

### III. Method

In the following text we assume that our input data is of signal/certainty type, i.e. every pixel value, denoted either \( f(x) \) or \( T(x) \) depending on whether we are referring to the gray-level \( f \) or the tensor \( T \), is associated with a corresponding certainty (or confidence) weight \( c(x) \in \mathbb{R} \) [0, 1], representing our trust in the value at pixel \( x \).

In the case of unknown data, e.g. occluded pixels in laser triangulation, we have for the original input data \( f(x) \)

\[
    c_f(x) = \begin{cases} 
    0 & \text{if the value of } x \text{ is unknown,} \\
    1 & \text{otherwise.} 
    \end{cases} \tag{17}
\]

As already pointed out, this case cannot be directly handled by the more naive approach given by (1)–(3), but the tensor formulation allows for a straight-forward adaption by applying normalized convolution. Generalizing convolutions for signals with corresponding certainty, this concept is highly useful for our purpose.

#### A. Estimating the Gradient

The benefits from using normalized differential convolution for estimating gradients in the case of missing data have been demonstrated by Knutsson and Westin [6], and we here simply adapt their terminology to this work: Following (12) and (13) we can estimate the gradient components as the coefficients \( \beta_1 \) and \( \beta_2 \) for the corresponding basis functions \( b_1(\xi) = \xi_1 \) and \( b_2(\xi) = \xi_2 \), as estimated by (14)–(16), i.e. the gradient \( \nabla f \) is given by

\[
    \nabla f(x) \approx C_{N\Delta}(x | a, [\xi_1, \xi_2]^T, c_f(x), f) = N^{-1}_{\Delta}(x)D_{\Delta}(x), \tag{18}
\]

where \( a_{\xi} \) is a user-chosen applicability function defining the influence of the neighborhood around a considered point.

#### B. Estimating the LST

We should not use our estimated \( \nabla f \) directly in (1), but instead identify that equation as a particular case of normalized convolution: Equation (1) can be seen as normalized convolution of a fully trusted signal (full certainty for all pixels) using a constant basis.

In our case, the fact that the estimated gradients \( \nabla f(x) \) rely on different amounts of available information around \( x \) should be taken into account. Hence, we need to set a certainty \( c_{\nabla f}(x) \), representing our trust in the gradient obtained by (18). A simple approach would be to use the quantity \( a_{\xi} \circ c_f \) (the standard convolution of the certainty weights), but this would not take the effects of the basis functions into account. The quantity \( N_{\Delta} \), however, contains a description of the certainties associated with the new basis functions [6], and thereby holds certainty information for the estimated gradient \( \nabla f \). As a measure of this certainty, we use the determinant of \( N_{\Delta} \), i.e.

\[
    c_{\nabla f}(x) = |N_{\Delta}(x)|. \tag{19}
\]

This choice of \( c_{\nabla f} \), previously used by Westin [9, p. 104], captures the amount of certainty associated with the new basis functions and allows for reduction of the impact on the final local structure tensor from gradients estimated from a small number of pixels on behalf of gradients estimated using more available information.

The local structure tensor \( T(x) \) can now be calculated from (9)–(11) as

\[
    T(x) = C_N(x | a_1, c_{\nabla f}(x), \nabla f(\nabla f)^T) = N^{-1}_{\Delta}(x)D(x), \tag{20}
\]

where \( a_1 \) is an applicability function chosen by the user (note that \( a_1 \) is, in general, different from \( a_{\xi} \)). This applicability determines the level of smoothing of the tensor field. Note that in practice the operations can be performed element-wise for the three defining elements of the symmetric \( 2 \times 2 \) matrix \( (\nabla f(\nabla f)^T) \), which simplifies calculations drastically.

#### C. Adaptive Morphological Processing

The eigenvectors and eigenvalues can then be used to set the elliptical SEs as described by (2) and (3) in Sect. II-A. In the actual morphological processing, based on min/max operations over the pixel-dependent neighborhood defined by the elliptical SEs, missing data is then handled by simply ignoring the values of such pixels. This is easily achieved by setting the corresponding pixel values to \(-\infty / +\infty \) depending on type of operation.

Apart from the maximum length of the semi-major axis, \( M \), the applicability functions \( a_{\xi} \) and \( a_1 \) need to be defined. We
set them as Gaussian kernels defined by radial bandwidths $r_\xi$ and $r_1$, related to the standard deviations $\sigma_\xi$ and $\sigma_1$ of the two Gaussian kernels by

$$\sigma(\cdot) = \frac{r(\cdot)}{\sqrt{2\ln 2}}.$$  \hspace{1cm} (21)

Hence, the applicability functions $a_\xi$ and $a_1$ decrease to half their maximum value at distances $r_\xi$ and $r_1$, respectively, from their centers.

For overview, the whole method can then be summarized into the following steps:

1) Calculate the gradient $\nabla f$ by (18). Except for the input image and its certainty the user needs to provide $r$, which defines the smoothing used when calculating the gradient (thereby regularizing gradient calculations).

2) Retrieve the LST $T(x)$ by (20). The user needs to set $r_1$, which defines the spatial scale for which the LST should be representative.

3) Calculate eigenvalues and eigenvectors for $T(x)$.

4) Define an elliptical structuring element for each pixel using (2) and (3). The parameter $M$ needs to be provided by the user.

5) Perform morphological operations based on (4) and (5), setting missing pixel values to $+\infty$ or $-\infty$ respectively.

IV. EXPERIMENTS AND RESULTS

In order to test the robustness of the presented method, we must know what the result would be in case we would have no missing data. This cannot be achieved for occlusions in measured 3D profile data, and we therefore apply the resulting morphological operations on two different data sets:

A. Gray-scale images corrupted by randomly removing pixels, and

B. 3D surface data for a casted steel surface, containing occluded pixels.

Apart from providing a case where we know the “truth”, the gray-scale data case also demonstrates the usefulness of the method on other types of data, although it was designed specifically for 3D profile data.

A. Gray-scale Image with Randomly Removed Data

The performance of the presented strategy for assigning shapes to SEs was studied by randomly removing an increasing fraction $\alpha$ of the pixels in a set of 30 gray-scale test images (see examples in Fig. 1) and studying how the result of the morphological filtering of the corrupted data compares to the filtering of the complete original data. For comparison, results for the uncorrupted data were also compared to the method in Ref. [3] using the same set of test images.

More specifically, we calculate the Root Mean Square Error (RMSE) for the adaptive opening $\gamma_\alpha$ at each level of missing data, as compared to the adaptive opening $\gamma_0$ of the original data, i.e.

$$RMSE(\gamma_\alpha) = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (\gamma_\alpha(x_i) - \gamma_0(x_i))^2}$$  \hspace{1cm} (22)

where $x_i, i = 1, 2, \ldots, N$, denote the $N$ pixels in the image.

For comparison against an opening $\gamma_R$ of uncorrupted data using the method described in Ref. [3] (with corresponding parameters), $RMSE(\gamma_R)$ was calculated, which resulted in RMSE values within the interval $[0.2169] \cdot 10^{-2}$ (mean value $0.8 \cdot 10^{-2}$ and standard deviation $0.2 \cdot 10^{-2}$). Hence, there are only small differences between the results of the two approaches, given fully certain data.

The error resulting from increased noise was compared to the corresponding RMSE values for two other alternative
Fig. 3. Calculated structuring elements (left) and corresponding openings (right) for an original uncorrupted image (a and b), the case when missing pixels are ignored in the min/max operations only (c and d), and the presented method where missing pixels are completely ignored throughout the whole procedure (e and f). Parameters were set to $r_2 = 1.25$, $r_1 = 4$, and $M = 5$.

Fig. 4. Calculated structuring elements (left) and corresponding openings (right) for an original uncorrupted image (a and b), the case when missing pixels are ignored in the min/max operations only (c and d), and the presented method where missing pixels are completely ignored throughout the whole procedure (e and f). Parameter setup: $r_2 = 1.25$, $r_1 = 4$, and $M = 5$.

approaches:

A1. Completely disregarding certainty (all values are considered valid in all operations).

A2. Considering certainty in the morphological min/max operations but not when calculating the SEs.

Results for the image set are shown in Fig. 2.

The error resulting from completely disregarding certainty is of course extreme, since more and more pixels in the resulting opened image will be set to zero by the increasing amount of black pixels in the original image. This result is trivial, but emphasizes why certainty must be taken into account when processing data where all pixel values cannot be considered equally valid. The RMSE values, although showing better results for our presented method, do not differ that much depending on whether certainty is considered when setting SE shapes or not, but this is because the RMSE is a global measure of image similarity and may not fully depict differences at a more local level. If we study the resulting images more in detail (see examples in Figs. 3 and 4), we see the benefit from taking confidence into account here as well: Details in the image become much more blurred if missing pixels are allowed to affect the shape of the SEs, because the contrast towards the randomly distributed black pixels cause evenly distributed high gradient values and thereby makes SEs wider overall. One can also easily predict that in the case of non-random distributions of missing pixels, any border along regions of missing data will cause directional bias for the resulting elliptical SEs.
B. 3D Surface Data Containing Occluded Pixels

We now turn to our intended application of the presented method: detection of cracks in casted steel slabs. The objective is to enhance crack signatures in 3D profile data acquired by laser triangulation, without letting missing pixels give rise to false crack-like features in the data. Figure 5 shows two crack examples and the adaptive openings of the data using the presented approach. Figure 6 shows examples of scale regions which are also present in the measured data, but should not be falsely detected as cracks. Occluded pixels in the original data are shown in red. Note that the spatial distribution of the occluded pixels along the edges of the scales in Fig. 6 are not unlike the dark crack pixels in Fig. 5.

The crack signature is successfully enhanced, while non-crack pixels containing missing data are not allowed to affect the result noticeably. There are a few pixels in the opening of the scales data where the SE could not reach correctly measured data. The number of such pixels obviously depends on the size of the structuring element. However, letting them remain with zero output certainty is actually not very problematic since they can be easily identified and excluded from further analysis by considering the amount of occluded and non-occluded pixels covered by the SE. An important benefit with the presented method is that occluded pixels are not allowed to affect other values in any way, so that the problem...
is contained within the originally occluded region.

The examples demonstrate the robustness of the presented method: the adaptive opening has enhanced and linked crack segments without being corrupted or otherwise distorted by the occluded pixels.

V. DISCUSSION

We have shown how techniques can be combined into a robust technique for adaptive morphological processing, handling the presence of missing pixels within the method itself. By formulating the problem so that the power of normalized convolution can be utilized for setting the shapes of Structuring Elements (SEs), incomplete data can be handled in a reasonable and systematic way.

Our results show that 3D profile data containing occluded pixels can be robustly processed, without introducing artificial structures that may be mistaken for cracks. If the SE is too small it may only cover missing data, but this can be handled by simply measuring the amount of valid (non-missing) data it covers. However, the possibility of changing the size of the SE and applicability functions used more dynamically should be considered in the future. Also, an output certainty useful for discarding unreliable output pixel values could be defined by considering the relationship between the numbers of valid and missing pixels covered by each structuring element.

The strength of the method lies in its generality: although designed for 3D profile data it can be applied to any type of image data. In this work, we have demonstrated its use on gray-scale and 3D profile data, but further extension into higher dimensions is theoretically straight-forward (but of course more computationally demanding).

REFERENCES