



Published in Image Processing On Line on 2017-01-14.
 Submitted on 2015-03-08, accepted on 2016-11-22.
 ISSN 2105-1232 © 2017 IPOL & the authors CC-BY-NC-SA
 This article is available online with supplementary materials,
 software, datasets and online demo at
<https://doi.org/10.5201/ipol.2017.174>

Cradle Removal in X-Ray Images of Panel Paintings

Gábor Fodor¹, Bruno Cornelis^{1,2}, Rujie Yin², Ann Doots³, Ingrid Daubechies²

¹ Department of Electronics and Informatics (ETRO), Vrije Universiteit Brussel, Belgium (gfodor@etro.vub.ac.be)

² Department of Mathematics, Duke University, NC, USA

³ Department of Mathematics, Vrije Universiteit Brussel, Belgium



This IPOL article is related to a companion publication in the SIAM Journal on Imaging Sciences:
 R. Yin, B. Cornelis, G. Fodor, D. Dunson, and I. Daubechies, "Removing Cradle Artifacts in X-Ray Images of Paintings". *SIAM Journal on Imaging Sciences*, vol. 9, no. 3, pp. 1247-1272, 2016. <http://dx.doi.org/10.1137/15M1053554>

Abstract

We address the problem of mitigating the visually displeasing effects of cradling in X-ray images of panel paintings. The proposed algorithm consists of three stages. In the first stage the location of the cradling is detected semi-automatically and the grayscale inconsistency, caused by the thickness of the cradling, is adjusted. In a second stage we use a blind source separation method to decompose the X-ray image into a so-called cartoon part and a texture part, where the latter contains mostly the wood grain from both the panel as well as the cradling. In the third and final stage the algorithm tries to learn the distinction between the texture patterns that originate from the cradling and those from other components such as the panel and/or the painting. The goal of the proposed research is to improve the readability of X-ray images of paintings for art experts.

Source Code

The source code and an online demonstration of the algorithm described in this article are accessible at the [IPOL web page of this article](#)¹.

Keywords: art investigation; source separation; cradle removal

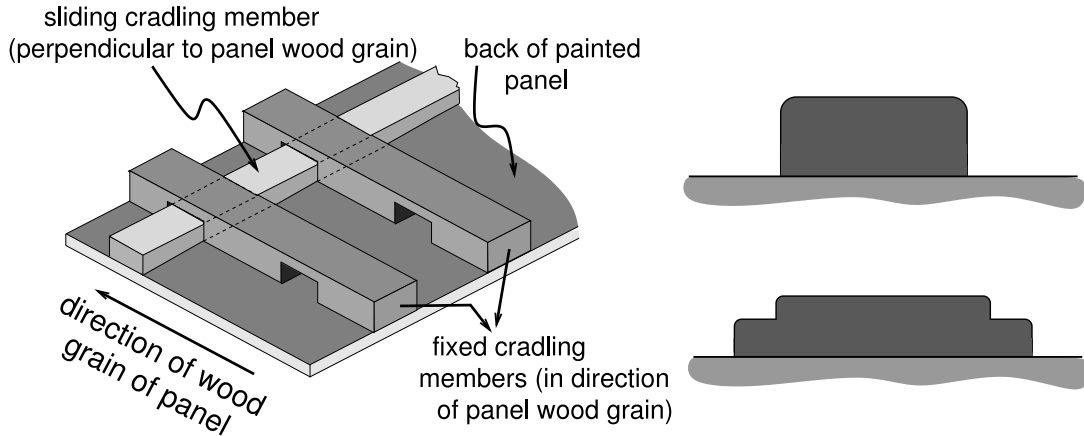


Figure 1: Left: Schematic illustrating the lattice of crossed perpendicular wooden members: members parallel to the panel wood grain are glued first; transverse members are slotted in. Right: Cradle member cross section profiles: regular rectangular profile (top) and irregular “T-shape” profile (bottom).

1 Introduction

Until about 1950 cradling was common practice for the restoration and preservation of paintings on wooden panel. This conservation procedure was often conducted on old master paintings from the 12th till the 17th century, when paintings on wood, and particularly on oak panel, were common. The procedure consists of permanently fixing hardwood slats or lattices to the back of a thinned panel, usually in the same direction as the wood grain of the panel, whilst the secondary perpendicular slats remain loose allowing for some degree of movement in the panel, which may warp, subject to changing levels in humidity. Figure 1 illustrates how cradles are attached to a panel painting together with two cross section profiles that are commonly used for the cradle members.

X-ray radiographs provide conservators and art historians with information related to the three-dimensional structure of a painting, not just its surface. The highly penetrating power of the X-rays provides information about structural aspects of the paintings and therefore make this investigative technique an important aid for modern-day conservators and art historians. However, due to their thickness, the cradling appears as a bright and highly visible grid, obscuring the viewing of the painting by art experts. Figure 2 depicts a 16th century painting of *Bernard van Orley*, currently in the collection of the North Carolina Museum of Art, entitled *The Pentecost* where the cradling is very prominent in the X-ray image of the painting. Hence we propose a semi-automatic procedure to remove these highly unwanted artifacts from X-ray images that is able to make the distinction between textures originating from different sources and subsequently isolate and remove the texture caused by the cradled parts, while leaving the texture from other components within the painting untouched. In practice, experienced conservators still do this manually, using Adobe PhotoshopTM, which is a time-consuming process, applied to only a limited number of paintings among the large museum collections. The algorithm presented here is an extension of our original cradle removal algorithm presented in [8].

Section 2 provides a detailed explanation of the different steps of our algorithm. Experimental results on a variety of paintings on panel with different cradling are contained in Section 3.

¹<https://doi.org/10.5201/ipol.2017.174>



Figure 2: Van Orley's The Pentecost (1525). Left: visual photograph. Right: X-ray image clearly showing the cradle structure.

2 Cradle Removal Algorithm

The cradle removal consists of two parts: a correction of the pixel intensity due to the thickness of the cradle member and a texture removal part that clears the X-ray from the wood grain of the cradle. The pipeline of this two-step removal process is presented in Figure 3 and Algorithm 1.

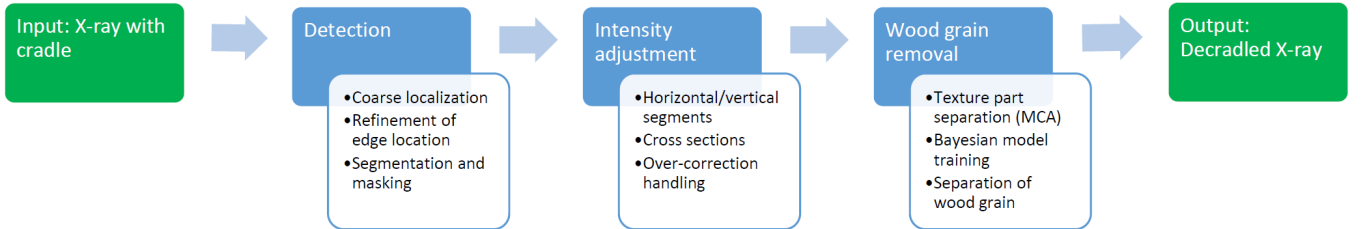


Figure 3: Cradle removal algorithm flowchart.

2.1 Cradle detection and intensity adjustment

This step attempts to correct the pixel intensities of cradle members, the most prominent artifact in X-rays of cradled paintings on panel. In order to achieve this, an accurate localization of the cradle segments is necessary, followed by a sequential removal of horizontal, vertical and cross sections.

2.1.1 Detection

The detection process is broken down into two steps. First, a coarse estimation is made of the cradle piece positions. In a second step, this estimation is refined by identifying the tilting/angle with the horizontal or vertical for each cradle segment. Since the detection of vertical and horizontal cradle pieces is identical, the following paragraphs will only focus on the detection of vertical cradle members. The same operations are applied for the detection of horizontal cradle pieces but on a rotated image by 90° .

Algorithm 1: Overview of the cradle removal algorithm

Data: Grayscale X-ray image I with cradle
Result: Grayscale X-ray image without cradle

// Step 1: Cradle detection and intensity adjustment
 $Mask \leftarrow cradleDetection(I)$; // Coarse estimation of cradle pieces - Section 2.1
 $Mask \leftarrow refineCradleLocation(I, Mask)$; // Refinement of cradle position - Section 2.1

for all horizontal/vertical segments $\in Mask$ **do**
 $(\mathcal{I}_p, \mathcal{I}_c) \leftarrow samplePixelPairs(I, Mask)$; // Section 2.1.1
 $(p_1, p_2) \leftarrow fitMultiplicativeModel(\mathcal{I}_p, \mathcal{I}_c)$; // Solution to Equation (3)
 if outliers(p_1, p_2); // If the regression is mis-fitted
 then
 $p_1 \leftarrow fitAdditiveModel(\mathcal{I}_p, \mathcal{I}_c)$; // Solution to Equation (4)
 $p_2 \leftarrow 1$;
 end
 $I \leftarrow intensityAdjusment(I, Mask, p_1, p_2)$;
end

for all cross sections $\in Mask$ **do**
 $(p_1, p_2) \leftarrow interpolateFromJointSegments(I, Mask)$; // Solution to Equation (6)
 $I \leftarrow intensityAdjusment(I, Mask, p_1, p_2)$;
 $I \leftarrow handleOverCorrection(I, Mask)$;
end

// Step 2: Wood Grain separation
 $(I^{text}, I^{cart}) \leftarrow MCA(I)$; // MCA algorithm (Algorithm 2)

for all cradle pieces $\in Mask$ **do**
 $(y^{cradle}, y^{non-cradle}) \leftarrow sampleShearletCoefficients()$;
 $(\Gamma, \Lambda) \leftarrow gibbsSampling(y^{cradle}, y^{non-cradle})$; // Train the sparse dictionaries, equations (11), (12)
 $I \leftarrow woodGrainSeparation(I, \Lambda, \Gamma)$; // Separation based on learnt dictionaries
end

return I ;

Coarse estimation: For the detection of vertical cradle members, a horizontal gradient operation is performed using a convolution operator with the kernel $[-1, -1, \dots, -1, 1, \dots, 1, 1]^T$ of size 40, determined heuristically. The width of the cradle pieces in our tested X-ray images range from 100 to 2000 pixels, and a kernel size of 40 pixels proved to be sufficient in all cases. The resulting pixel responses are summed column-wise. As cradle members always have higher pixel intensity values than the rest of the X-ray image, the cradle pieces can efficiently be localized by identifying minima-maxima pairs from the sum of pixel responses. The resulting signal is smoothened with a moving average filter with size $s = round(\max(\#rows, \#cols)/230)$, but restricted to a maximum of 10 and a minimum of 3. The smoothed signal is then normalized by extracting the mean of the signal. Local maxima/minima are considered to be valid candidates if they are sufficiently large in magnitude relative to the global maximum/minimum. In our implementation, we set the threshold to half the magnitude of the highest intensity peak. Candidate minima/maxima peaks are then paired up to determine cradle member locations. Let $P = \{p_1, p_2, \dots, p_N\}$ be the sorted list of maxima/minima peak column indices. For every minimum p_i in P , we pair it with the first maximum $p_{j \geq i}$ in the

list, with the resulting (p_i, p_j) pairs marking the location of the vertical cradle members. Special attention needs to be given to cases where a vertical cradle member is located on the very edge of the image, resulting in a missing minimum or maximum. Such cases arise when the leftmost candidate point corresponds to a maximum (i.e. the cradle piece is on the left edge of the X-ray and p_1 is a maxima peak) or the rightmost point corresponds to a minimum (i.e. the cradle piece is on the right edge of the X-ray and p_N is a minima peak). In such cases, we insert an auxiliary peak $p_0 = 0$ as an additional minimum or $p_{N+1} = \#cols$ as a new maximum, effectively marking the image border as the starting/finishing position of these special cradle pieces. An example of the whole process is given in Figure 4



Figure 4: Coarse estimation of cradle segment positions using a horizontal gradient operation. Left: original X-ray. Middle: result after vertical edge detection. Right: plot of edge detection coefficients, summed along the columns and smoothed. Points in red mark the start/end of cradle pieces and gray marking their location.

In highly contrasted or noisy X-ray images, the automatic cradle member detection described above can return more or less cradle members than the actual amount. To address this issue, an alternate mode of detection was implemented. Namely, the method allows a user to specify the number of cradle members prior to detection. In this case, instead of fixing the threshold for peaks at half the global maximum/minimum (as it is done for the fully automatic method), minima/maxima pairs are selected iteratively based on their likeliness of marking the location of cradle members. The ‘cradle likeliness’ of a minimum/maximum pair is evaluated as the sum of absolute values of the peak magnitudes. At each iteration, the most likely minimum/maximum pair is found, added to the list of cradle pieces and removed from the list of peaks (corresponding to potential cradle member locations). This process is repeated until the desired number of peaks is reached. This method results in a more robust cradle member detection. However, in some rare cases it can occur that the actual cradle member positions are still erroneously detected.

Refinement step: The cradle member positions are refined further by identifying the tilting angle of the cradle members, as they are rarely perfectly horizontal or vertical. The most likely rotation angle can be determined by using the Radon transform on the gradient image. The algorithm calculates the energy of the gradient image along lines of different angles, where the energy is defined as the sum of the gradient filter responses along the lines at a given angle. Practically, we limit the set of angles to a $[-5, 5]$ degree range with increments of 0.1 degrees, corresponding to a total of 101 possible angles. The tilting angle of the cradle edge is then considered to be the one with the highest energy.

Segmentation and masking: In practice, the location of each cradle piece is stored in a mask, where each pixel (x, y) is labeled depending on whether it is part of a vertical or horizontal cradle

member or both. This provides a straightforward segmentation of cradle pieces into horizontal, vertical and cross-sections, as shown in the results in Figure 5.

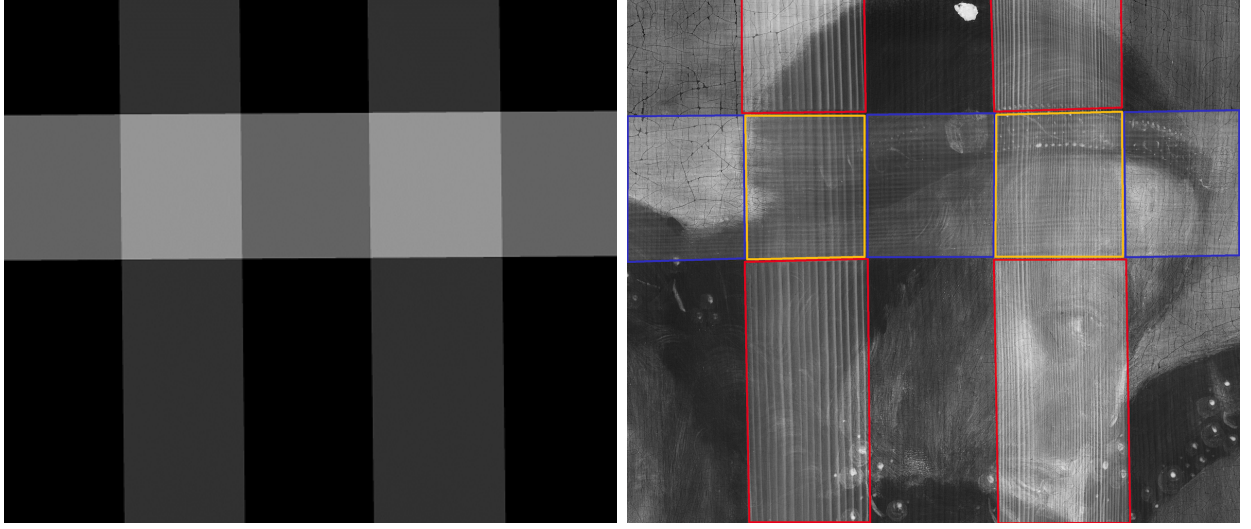


Figure 5: Left: mask marking cradle piece locations. Right: horizontal (blue), vertical (red) and cross-section (yellow) cradle segments, as derived by the algorithm.

2.1.2 Intensity adjustment

Once all the pixels that are part of cradle members are identified, their grayscale value can be adjusted. To this end, we define a model that is inspired by the physical properties of X-rays. The attenuation of X-rays as they pass through a material with absorption d can be expressed as

$$I = I_0 \exp(-\lambda d), \quad (1)$$

where I_0 is the initial X-ray beam intensity and λ is the decay factor of the X-ray. The absorption d depends on the thickness of the wood and the paint material that the X-ray penetrates through, thus it is spatially dependent on the cradle location and the paint composition. Since an X-ray image is brighter in places with higher attenuation, we assume that the X-ray image $\mathcal{I}$ is obtained by a linear transformation of I as $\mathcal{I} = AI + C$, where C is the saturation pixel value and $A < 0$ is a negative factor.

Suppose $(\mathcal{I}_c, \mathcal{I}_p)$ to be a pair of pixel intensities associated with the same painting content, with and without cradle. It follows from Equation (1) that

$$C - \mathcal{I}_c = (C - \mathcal{I}_p)e^{-\lambda d_c} = (C - \mathcal{I}_p)C_0(d_c), \quad (2)$$

where d_c is the cradle absorption and $C_0(d_c) = e^{-\lambda d_c} < 1$ is the attenuation function. Rewriting Equation (2), we have the following linear model of $\mathcal{I}_c$ and $\mathcal{I}_p$

$$\mathcal{I}_p = \frac{1}{C_0} (\mathcal{I}_c - C) + C = \frac{1}{C_0} \mathcal{I}_c + C \left(1 - \frac{1}{C_0}\right) = p_1 \mathcal{I}_c + p_2. \quad (3)$$

In short, as derived from the physical properties of gamma radiation propagation through matter, the relationship between a cradled pixel and its de-cradled version is linear and should be unique to all pixels where the thickness of the cradle is the same.

By sampling cradled and non-cradled pixels around the edges of each cradle piece, we are aiming for an accurate estimation of parameters (p_1, p_2) to apply the right correction for each cradle segment.

By calculating the regression parameters independently for each segment, as opposed to calculating a single model globally, we are obtaining better visual results. A single, global model would assume the thickness of the wood to be the same for all cradle pieces, which is rarely the case. A global model is also more vulnerable to artifacts in the X-ray, such as nails, metallic parts, or just in general overexposed/underexposed parts of the X-ray that are not part of the painting or cradle, but introduce errors to the regression. By fitting a separate set of regression parameters for each cradle segment, a better model can be fitted for each of these segments and all possible errors are kept local.

Note that most of the time, a misfitted regression model can be detected and partially corrected for. From Equation (3), we expect the constant term p_2 to be negative (cradled pixels have higher intensity than non-cradled ones) and that the multiplicative term p_1 is within a reasonable range. Based on experimental evaluation, we found that the correct values for p_1 lie within the range of $0.9 - 1.2$. When this is not the case we are confident that (p_1, p_2) were wrongly fitted and we revert to a more simple additive model of the form

$$\mathcal{I}_p = \mathcal{I}_c + p, \quad (4)$$

where p , similarly to the multiplicative model, is derived from the cradled/non-cradled sample pairs.

While there is usually a sufficient number of sample pairs to accurately estimate the regression parameters for horizontal and vertical segments, the cross section segments are much more challenging due to the varying thickness of the wood and the low number of non-cradled samples in those particular areas. Hence, cross section segments are not fitted with their own local model, but instead, their correction is interpolated from neighboring horizontal/vertical segments, which will be explained in more detail later on in this section.

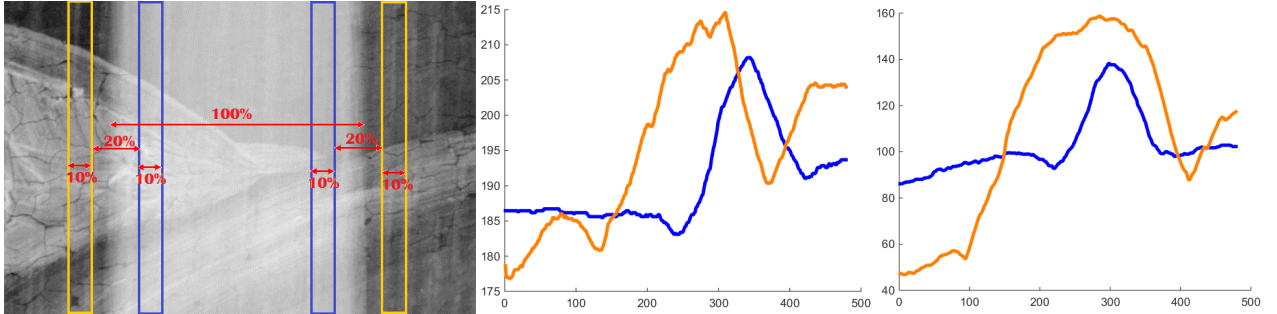


Figure 6: Left: cradle segment with out-of-focus (blurry) edges, marking the location for sampling cradled (blue) and non-cradled (orange) pixels. Middle: plot of paired cradle (blue)/non-cradle (orange) samples from the left edge of the cradle segment. Right: plot of paired cradle (blue)/non-cradle (orange) samples from the right edge of the cradle segment.

Correcting horizontal/vertical segments: Cradle/non-cradle pixel pairs are sampled on both sides of the edge of cradle pieces as the proximity of these pixels makes it more likely that they represent the same paint material. On the other hand, in order to correctly estimate grayscale correction, sample pairs need to be sufficiently far from the edges of the cradle as the wood tends to be thinner there and the X-ray might suffer from parallax. This also means that edge sections need to be corrected separately from the middle regions of a cradle. Sample pairs are taken at a distance from the cradle edge, corresponding to 10% of the cradle width, in order to avoid sampling from badly exposed edge regions of the cradle. We compute the median of the samples in 10% cradle width segments, as this operation is less sensitive to outliers than the sample mean. Figure 6 shows an example of the sampling location and the obtained sample pairs. We use the cradle/non-cradle sample pairs obtained this way to approximate the multiplicative model regression parameters (p_1, p_2) from Equation (3), or, if this is not possible, we revert back to a simplified additive model as

given by Equation (4). For the example in Figure 6, both the multiplicative and additive models are very similar (Figure 7), both of them being prone to the noise in the dataset, resulting in the blurry artifact after removal.

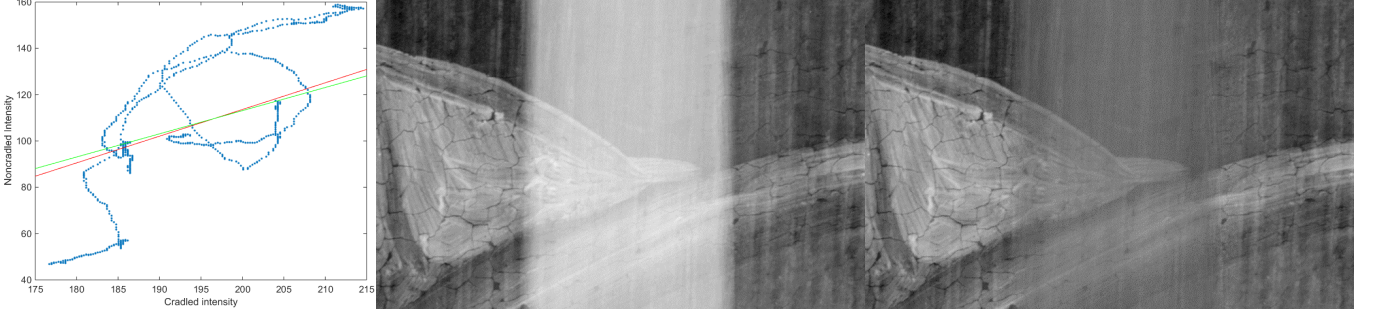


Figure 7: Fitting of the model. Left: cradled/non-cradled sample pair cloud from Figure 6, with multiplicative model (red) and additive model (green). Middle: original X-ray patch before correction. Right: corrected patch using multiplicative model.

Correcting segment edges: As discussed earlier, the corrective models described in Equation (3) and Equation (4) no longer hold for pixels that are part of the edges of the cradle segment and these areas need to be treated separately. To solve this problem, the algorithm performs an estimation of the edge profile of the cradle segment. Pixels along the cradle edges within an area of $\pm 10\%$ of the cradle width are summed along the tilting direction of the segment, normalized by the number of pixels and the resulting intensity distribution, as presented in Figure 8, is then used to correct pixels around the edges. Let I_{inner} be the intensity of the innermost pixels of the edge intensity distribution, and I_{outer} the intensity of the outermost pixels. We can use the derived corrective model for I_{inner} , as those pixels are sufficiently far from the edges, and no correction is necessary for the outermost pixel intensity I_{outer} , as it contains no cradle. For all other pixel positions, we use a linear interpolation of the form

$$\tilde{I}_{edge} = \frac{I_{edge} - I_{outer}}{I_{inner} - I_{outer}}(p_1 I_{inner} + p_2), \quad (5)$$

where I_{edge} is the normalized pixel intensity as given by the edge profile and $\tilde{I}_{edge}$ is the corrected value.

Note that this approach does not correct based on the individual pixel intensities around the edges, but rather on the general shape of the cradle edge. We found this approach to be less prone to outliers and less likely to introduce artifacts.

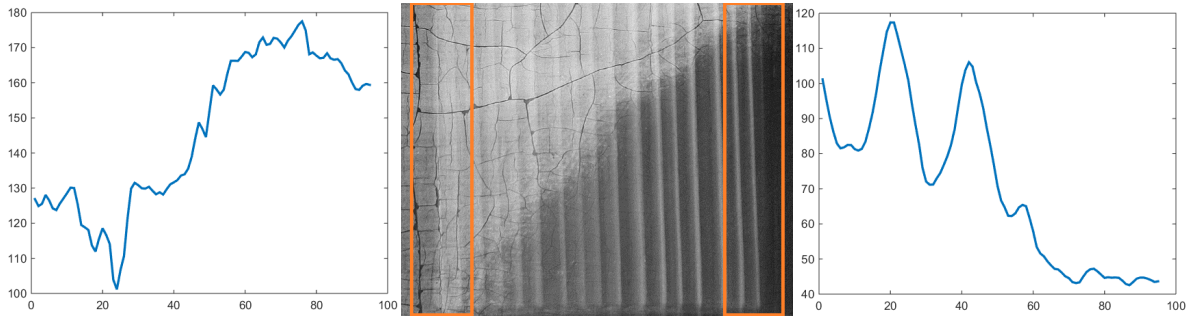


Figure 8: Edge profile estimation on a cradle segment. Left: left edge profile. Middle: X-ray of cradle segment, with zones used for edge profile estimation marked in orange. Right: right edge profile.

Correcting cross section: As mentioned earlier, sampling cradled/non-cradled pixels for cross-sections is difficult. Instead of fitting their own regression models, we will use the multiplicative models of the joint horizontal/vertical cradle segments instead. For each pixel of the cross section, we correct its intensity by a weighted average of the neighboring cradle segment models, with the weights determined by the distance from the segments. More specifically, let (p_i^1, p_i^2) be the regression parameters for the multiplicative model of some neighboring segment i and $I(x, y)$ be the intensity of a pixel at position (x, y) in the cross-section, then the corrected intensity $\tilde{I}(x, y)$ is defined as

$$\tilde{I}(x, y) = \frac{\sum_i w_i(x, y)(p_i^1 I(x, y) + p_i^2)}{\sum_i w_i(x, y)} \quad (6)$$

where $w_i(x, y) = 1/(1 + d_i(x, y))$ with $d_i(x, y)$ being the ℓ_1 -norm distance between point (x, y) and the closest pixel to it in segment i .

Handling over-corrections: Sometimes there is a small gap between the sliding and fixed cradle members. These gaps manifest themselves in X-ray images as thin elongated areas at the cross-sections with lower intensities relative to other cradled parts. When correcting these pixels assuming the same thickness as the rest of the cradle, a black segment is introduced, resulting in an unwanted artifact, as shown in Figure 9.

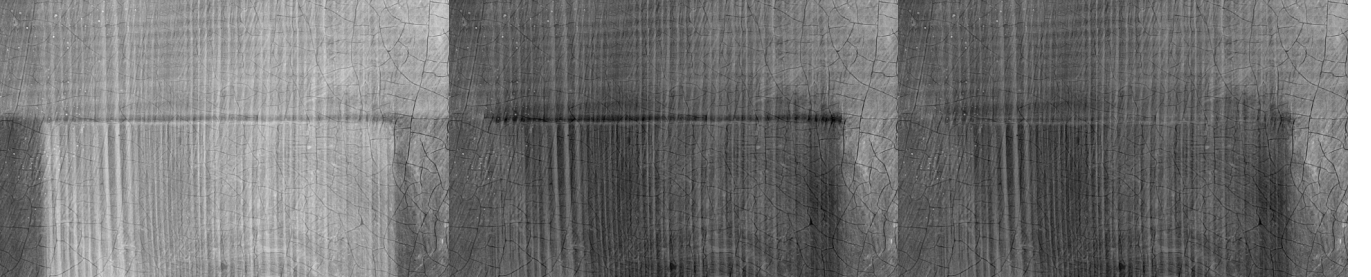


Figure 9: Overcorrection at cross-segments. Left: original X-ray at cross-section. Middle: over-corrected cross-section artifact. Right: cross-section after artifact removal.

To correct this artifact, we work on rectangular sub-selections covering the horizontal/vertical cradle segment intersections. First, we verify if the algorithm introduced any artifact by summing up pixel intensities along the intersection, confirming if there are any significant outliers/darker regions. We use the median pixel intensity times 0.7 as our threshold value to determine if there are any black segments and if so, how wide they are.

If a black segment is detected, the correction proceeds as follows. Given the selected segment containing the artifact (Figure 10a-left) and the removed cradle component for that segment (Figure 10a-right), the locations of the dark pixels due to overcompensation are determined. First, the segment is separated into a low-pass (Figure 10b-left) and high-pass component (Figure 10b-right) by using a top-hat transform. Frequency components, parallel to the dark edge, are zeroed out in the high-pass component and the difference between this filtered image and the original high-pass component (Figure 10c-left) is then used as input to a watershed algorithm to accurately mark the location of the overcompensated pixels (Figure 10c-right). The watershed algorithm starts from the lowest intensity location of the difference image (Figure 10c-left) and marks all neighboring pixels with intensities below 0. The final step calculates the optimal multiplicative factor for the cradle component of the marked pixels. The thickness of the wood varies greatly from X-ray to X-ray for these marked regions, hence the need for an adaptive correction factor that does not simply replace the dark pixels with the original X-ray pixel intensities, but a partial correction instead. The resulting image and cradle component is shown in Figure 10d.

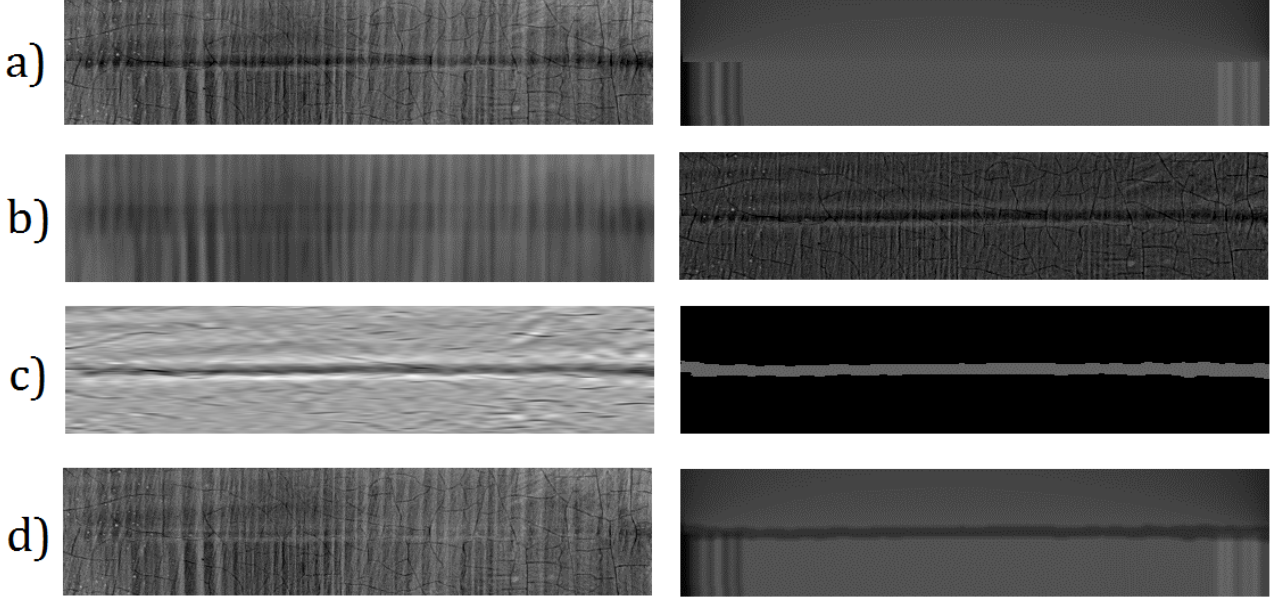


Figure 10: a) Left: image segment with dark edge from over-correction. Right: cradle component over the same segment. b) Left: low-pass component of image segment. Right: high-pass component of image segment. c) Left: difference image of high-pass and filtered high-pass. Right: position marking location of pixels to be corrected. d) Left: image segment after correction. Right: cradle component after correction.

2.2 Wood grain removal

While the first part of the algorithm addresses removing the grayscale difference caused by the cradle members, the texture caused by the wood-grain of the cradle is still present in the corrected image, as visible in Figure 9. These artifacts are removed in the second step of the algorithm that consists of two parts: a Morphological Component Analysis (MCA) [3] part separates the wood grain-like texture component from the rest of the image and a Sparse Bayesian Model based learning part further separates the texture into the wood grain of the cradle and the wood grain of the painting.

2.2.1 Morphological Component Analysis

Morphological Component Analysis (MCA) is a method that allows to separate features contained in an image provided that these features present different morphological properties. The algorithm decomposes an image into a texture and piecewise smooth (cartoon) part, as shown in Figure 11. MCA was successfully applied previously for the removal of wood grain texture to improve crack detection in X-ray images [5].

The wood grain texture in the cradled regions of X-rays is typically long, thin and periodic, a morphological characteristic different from the basic geometric shapes of the painting and the highly curved brush stroke details. Given this consideration, we choose two dictionaries such that one is optimal for sparse encoding of wood grain texture and the other is optimal for sparse encoding of painting content. For the wood grain components, we choose a redundant curvelet transform [7] while the painting image is represented using the dual-tree complex wavelet transform (DT-CWT) [6].

Given our choice for the dictionaries, the MCA algorithm solves the minimization problem formally defined as

$$(c_1^*, c_2^*) = \operatorname{argmin}_{c_1, c_2} \|c_1\|_1 + \|c_2\|_1 \quad (7)$$

$$\text{s.t. } I = I^{text} + I^{cart} = D_{curvelet}c_1 + D_{DT-CWT}c_2, \quad (8)$$

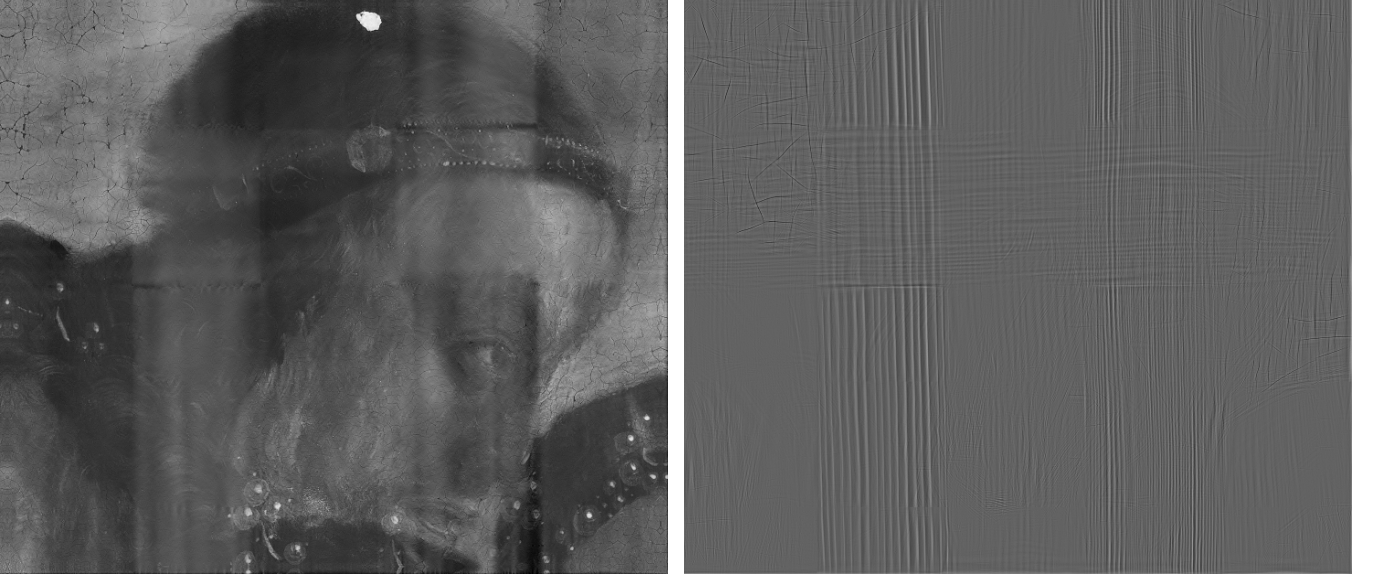


Figure 11: MCA decomposition on the X-ray from Figure 4. Left: cartoon part. Right: texture part.

where the coefficients (c_1^*, c_2^*) are the solutions to the constrained ℓ_1 -minimization which promotes the sparse encoding of the image I using the two selected dictionaries $D_{curvelet}$ and D_{DT-CWT} and its decomposition into the texture and image parts I^{text} and I^{cart} , with 7 levels of decomposition for the curvelet transform and 6 levels of decomposition for the DT-CWT.

Our implementation of the MCA algorithm uses an adaptive thresholding strategy, as proposed in [3]. The algorithm is applied on blocks of 512×512 pixels, with overlaps of 54 pixels between blocks. This is necessary since performing curvelet and dual-tree wavelet transformations on the full resolution images would be intractable due to memory restrictions. To avoid artifacts around the borders of the image, first a reflective border extension is applied to the image resulting from the first stage of the cradle removal. Let I be an image block of this image and I^{text} and I^{cart} the initially empty texture and cartoon parts of this block. The residual image I^{res} is defined as $I^{res} = I - I^{text} - I^{cart}$. In each iteration of the MCA algorithm the texture and cartoon parts are updated as

$$I_{i+1}^{text} = D_{curvelet}^{-1}(F_{text}(D_{curvelet}(I_i^{text} + I_i^{res}), \delta_i)) \quad (9)$$

$$I_{i+1}^{cart} = D_{DT-CWT}^{-1}(F_{cart}(D_{DT-CWT}(I_i^{cart} + I_i^{res}), \delta_i)), \quad (10)$$

where D_{DT-CWT} and $D_{curvelet}$ are the decomposition operators for the specified dictionaries, D_{DT-CWT}^{-1} and $D_{curvelet}^{-1}$ are the reconstruction operators and $F_{text}(\cdot, \delta)$ and $F_{cart}(\cdot, \delta)$ are thresholding operators with threshold δ for the texture and cartoon parts. Note that decomposition coefficients from D_{DT-CWT} and $D_{curvelet}$ must return normalized values for both dictionaries in order to compare coefficients across different dictionaries. This is achieved by dividing all decomposition coefficients by their respective sub-band energy after decomposing a common reference image (i.e. a 2D Dirac pulse) with the two dictionaries.

The filtering thresholding operators $F_{text}(\cdot, \delta)$ and $F_{cart}(\cdot, \delta)$ simply zero out all coefficients of norm smaller than δ . This forces morphological features that are strongly represented in one dictionary to be picked up by that respective dictionary, achieving the desired separation. Low frequency components have a similarly good sparse representation under any of the two dictionaries, resulting in these components being picked up by both dictionaries. Since we know that the wood grain is only present in the higher frequencies, we will force our implementation of MCA to automatically assign low-pass components to the cartoon part, by having the operator $F_{text}(\cdot, \lambda)$ to also zero out low-pass coefficients of the texture dictionary.

The threshold value δ , initially set to $\delta_0 = \min(\max(D_{\text{curvelet}}(I), \max(D_{DT-CWT}(I)))$, is updated at each iteration according to an exponential decay formula $\delta_{i+1} = \delta_i * \lambda$, with $\lambda = (\delta_0/10^{-7})^{1/(1-100)}$ a constant determined by the maximum number of iterations (100 in our case), and the target threshold to be reached by the 100th iteration (10^{-7} in our case). The algorithm stops when the maximum number of 100 iterations is reached or the residual error is below $5 * 10^{-4}$. The pseudo code of the MCA algorithm is given in Algorithm 2.

The resulting block-based MCA decomposition results are then put together into a single global texture and cartoon part, as shown in Figure 11. In order to avoid edge artifacts and ensure smooth transitions between blocks, the first half of the pixels in the overlapping regions are copied from the first block while the second half originate from the second block.

Algorithm 2: MCA algorithm pseudo code

Data: Result image I from Stage 1
Result: I^{text} and I^{cart} images resulting from separation

```

 $\delta \leftarrow \min(\max(D_{\text{curvelet}}(I)), \max(D_{DT-CWT}(I)))$  ; // Initial threshold for coefficients
 $\lambda \leftarrow (\delta/10^{-7})^{1/(1-100)}$  ; // Step size for threshold
 $I^{\text{text}} \leftarrow 0$ ;  $I^{\text{cart}} \leftarrow 0$ ;
 $I^{\text{res}} \leftarrow I$ ;
 $err \leftarrow \|I\|$ ; // Initial residual error set to image norm

for iter=1:100 do
     $C^{\text{text}} \leftarrow D_{\text{curvelet}}(I^{\text{text}} + I^{\text{res}})$  ; // Normalized curvelet decomposition
    foreach  $c \in C^{\text{text}}$  do
        if  $\|c\| < \delta$  OR  $c$  is low-pass ; // Set all low-pass and below threshold  $\lambda$ 
        then
             $c \leftarrow 0$  ; // coefficients to 0
        end
    end
     $I^{\text{text}} \leftarrow D_{\text{curvelet}}^{-1}(C^{\text{text}})$  ; // Reconstruct using filtered coefficients
     $C^{\text{cart}} \leftarrow D_{DT-CWT}(I^{\text{cart}} + I^{\text{res}})$  ; // Normalized DT-CWT decomposition
    foreach  $c \in C^{\text{cart}}$  do
        if  $\|c\| < \delta$  ; // Set all coefficients below threshold  $\lambda$  to 0
        then
             $c \leftarrow 0$ ;
        end
    end
     $I^{\text{cart}} \leftarrow D_{DT-CWT}^{-1}(C^{\text{cart}})$  ; // Reconstruct using filtered coefficients
     $\delta \leftarrow \delta * \lambda$  ; // Update threshold
     $I^{\text{res}} \leftarrow I - I^{\text{cart}} - I^{\text{text}}$ ;
     $err \leftarrow \|I^{\text{res}}\|$  ; // Update residual error
    if  $err < 5 * 10^{-4}$  ; // Check stopping criteria
    then
        return  $I^{\text{text}}, I^{\text{cart}}$ ;
    end
end
return  $I^{\text{text}}, I^{\text{cart}}$ ;

```

2.2.2 Bayesian Sparse Factor Model

The texture component resulting from the MCA decomposition contains a mixture of high-frequency, elongated objects, including wood grain, brush strokes, cracks and other elongated painting elements. In order to isolate the wood grain of the cradle, we employ a Bayesian sparse factor model that is able to learn a sparse representation of the wood grain only component and effectively separate it from the texture component.

We make the following assumptions for our theoretical framework:

1. The wood grain texture within cradle segments is homogenous;
2. The panel content is similar in the cradled and non-cradled regions;
3. The wood grain texture and the panel content are sufficiently different to be separable;

Assumption 1 holds reasonably well in practice, with the same piece of cradle typically having identical wood grain characteristics. As the wood grain might vary from cradle segment to cradle segment, the homogeneity assumption is restricted to within cradle segments only. Assumption 2 arises naturally as the panel painting content is independent of the cradling at the back of the panel. Finally, Assumption 3 is necessary in order to be able to separate wood grain from the rest of the painting content and is generally satisfied in all cases where the cradle wood grain can distinctively be identified visually.

We tackle this source separation problem by using a Bayesian dictionary learning approach. Assuming a sparse representation of both the wood grain of the panel and the cradle is possible, our aim is to learn the sparsifying dictionaries for both components. To make such an approach viable, we need an efficient sparse representation of the wood grain signal, for which we will be using the shearlet decomposition. The shearlet decomposition, conceptually related to wavelets, generates a multi-resolution representation of an image. For an L level shearlet decomposition, the linear shearlet transform operator $D_{shearlet} = [\Psi_0, \Psi_1, \dots, \Psi_L]$ is a combination of L curvelet transform operators. For all resolution levels l , the shearlet transform decomposes the image into θ_l equiangular directions so that $\Psi_l = [\Psi_l^0, \Psi_l^1, \dots, \Psi_l^{\theta_l}]$, where each $\Psi_l^k \in R^{n \times n}$ is generated by a direction k spatial convolutional operator at level l .

The wood grain texture of the cradle has a preferred direction, dictated by the wood fiber. All wood grain is contained in the horizontal and vertical direction of the high-frequency shearlet coefficients. Hence, we will train our separation algorithm on the reduced set of coefficients, corresponding to horizontal/vertical directions, as determined by the cone of influence of the shearlet transform. For a shearlet decomposition with 5 levels, we have $\{1, 4, 8, 16, 32\}$ directions per decomposition level, out of which we select the $\{0, 1, 3, 7, 15\}$ coefficients per resolution level for horizontal/vertical directions, resulting the total of 26 coefficients, forming the feature set for the separation algorithm.

The model: Given the previously described set of coefficients y_i for cradled or non-cradled pixel samples, our sparse factor model is formalized as

$$y_i = \Lambda \eta_i + \epsilon_i, \quad \text{if } y_i \text{ is not cradled;} \quad (11)$$

$$y_i = y_i^{cr} + y_i^{n_{cr}} = \Gamma \xi_i + \Lambda \eta_i + \epsilon_i, \quad \text{if } y_i \text{ is cradled;} \quad (12)$$

where Λ and Γ are the sparse dictionaries of the painting and cradle parts, η_i and ξ_i are the linear combinations of these dictionary elements for the signal y_i , ϵ_i is the error term. Under this model, the wood grain of cradled samples is given by $y_i^{cr} = \Gamma \xi_i$ and painting content by $y_i^{n_{cr}} = \Lambda \eta_i$. In order to guarantee that the algorithm only separates the wood grain, the error term ϵ_i is considered to also be part of the painting part.

Proposed Bayesian approach: Most of the literature assume existing bases or dictionaries such as wavelets, shearlets and DCT, but research has demonstrated the significance of learning a (usually) overcomplete dictionary matched to the signals of interest [1, 4], while simultaneously encouraging a sparse representation. These methods have demonstrated state-of-the-art performance in applications such as denoising, super resolution and inpainting. However, many of these algorithms portray restrictions, e.g. the noise variance is often assumed to be known and the size of the dictionary is usually set beforehand. To mitigate the aforementioned limitations, dictionary learning was posed as a factor-analysis problem where the factor loadings correspond to the dictionary atoms.

For the generic form of a latent factor model, as expressed in Equation (11), the dictionary matrix Λ is called the factor matrix and the elements in the coefficient vector η_i are called the factor scores. Equation (12) can also be viewed as a mixed factor model, with an extra factor matrix Γ . In the Bayesian approach, prior distributions need to be specified for model parameters to enforce model constraints. In particular, we use the sparse Bayesian infinite factor model formulation introduced in [2] for Equation (11), and adapt the formulation for Equation (12). Once Λ and Γ are learned based on cradled and non-cradled data pairs, η , ζ and ϵ can be estimated by post inference for Equation (12) on any mixed cradle signal y .

Following [2], we set up the priors for Equation (11) as

$$\begin{aligned} \forall i \in D_p, p \in \mathbb{Z}^+, k \in \{1, \dots, K\} \\ \eta_{i,p} &\sim \mathbf{N}(0, 1) \\ \epsilon_{i,k} &\sim \mathbf{N}(0, \sigma_k^2), \sigma_k^{-2} \sim \mathbf{Ga}(a_\sigma, b_\sigma) \\ \lambda_{k,p} \mid \phi_{k,p}^{(1)}, \tau_p^{(1)} &\sim \mathbf{N}(0, (\phi_{k,p}^{(1)} \tau_p^{(1)})^{-1}), \phi_{k,p}^{(1)} \sim \mathbf{Ga}(\nu/2, \nu/2) \\ \tau_p^{(1)} = \prod_{l=1}^p \delta_l^{(1)}, \delta_1^{(1)} &\sim \mathbf{Ga}(a_1, 1), \delta_l^{(1)} \sim \mathbf{Ga}(a_2, l), l \geq 2. \end{aligned}$$

Suppose the signal y_i is normalized to zero mean, unit variance. Then we may assume its factor η_i and its noise term ϵ_i are zero-mean Gaussian random vectors. An inverse gamma prior is used as the conjugate prior for the noise variance. A multiplicative gamma process shrinkage prior is used for the entries of Λ , such that Λ may have infinite columns with finite Frobenius norm [2]. The ℓ_2 -norm of columns of Λ decreases and the speed of decay is controlled by the global shrinkage prior $\tau_p^{(1)}$ as long as $a_2 > 2$ and the size of each entry is further controlled by the local shrinkage prior $\phi_{k,p}^{(1)}$. In practice, an infinite matrix Λ is approximated by a finite matrix by omitting columns with an ℓ_2 -norm. An advantage of such an “infinite” factor model is that we don’t need to know the number of factors, or equivalently the size of the dictionary, a priori.

We put the same priors on Λ, η_i and ϵ_i in Equation (12) and another set of multiplicative gamma process shrinkage priors on $\phi_{k,p}^{(2)}, \tau_p^{(2)}, \delta_l^{(2)}$ for Γ and a normal distribution prior for ξ_i ,

$$\xi_{i,p} \sim \mathbf{N}(\kappa_p, 1), \quad \kappa_p \sim \mathbf{N}(0, 1).$$

Gibbs sampler: To fit the Bayesian models in equations (11) and (12), we adopt the Gibbs sampler proposed in [2] to compute the posterior, cycling through model parameters within each iteration. The factor matrices Λ and Γ are kept finite in the algorithm by using an adaptive strategy for truncation of columns, presented in [2]. Let $\Lambda_{k_1^*}, \Gamma_{k_2^*}$ be the truncation of Λ, Γ to k_1^*, k_2^* columns respectively, then the Gibbs sampler update steps occur in the following order:

1. $\pi(\Lambda_{k_1^*} \mid Y_p, -)$, the posterior of the panel specific dictionary given the panel-only pixels Y_p and the other model parameters (represented by the shorthand notation $-$); Let λ_j^T be the

j^{th} row of $\Lambda_{k_1^*}$, $D_j^{(1)} = \text{diag}\{\phi_{j,1}^{(1)}\tau_1^{(1)}, \phi_{j,2}^{(1)}\tau_2^{(1)}, \dots, \phi_{j,k_1^*}^{(1)}\tau_{k_1^*}^{(1)}\}^{-1}$, $(y_p^{(j)})^T$ be the j^{th} row of Y_p and $\eta_p = [\eta_i]_{i \in \mathcal{D}_p}$, then the conditional conjugate posterior of each row λ_j^T is

$$\pi(\lambda_j^T | Y_p, -) \sim \mathbf{N}_{k_1^*} \left\{ \Sigma(\lambda_j^T)^{-1} \eta_p \sigma_j^{-2} y_p^{(j)}, \Sigma(\lambda_j^T)^{-1} \right\},$$

where $\Sigma(\lambda_j^T) = (D_j^{(1)})^{-1} + \sigma_j^{-2} \eta_p \eta_p^T$.

2. $\pi(\Gamma_{k_2^*} | Y_c, -)$, the posterior of the cradle specific dictionary; Let γ_j , $D_j^{(2)}$, $y_c^{(j)}$, η_c and ξ be defined in the same way as above, then the conditional conjugate posterior of γ_j is

$$\pi(\gamma_j | Y_c, -) \sim \mathbf{N}_{k_2^*} \left\{ \Sigma(\gamma_j)^{-1} \xi \sigma_j^{-2} (y_c^{(j)} - \eta_c^T \lambda_j), \Sigma(\gamma_j)^{-1} \right\},$$

where $\Sigma(\gamma_j) = (D_j^{(2)})^{-1} + \sigma_j^{-2} \xi \xi^T$.

3. The posterior of the inverse of the noise variance,

$$\pi(\sigma_k^{-2} | Y, -) \sim \mathbf{Ga} \left\{ a_\sigma + \frac{N_p + N_c}{2}, b_\sigma + \frac{1}{2} \mathbf{1}_{N_p}^T (y_p^{(k)} - \eta_p^T \lambda_k) + \frac{1}{2} \mathbf{1}_{N_c}^T (y_c^{(k)} - \eta_c^T \lambda_k - \xi_c^T \gamma_k) \right\}.$$

4. The posterior of the factor scores,

$$\begin{aligned} \pi(\eta_i | Y_p, -) &\sim \mathbf{N}_{k_1^*} \left\{ \Sigma(\eta)^{-1} \Lambda_{k_1^*}^T \Sigma^{-1} \mathbf{y}_i, \Sigma(\eta)^{-1} \right\}, & \text{for } Y_p \text{ non-cradled} \\ \pi(\eta_j | Y_c, -) &\sim \mathbf{N}_{k_1^*} \left\{ \Sigma(\eta)^{-1} \Lambda_{k_1^*}^T \Sigma^{-1} (\mathbf{y}_j - \Gamma \xi_j), \Sigma(\eta)^{-1} \right\}, & \text{for } Y_c \text{ cradled} \\ \pi(\xi_j | Y_c, -) &\sim \mathbf{N}_{k_2^*} \left\{ \Sigma(\xi)^{-1} (\Gamma_{k_2^*}^T \Sigma^{-1} (\mathbf{y}_j - \Lambda \eta_j) + \kappa), \Sigma(\xi)^{-1} \right\}, & \text{for } Y_c \text{ cradled} \end{aligned}$$

where $\Sigma = \text{diag}(\sigma_1^2, \sigma_2^2, \dots, \sigma_K^2)$, $\Sigma(\eta) = I_{k_1^*} + \Lambda_{k_1^*}^T \Sigma^{-1} \Lambda_{k_1^*}$ and $\Sigma(\xi) = I_{k_2^*} + \Gamma_{k_2^*}^T \Sigma^{-1} \Gamma_{k_2^*}$.

5. The posterior of the hyperparameters $\pi(\phi_{k,p}^{(1)} | -)$, $\pi(\phi_{k,p}^{(2)} | -)$, $\pi(\delta_l^{(1)} | -)$, $\pi(\delta_l^{(2)} | -)$ are defined as in [2].
6. The posterior of κ , the mean of $[\xi_j]_{j \in \mathcal{D}_c}$,

$$\pi(\kappa_j | -) \sim \mathbf{N}((1 + N_c)^{-1} \mathbf{1}_{N_c}^T \xi^{(j)}, (1 + N_c)^{-1}),$$

where $(\xi^{(j)})^T$ is the j^{th} row of ξ .

Implementation details: Similarly to the MCA algorithm, the shearlet transform is applied to the texture image I^{text} on blocks of 512×512 , with overlaps of 80 pixels. I^{text} still contains the reflective border of 80 pixels from the MCA part to guarantee that we are not introducing edge artifacts around the border of the image. Wood grain separation is performed one-by-one on each cradle segment due to memory constraints, as storing all feature vectors for all cradled and non-cradled pixels would be unfeasible on current hardware.

For each cradle segment, the overlapping blocks containing pixels of the target segment are identified, creating the set $Y^s = [y_0^s, y_1^s, \dots, y_{N_s}^s]$ of shearlet coefficient feature vectors y_i^s for cradle segment s . We will note by Y^0 the set of painting coefficient feature vectors, given by all pixels in the X-ray that do not belong to any cradle segments. The feature vectors Y^i are then normalized to zero mean, unit variance with respect to Y^0 . That is, first Y^0 is normalized, for each dimension of the shearlet coefficients feature vector, followed by the normalization of the cradle segment feature sets Y^s using the mean and variance of set Y^0 .

Depending on the resolution of the images, the number of cradled and non-cradled feature vectors can still be so large that the Bayesian dictionary learning on the full dataset would be intractable. This is solved by only running the dictionary learning on a small, randomly selected subset of feature vectors for both cradled and non-cradled segments. We use a reduced set $\bar{Y}^i$ of 5000 randomly selected features for the Gibbs sampler. The Gibbs sampler runs for 2400 iterations, with the first 2000 iterations being dropped as part of the burn-in (mixing) phase of the algorithm and the last 400 iteration parameters being saved out as the solutions for Equation (11) and Equation (12). For each iteration i of these last 400 iterations, the algorithm returns the values for $\{\Lambda_i, \Gamma_i, \kappa_i, \xi_i, \eta_i\}$. We consider each of these solutions equally good approximations of the ‘true’ solution, with the best estimator given by averaging out these solutions during post-inference.

Once the cradle segment model is trained, post-inference takes place. To improve computation speed, the separation model is only applied to a sub-sampled set of coefficients. These shearlet coefficients are obtained by sub-sampling only every 9th row/column, in a grid like fashion, resulting in a subset of the full set of coefficients that is about 80 times smaller. For all shearlet coefficients of this subset $Y^s = [y_0^s, y_1^s, \dots, y_{N_s}^s]$ and segment s , a separation of each coefficient vector y_i^s is determined for all 400 solutions, as specified by Equation (12)

$$y_{i,k}^s = y_{i,k}^{cr} + y_{i,k}^{ncr} = \Gamma \xi_k + \Lambda \eta_k + \epsilon_k, \quad (13)$$

where $1 \leq k \leq 400$ is the iteration number and $y_{i,k}^{cr}$ and $y_{i,k}^{ncr}$ are the cradle and non-cradle components of coefficient vector $y_{i,k}^s$ at iteration k . The separation coefficients

$$\bar{y}_{i,k}^{cr} = \sum_{k=1}^{400} \frac{y_{i,k}^{cr}}{400} \quad \bar{y}_{i,k}^{ncr} = \sum_{k=1}^{400} \frac{y_{i,k}^{ncr}}{400} \quad (14)$$

are given by averaging over all cradled and non-cradled separations.

Once the post-inference has been applied to the sub-sampled set, a KD-tree is built out of the sub-sampled coefficients in order to make fast nearest neighbor searches possible. Then for all the coefficients to be separated, the separation is given by a weighted average of its 5 nearest neighbors from the sub-sampled set and their separation coefficients. The distance metric used is the standard Euclidean distance over the 26 dimensional coefficient space.

In the end, all separation coefficients are de-normalized and the shearlet reconstruction of these coefficients, corresponding to the undesired wood grain of the cradle, is subtracted from the image.

3 Experimental Results

This section contains a number of representative examples of panel paintings on which the cradling was removed.

4 Conclusions

We proposed an automated algorithm to remove the grayscale inconsistency as well as the texture, caused by the presence of wooden cradling structures in the X-ray images of paintings. We tested our digital cradle removal algorithm against a wide range of X-ray images of paintings which we believe are representative for many use cases.

Focus was on robustness and stability, with our solution handling and adapting to even high noise levels, artifacts or different cradle deformations, giving perceptually acceptable results and improving the readability of cradled X-ray images significantly.

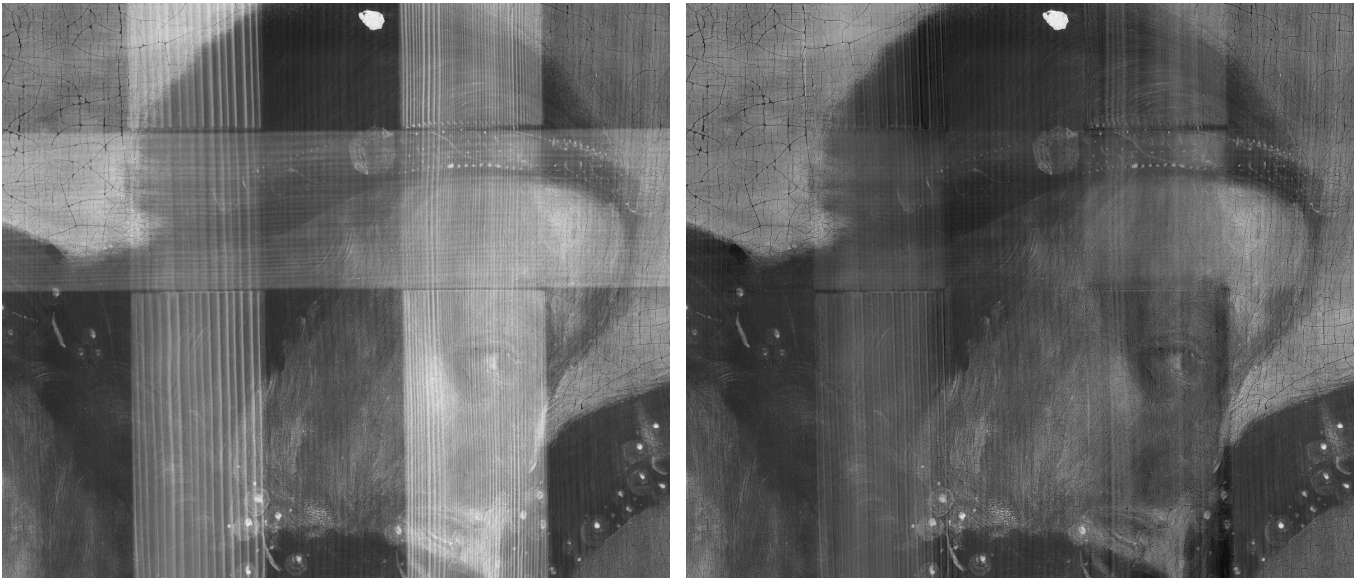


Figure 12: Ghent Altarpiece, panel of the Singing Angels (J. and H. van Eyck). Left: original. Right: decradled.

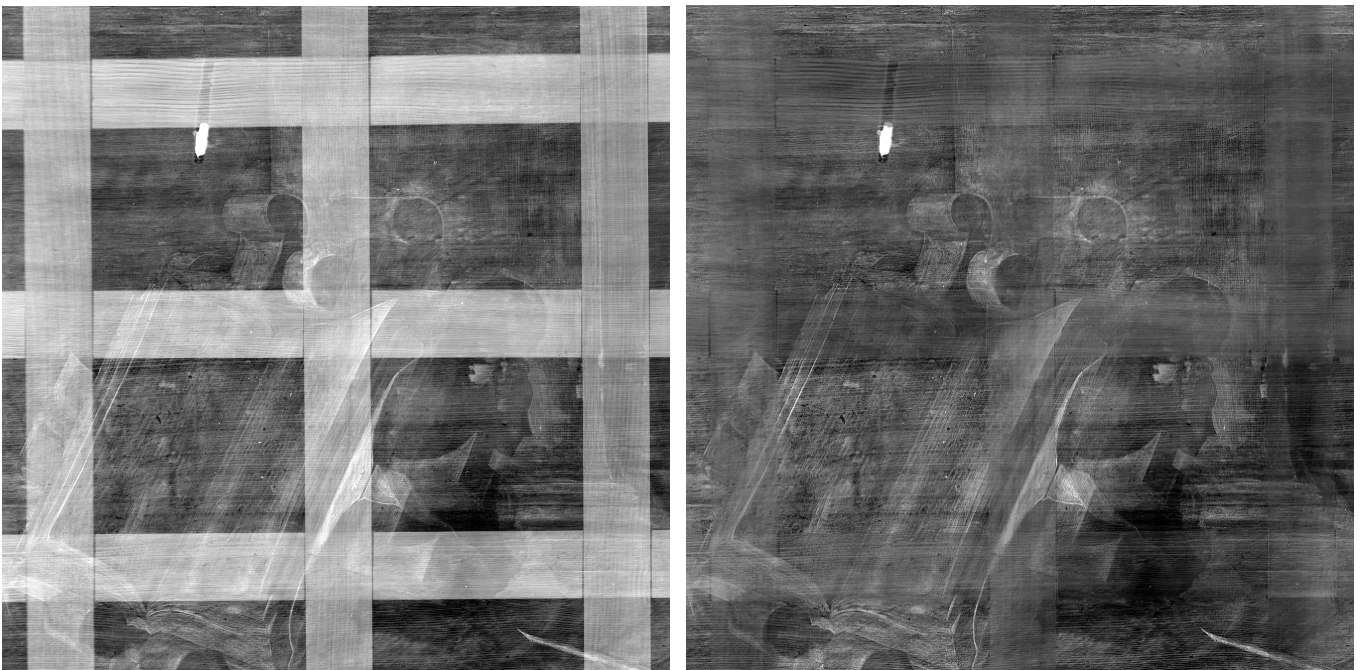


Figure 13: Saint Jerome in His Study (unknown). Left: original. Right: decradled.

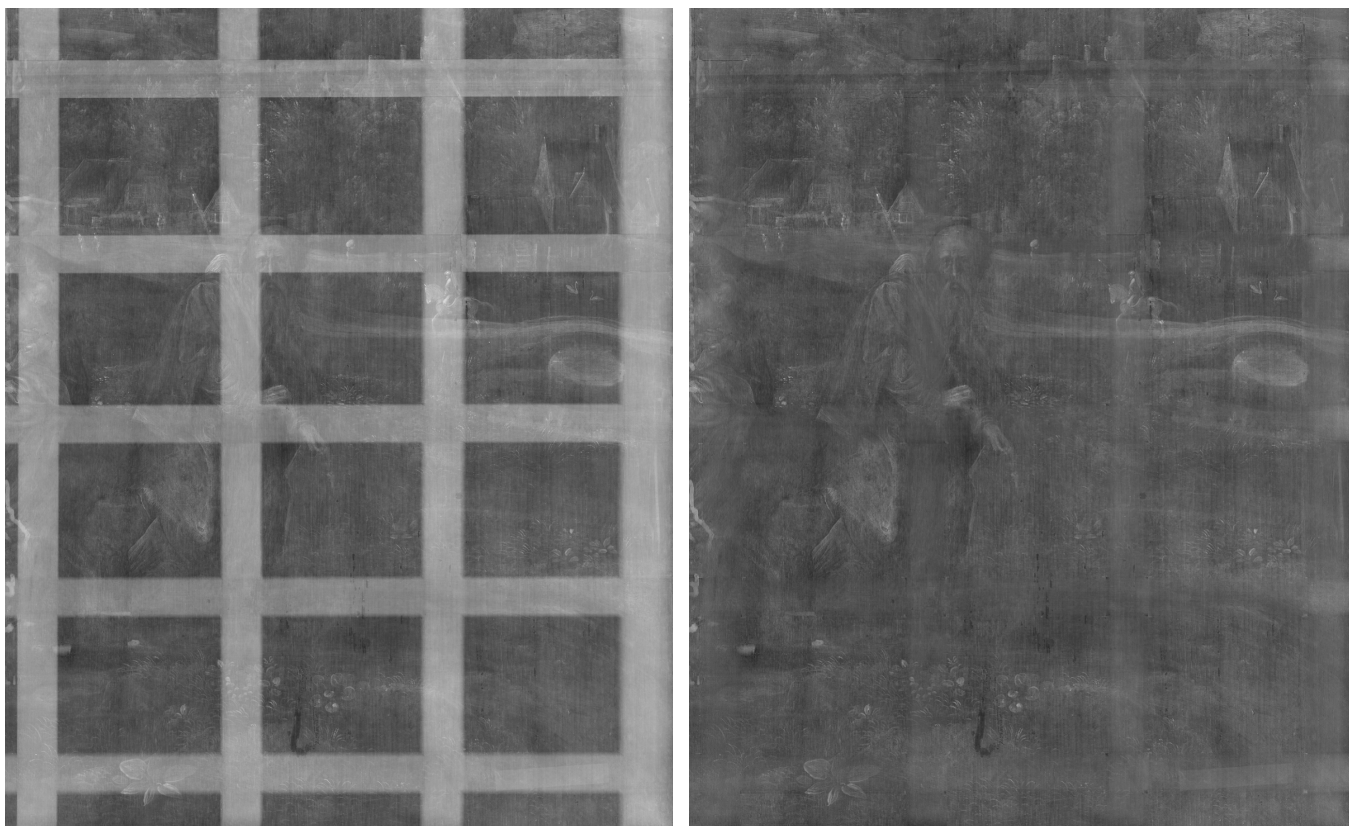


Figure 14: The Flight into Egypt (Master of the Female Half-Lengths). Left: original. Right: degraded.

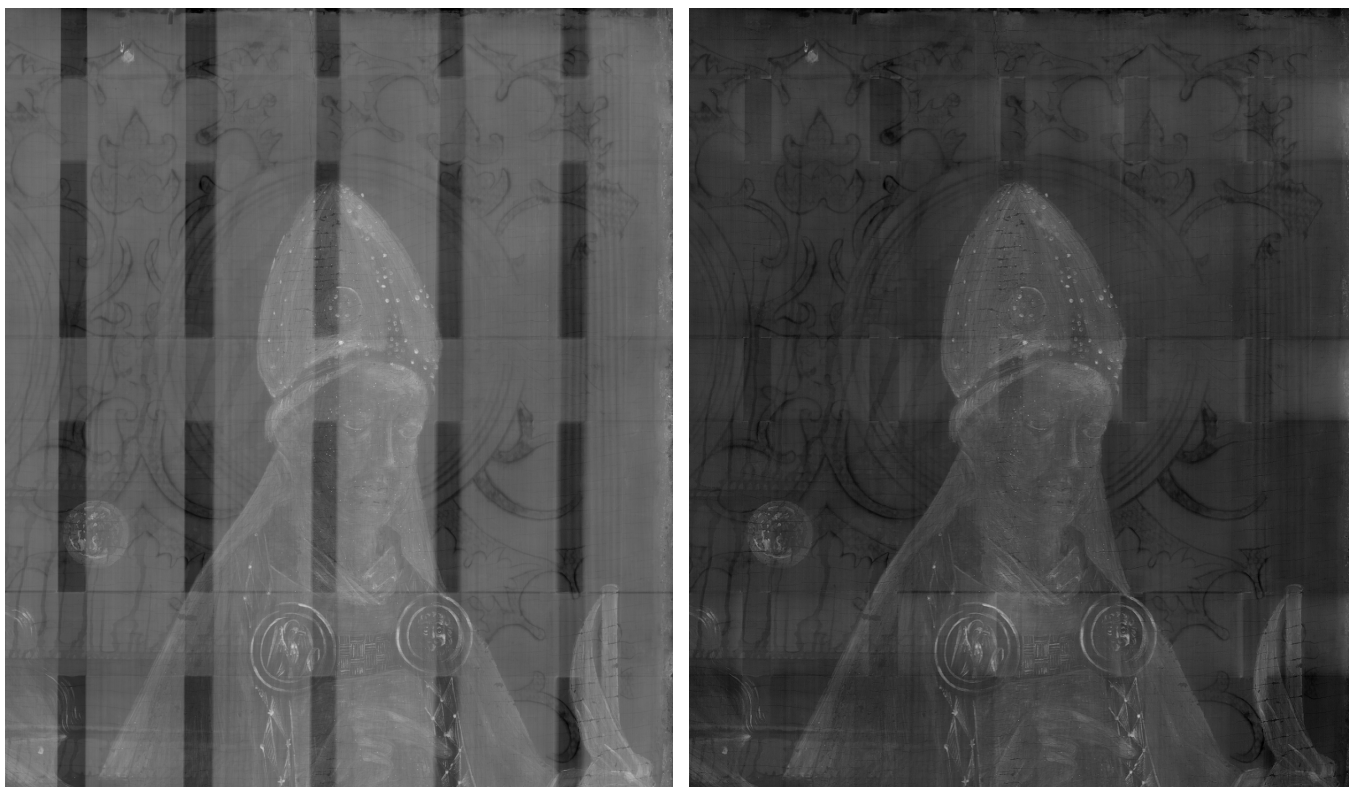


Figure 15: St. Barbara and St. Valentine with Caspar von Laubenberg and His Sons (Master of the Laubenberg Altarpiece). Left: original. Right: degraded.



Figure 16: The Dentist (Jan Miense Molenaer). Left: original. Right: decradled.

In general, the proposed algorithm, using either additive or multiplicative model, provides good or better results compared to the ones obtained manually by art conservators using Photoshop. The algorithm not only performs better than best current practices, but it also improves on it by removing the texture component that is caused by the cradling, which has never been done before. It has the significant added advantage that in many cases it can be applied fully automatically to the X-ray image while the manual removal of the cradling structure is a tedious and painstaking process.

Acknowledgements

We are very grateful to the North Carolina Museum of Art for making available to us, and giving us the permission to use, the X-rays of the paintings in their collection. Author Ingrid Daubechies gratefully acknowledges partial support by NSF-DMS grant 1320655 (supporting Rujie Yin) and a grant by the Kress Foundation (supporting Bruno Cornelis); author Gábor Fodor gratefully acknowledges support by a fellowship of the Belgian IWT (Innovatie door Wetenschap en Technologie) Foundation. We would like to thank Noelle Ocon for her expert advice and encouragement throughout this challenging project.

Image Credits

All images are courtesy of the North Carolina Museum of Arts (NCMA).

References

- [1] M. AHARON, M. ELAD, AND A. BRUCKSTEIN, *K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation*, Transactions on Signal Processing, 54 (2006), pp. 4311–4322. <http://dx.doi.org/10.1109/TSP.2006.881199>.
- [2] A. BHATTACHARYA AND D.B. DUNSON, *Sparse Bayesian infinite factor models*, Biometrika, 98 (2011), pp. 291–306. <https://doi.org/10.1093/biomet/asr013>.
- [3] J. BOBIN, J. L. STARCK, J. M. FADILI, Y. MOUDDEN, AND D. L. DONOHO, *Morphological Component Analysis: An Adaptive Thresholding Strategy*, IEEE Transactions on Image Processing, 16 (2007), pp. 2675–2681. <https://doi.org/10.1109/TIP.2007.907073>.

- [4] A.M. BRUCKSTEIN, D.L. DONOHO, AND M. ELAD, *From sparse solutions of systems of equations to sparse modeling of signals and images*, SIAM Review, 51 (2009), pp. 34–81. <http://dx.doi.org/10.1137/060657704>.
- [5] B. CORNELIS, Y. YANG, J.T. VOGELSTEIN, A. DOOMS, I. DAUBECHIES, AND D. DUNSON, *Bayesian crack detection in ultra high resolution multimodal images of paintings*, in International Conference on Digital Signal Processing, 2013. <https://doi.org/10.1109/icdsp.2013.6622710>.
- [6] N.G. KINGSBURY, *A dual-tree complex wavelet transform with improved orthogonality and symmetry properties*, in Proceedings of IEEE International Conference on Image Processing, 2000. <https://doi.org/10.1109/icip.2000.899397>.
- [7] J-L. STARCK, E.J. CANDLES, AND D.L. DONOHO, *The curvelet transform for image denoising*, IEEE Transactions on Image Processing, 11 (2002), pp. 670–684. <https://doi.org/10.1109/icip.2001.958937>.
- [8] R. YIN, D. DUNSON, B. CORNELIS, B. BROWN, N. OCON, AND I. DAUBECHIES, *Digital cradle removal in X-ray images of Art Paintings*, in IEEE International Conference on Image Processing (ICIP), 2014. <https://doi.org/10.1109/icip.2014.7025873>.