Document Enhancement using Visibility Detection

Netanel Kligler
Technion
Haifa, Israel
nati.bk@gmail.com

Sagi Katz
Technion
Haifa, Israel
sagi.katz@gmail.com

Ayellet Tal
Technion
Haifa, Israel
ayellet@ee.technion.ac.il

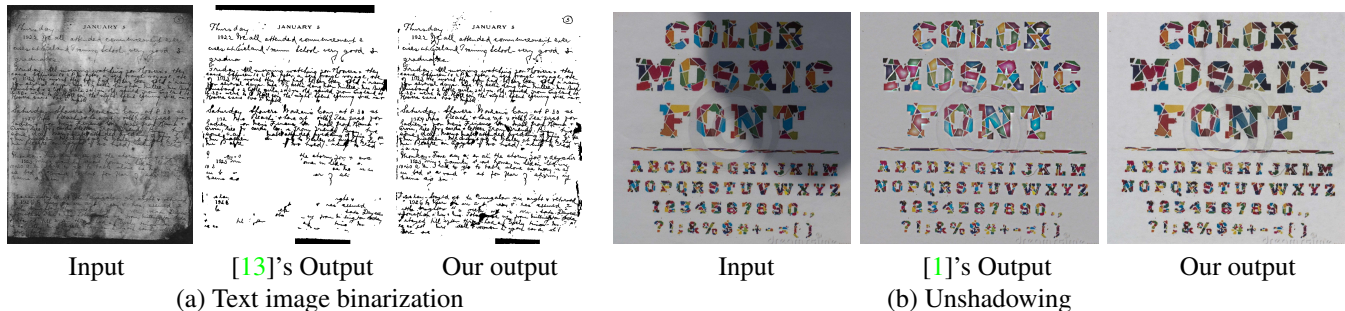| Input	[13]'s Output	Our output	Input	[1]'s Output	Our output

(a) Text image binarization (b) Unshadowing

Figure 1: **Document enhancement:** Combining visibility detection with state-of-the-art algorithms for document processing, improves performance. (a) Producing a binary image of a document, given a gray-level stained one, reveals the text behind the stain. (b) Removing shadows from a color image recovers the original colors.

Abstract

This paper re-visits classical problems in document enhancement. Rather than proposing a new algorithm for a specific problem, we introduce a novel general approach. The key idea is to modify any state-of-the-art algorithm, by providing it with new information (input), improving its own results. Interestingly, this information is based on a solution to a seemingly unrelated problem of visibility detection in $\mathbb{R}^3$. We show that a simple representation of an image as a 3D point cloud, gives visibility detection on this cloud a new interpretation. What does it mean for a point to be visible? Although this question has been widely studied within computer vision, it has always been assumed that the point set is a sampling of a real scene. We show that the answer to this question in our context reveals unique and useful information about the image. We demonstrate the benefit of this idea for document binarization and for unshadowing.

1. Introduction

Images of documents such as books, articles, and fliers frequently show up in our everyday lives. Many of these images are damaged, often for one of two reasons. The first is that the documents might contain artifacts such as stains, ink spills or simple degradations that happen over time [4, 13, 19, 20, 29] (Figure 1(a)). The second is that documents, when captured using cellphone cameras, become highly susceptible to non-uniform lighting caused by multiple light sources or by shadows, resulting from occlusion of the light sources [1, 22] (Figure 1(b)). Though these problems are inherently different and are thus traditionally handled by specialized algorithms, we propose a unified modification approach, applicable to both problems, which can be combined "on-top" of existing algorithms.

The key idea of our algorithm is that if we choose a suitable representation of pixels as points in $\mathbb{R}^3$ and then detect the subset of the points that are visible from a viewpoint (and/or occluding it), hidden information about the image is revealed. This information is used to create a new image, which when given as input to an existing algorithm, will improve its own results, while leaving the algorithm's general structure as usual.

Visibility has been defined in various dimensions and for different types of objects [3, 5, 6, 7, 8, 17, 18]. This paper focuses on a specific variant of it: Given a point set in $\mathbb{R}^3$ and a viewpoint, the goal is to determine the subset of points visible from the viewpoint. Since points cannot occlude each other (unless they accidentally fall on the same
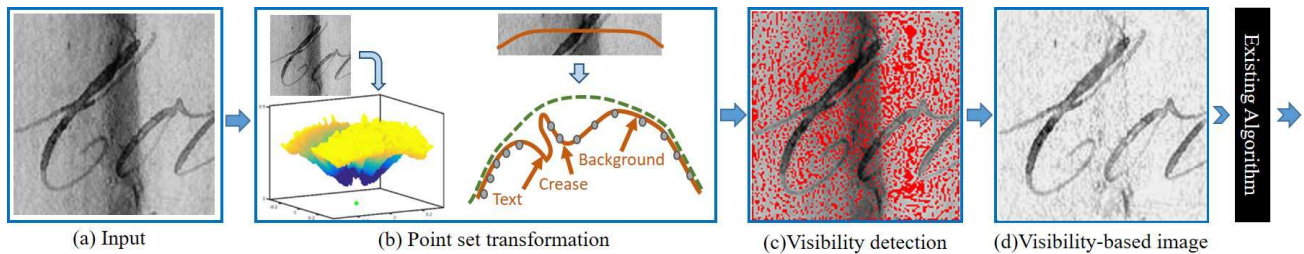
|(a) Input | (b) Point set transformation | (c)Visibility detection | (d)Visibility-based image |

Figure 2: **Algorithm outline:** The intensity image (a) is transformed to a 3D point set on a "globe", where the distance of a point from the globe's center (in green) is determined according to the intensity (b). Visibility detection operators detect which pixels have been transformed to visible/occluding points from/to the viewpoint (the globe's center). Visible/occluding pixels are marked red in (c). Notice that the letters hardly have any red points, whereas the background and the stain have many. This implies a strong visibility–foreground correlation. Using the information of (c), a visibility-based image is created (d). It maintains the text of (a), while making stains disappear. This image is used to alter existing algorithms.

ray from the viewpoint), the more precise goal is to determine the subset of points that would be visible, if the surface from which the points were sampled, was known.

Why does visibility information assist us? Take for instance Figure 1(a). The representation we propose will assign text pixels, which are dark on white background, to points in deep valleys. The background, which is constant and of high intensity, will turn to elevated plateaus. Stains on text, which have no distinct intensity characteristics, will be represented as structure-less, resembling "volcanic" terrains. Stains on background, which are darker local regions, will outcome in craters or shallow valleys upon the elevated plateaus. It turns out (and later rigorously explained why) that visibility detection operators manage to distinguish between deep valleys and "volcanic" terrains (text) to elevated plateaus, craters and shallow valleys (background).

We demonstrate the benefits of this idea in two tasks, as illustrated in Figure 1:

1. **Text image binarization**: Extracting text out of distorted or degraded documents, creating a binary image.
2. **Document unshadowing:** Readjusting the lighting of an image to produce an unshadowed image.

To summarize, the main contributions of this paper are: (1) a novel idea of introducing visibility detection into document processing. In this sense, this paper combines two fundamental and widely-explored problems in computer vision, which are seemingly unrelated; (2) creating a generic framework that realizes the above idea; (3) providing state-of-the-art results for two applications.

2. Algorithm

Given an image of a document (Figure 2(a)), which can be stained, shadowed or otherwise distorted, our goal is to produce another image (Figure 2(d)), which better distinguishes the foreground from the background. This image will benefit any algorithm that should process the document.

The key idea of our algorithm is to transform the given image into a set of points in 3D. The classification into visible (/occluding) and invisible (/non-occluding) points guides us of the background and foreground pixels. Our algorithm could be encapsulated into four stages:

Stage 1: Transform each pixel to a 3D point. To acquire initial intuition to this stage, one could imagine the image as if it was pasted onto a portion of the world globe. This is performed such that valleys and craters on the surface are created from pixels of low intensity relatively to their vicinity; ridges and mountains are created from high intensity pixels relatively to their vicinity; plateaus are created from areas of pixels with low varying intensity.

To realize it, the pixel's (x, y) coordinates are used to spread the points over a 3D sphere, and then low-intensity pixels are transformed closer to the sphere's center than high intensity pixels. Figure 2(b) shows both the full point cloud representing the whole image, as well as the transformation of a single (orange) curve through the image. Section 2.1 elaborates on this stage.

Stage 2: Detect the visible/occluding points from (/ to) the globe's center. As illustrated in Figure 2(c), red (occluding) dots exist on the background, including the stain, but not on the foreground letters. This is the key to the success of our approach. Section 2.2 explains how the globe analysis is related to a point being visible or occluding.

Stage 3: Create a visibility-based image by using the information of Stage 2 above, as shown in Figure 2(d).

Stage 4: Apply an existing algorithm, using the visibility-based image rather than the input image.

The computational complexity of our approach is $O(n \log n)$, for n points. Steps 1&3 are linear, whereas Step 2 is $O(n \log n)$ due to convex hull construction it applies.

We elaborate on Stages 1–2 in the following subsections. Stages 3–4 are task dependent and their details will be described in Sections 3– 4.

2.1. Image to point-set transformation (Stage 1)

Let p_i be a pixel within the image at location (x_i, y_i) and intensity level $I_i \in [0, 1]$. For color images the luminance channel of the $L^*a^*b^*$ colorspace is regarded as the pixel's intensity. The transformation first spreads the image onto a portion of the unit sphere's surface and then brings the pixels closer or further away according to their intensity.

In spreading, p_i is first translated to a scaled coordinate system relatively to the image coordinate system, having its origin located in the center of the image s.t.:

$$x'_i = \frac{x_i - \frac{width}{2}}{M}, \quad y'_i = \frac{y_i - \frac{height}{2}}{M}, \quad (1)$$

where M is the maximum image dimension. This guarantees that $(x'_i, y'_i) \in ([-0.5, 0.5] \times [-0.5, 0.5])$.

To set the height of a point on the globe, an initial height is defined for all pixels and p_i is then represented in polar coordinates (r_i, θ_i, ϕ_i). Then, each pixel is transformed to the unit sphere at $(1, \theta_i, \phi_i)$. Finally, r_i is set relatively to p_i's intensity using a linear transformation to I_i s.t.:

$$r_i = a_1 \cdot I_i + a_0. \quad (2)$$

In our earth analogy, the choice of a_1 and a_0 effects the nature of the terrain. When a_0 is high, even the deepest valleys (or areas of low intensity) remain far from the viewpoint. When setting a_1 to high values, the transformation will result in significant valleys and ridges even when the pixels that created them had similar intensity values. The choice of a_1 and a_0 differs between applications, but constant across all images of a dataset. Their values will be specified when elaborating on each of the applications.

2.2. Detecting visible/occluding points (Stage 2)

We start with some intuition, considering Figure 2(b). A text character is composed of pixels with low intensity, within a vicinity of many high intensity and slowly variating pixels—a typical background for text images. Our image transformation will cause the points that originated from this character to reside within a narrow valley. Meanwhile, the vertical crease creates a shallower and wider valley than the text pixels. Finally, the surrounding points, which originated from the background, will reside on plateaus.

We are basically looking for an operator that distinguishes between narrow valleys (characters) and all other cases. While this can be performed in various ways for surfaces, the problem is much more difficult for point sets. However, interestingly enough, operators that detect visible/occluding points can perform the task, with a little twist.

We first provide a short introduction to the problem and to these operators. Given a point set $P \subset \mathbb{R}^3$ and a viewpoint C, the goal is to find the subset of visible points. What does it mean for a point to be visible? If P is sampled from
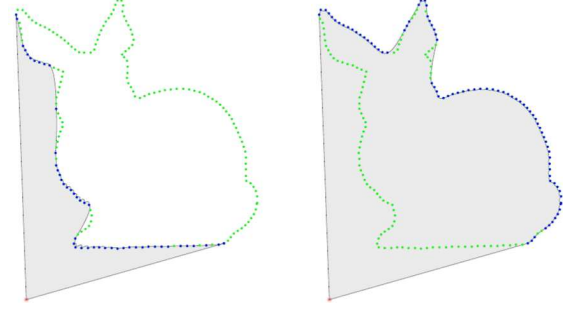
(a) HPR: from-point visibility   (b) TPO: to-point occlusion

Figure 3: **Visibility in 2D:** Blue points will be detected visible (/occluding) by the HPR (/TPO) operator, while green points will not. The red point is the viewpoint.

a surface S, then a point is considered visible if it would be visible to C on S (where visibility is well-defined).

The *Hidden Point Removal (HPR)* operator of [15, 16] determines the subset of visible points of P directly, sparing surface reconstruction. The operator consists of two steps (w.l.g, assume that C is at the origin):

1. **Point Transformation:** A function $F_\gamma : \mathbb{R}^D \to \mathbb{R}^D$ maps all $p_i \in P$ s.t.:

$$\widehat{p}_i = F_\gamma(p_i) = \frac{p_i}{\|p_i\|} \cdot \|p_i\|^\gamma. \quad (3)$$

Thus, F_γ moves p_i along the ray from the origin, where the exact distance depends on the parameter γ.

2. **Convex hull Construction:** The convex hull of $\widehat{P} \bigcup C$ is calculated, where $\widehat{P} = \{\widehat{p}_i \,|\, \widehat{p}_i = F_\gamma(p_i), \, p_i \in P\}$.

The main result of [15, 16] is that the points that reside on the above convex hull are the pre-images of the visible points.

A dual operator, the *Target Point Occluding (TPO)* [14], finds the subset of P that occludes C from outside observers positioned at ∞ (at all directions). In Equation 3, HPR requires $\gamma < 0$ whereas TPO requires $\gamma > 0$. Figure 3 presents the results of these operators for a point set in 2D.

Why are these operators relevant to our goal? To answer this, we note a limitation of the HPR (/TPO) operator [14, 15]. The key idea of our algorithm is to use this limitation to our benefit, as explained below.

The HPR operator will detect a visible point as such (/TPO & an occluding point), only if the curvature κ of the visible point is below (/above) a threshold:

$$\kappa < \frac{\gamma(1-\gamma)\sin(\beta)(\cos^2(\beta) - \gamma\sin^2(\beta))}{d(\gamma^2\sin^2(\beta) + \cos^2(\beta))^{\frac{3}{2}}}. \quad (4)$$

Here, γ is the parameter from Equation 3, β is the angle between the surface normal and the line of sight, and d is the distance between the surface and the viewpoint.
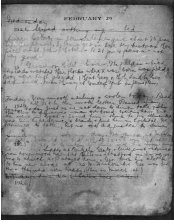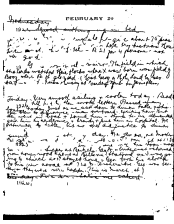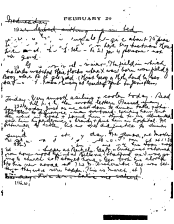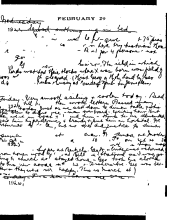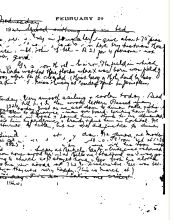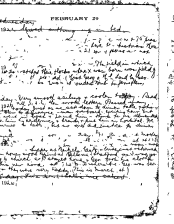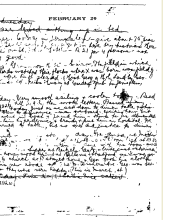
Input	Howe [13] Alg.2		Howe [13] Alg.3		Su et. al [29]	
Binarization Alg.	[13] Alg.2	ours	[13] Alg.3	ours	[29]	ours
Pf-measure	67.94%	75.32%	65.72%	75.58%	75.79%	85.16%

Figure 4: **Comparison:** The results of leading binarization algorithms are compared to their own results when replacing the input by our lowlights map, on an image from [2]. Thanks to our improved input, the text is better maintained in all the cases and the image borders are treated better. In the case of [29], this comes at the cost of an added stain.

In our globe analogy, the HPR operator will correctly detect visible points as long as they are in valleys or craters (the local surface is concave) or they are on non-steep ridges (convex with small curvature). Similarly, the TPO will detect occluding points as such as long as they are on ridges (local surface is convex) or when they are in a shallow valleys (concave with small absolute curvature). Note that in the original papers discussing the HPR and the TPO operators [14, 15], convexity and concavity are discussed from the viewpoint, whereas we address them with respect to the globe's surface, hence they are opposite.

This information is invaluable in image analysis. For instance, in the domain of text images, our image transformation causes the points that originated from a character to reside within a deep valley; accordingly, they are less likely to be detected as occluding by the *TPO* operator. Conversely, points that originated from the background are more likely to be detected as occluding. Both these observations could be noticed in Figure 2(c). The density of red (occluding) points within the characters is substantially lower than the background's red point density. The vertical crease, however, creates a shallower and wider valley than the text pixels. Hence, the TPO operator *does* detect the occluding points within it, which results in a "cleaner" image (d).

The next sections show the adaptation of our general approach to two text processing applications.

3. Application: Text Image Binarization

Given an image consisting of hand-written or printed text, the goal is to classify each pixel either as text (foreground) or as background. Thus, the output is a binary map that separates foreground and background pixels, as illustrated in Figure 1(a). This task is challenging due to artifacts in the image, such as stains, ink spills, creases, loss of text contrast etc.

Although the existing algorithms differ from each other, those considered state-of-the-art today share a general framework. Each pixel is first assigned some score, based on global and local cues, acquired from the intensity map. Later, it is "fine tuned" and manipulated using different assumptions regarding text images. The final stage is choosing a threshold, local or global, which separates between foreground and background pixels according to their scores [4, 19, 20, 29]. Alternatively, some algorithms minimize an energy function according to statistics representing text images, in order to separate the text pixels class from the background pixels class [13].

We propose to alter this framework by replacing the intensity map with our visibility-based ***Lowlights Map*** as input to the algorithm. We will show that the values of text pixels are closer to one another and are separated better from those of non-text pixels, making our lowlights map a more discriminate input. Therefore, even when the rest of the binarization algorithm is left untouched, its performance will improve. This holds regardless of the specific binarization algorithm we choose to alter; see Figure 4.

Figure 5 outlines our approach, which implements the framework presented in Section 2 for binarization. Similarly to the analysis done for Figure 2, given an intensity map, the point set transformation (Stage 1) causes text characters to create deep valleys, while the background and stains create ridges or shallow valleys and large plateau within the transformed point set. Thus, it suffices to use only the TPO operator for Stage 2, as the TPO operator is liable to misdetect points within deep valleys. Therefore, background pixels are detected by the TPO operator as occluding, whereas text pixels are not. Indeed, in Figure 5(b) the text characters contain almost no red points, differently from the background, the stain, and noisy pixels.

To process this information (Stage 3), a smoothed image is created using only the intensity values of the pixels that generated the occluding points. For the rest of the pixels, we apply the *natural-neighbor interpolation* [28], using only the occluding pixels. Hence, their values are approxi-

(a) Intensity Image

(b) Visibility Detection

(c) Smoothed Image

(d) lowlights map

Binarization Blackbox
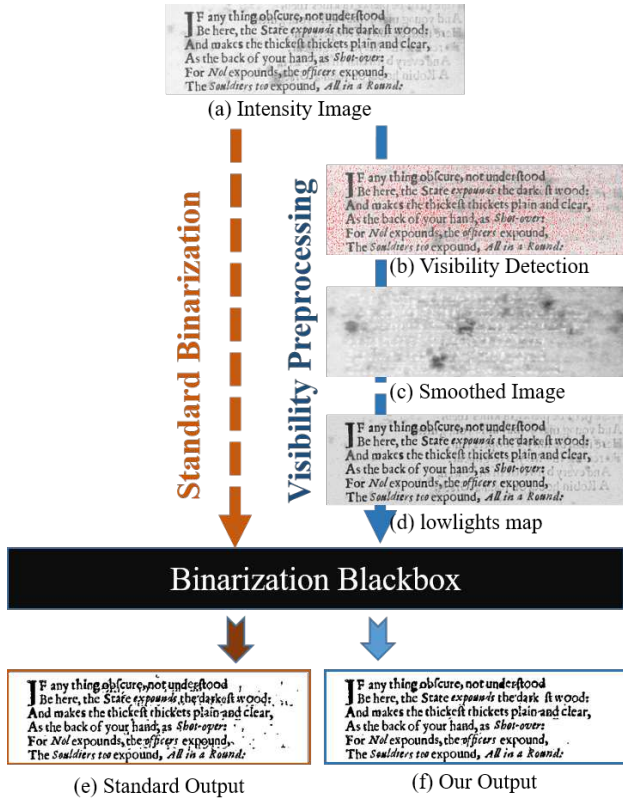
(e) Standard Output

(f) Our Output

Figure 5: **Outline—text image binarization:** The input (a) (from [24]) is replaced by our lowlights map (d), creating a much better result (f) than (e) [13]. To acquire (d), the occluding pixels are first detected (marked red in (b)) and a smoothed image (c) is created using only the intensity values of these pixels. Since text pixels are not likely to be detected as occluding, (c) contains only background and stains. Thus, subtracting (c) from (a) zeros out the intensity of the background, increasing the text-background contrast.

mated by their neighboring occluding (background) pixels. In Figure 5(c), the smoothed image contains the background of the input image, even if it is stained. Text characters, on the other hand, are not apparent.

Next, the smoothed image of Figure 5(c) is subtracted pixel-wise from the input image and subsequently normalized to the range of $[0, 1]$, resulting in a visibility-based lowlights map. Since the value of a background pixel in the original image and in the smoothed image are equal, its value in the lowlights map will be 1, as desired. Even if it is not detected occluding, the fact that it resides on a plateau or a ridge, means that its vicinity contains points that originated from pixels with similar intensity values. Consequently, the pixel will end up with a value approaching 1. On the other hand, if the pixel is a text pixel, then its 3D point will end up within a valley. This point, as well

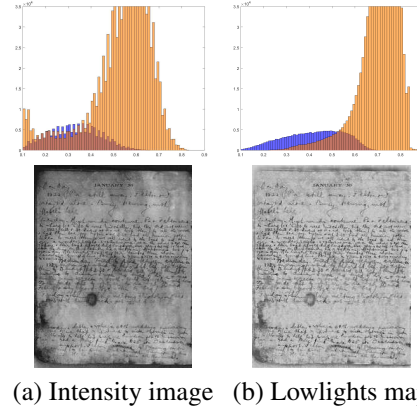

(a) Intensity image   (b) Lowlights map

Figure 6: **Histograms of intensity and lowlights maps:** Foreground pixels (blue) and Background pixels (orange) histograms are better separated in our lowlights map (b) than in the input intensity image (a). This makes the lowlights map more attractive for any binarization algorithm.

as points around it, are unlikely to be marked occluding. Thus, its interpolated value in the smoothed image will be much greater than its true intensity. Upon creation of the lowlights map, its value will be close to 0. This implies a stronger background-value correlation in Figure 5(d) than in 5(a), less noise in the background, and text strokes that contain lower internal intensity variance.

Figure 6 demonstrates the discriminative power of our approach. Using the ground truth, we plot the histograms of the background pixels (in orange) and the text pixels (in blue) both for the intensity map and for our lowlights map. The overlapping between the histograms of the two pixel types is much smaller in the lowlights map than that in the intensity map, and the separation is evident. Thus, the lowlights map is a better starting point for any foreground extraction algorithm.

Therefore, at Stage 4 of Section 2's framework, the intensity image is simply replaced by our lowlights map, as input to *any* text binarization algorithm. The improvement achieved is presented in Figure 5(e)+(f) and in Figure 4 for a variety of algorithms.

Results: Text image binarization has standard quantitative and qualitative visual evaluation, together with an agreed and widely-used set of benchmarks. We tested our approach on algorithms that performed best in the latest Document Image Binarization Contests [21, 27]. We measured their performance on the three common datasets in the field: (1) the hand written documents of Dibco 2009-2016 [10, 21, 23, 24, 25, 26, 27] (2) the printed documents of Dibco 2009, 2011, 2013 [10, 24, 26], and (3) the Bishop Bickley's Diary dataset [2]. The comparison was performed using the F-measure and the pseudo-FMeasure [21].

The average results for each of the datasets are pre-

Dataset	method	PF-measure			F-measure		
		[Howe '13]-2	[Howe '13]-3	[Su '13]	[Howe '13]-2	[Howe '13]-3	[Su '13]
DIBCO HW	original	93.24	93.72	90.28	91.23	92.42	88.87
	ours	**94.76**	**94.93**	**90.48**	**92.70**	**93.52**	**88.99**
DIBCO PR	original	91.36	92.55	92.01	90.57	91.89	90.38
	ours	**92.63**	**93.68**	**93.54**	**91.60**	**92.75**	**91.47**
Bickley Diary	original	66.34	64.72	69.55	68.37	67.92	71.46
	ours	**71.53**	**70.64**	**72.02**	**72.75**	**72.54**	**72.83**

Figure 7: **Binarization evaluation:** Performance evaluation of leading algorithms on benchmark datasets: average performance for DIBCO'09-DIBCO'16 ([10, 23, 24, 25, 26, 21, 27]) handwritten (top) & printed (middle) documents and Bickley Diary Dataset [2] (bottom). The best performing algorithm for each dataset is in bold font. With all measures on all datasets, our processing improves the results.

sented in Figure 7. Our results outperform the original results for all algorithms. In fact, it won the first place in the ICFHR2016 Handwritten Document Image Binarization Competition [27]. We note that for most of the images within the Dibco datasets, previous methods reach almost perfect scores. For such cases, visibility detection produces little improvement. Yet, visibility does matter when the input is challenging. Therefore, the improvement rates for the more difficult Bickley Dataset are more profound.

Figure 8 shows a couple of additional examples out of [24], where the results of the original best performing algorithms are improved. Note how the vertical crease (top) is detected as background and how the noisy region on the right-hand side of the bottom image is better treated. **Parameters:** We used fixed parameters, as follows. In Eq. 2 we set $a_1 = 1$, $a_0 = 0.05$. In Eq. 3, we set $\gamma = 0.01$ for the Dibco hand-written dataset, $\gamma = 0.005$ for the Dibco printed dataset and $\gamma = 0.001$ for the Bickley dataset.

4. Document Unshadowing

Images of documents such as letters, articles, and manuals are commonplace in our lives. Many of these images are captured using cellphone cameras, under sub-optimal conditions, such as multiple light sources and objects occluding light sources. The unshadowing task aims to "recover" a fully detailed document, as if the image was taken in a properly (uniform) lit setting, as demonstrated in Figure 1(b). Note that while binarization is a classification problem, unshadowing is a restoration problem [11, 31].

State-of-the-art algorithms [1, 22] perform this task in two stages. First, a *shadow mask* of the image is created, using global and local lighting properties of the image. Next, the mask is combined with the input image, to create the output. The differences between the algorithms occur mostly in the creation of the shadow mask. We propose to use our *visibility-based shadow mask*, leaving the general structure of the algorithms untouched.

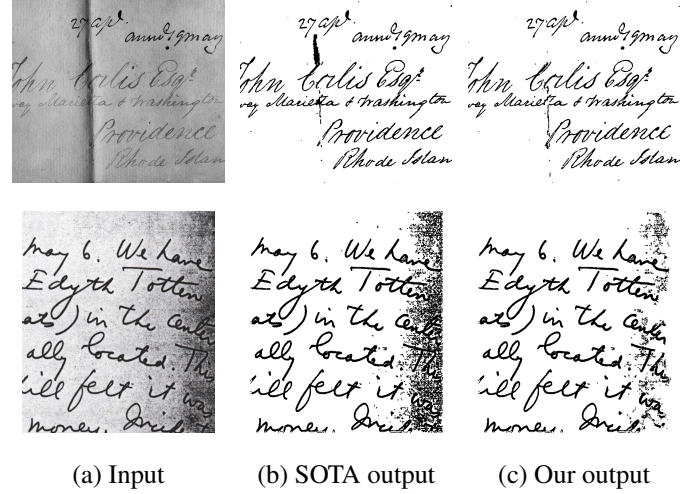| (a) Input	(b) SOTA output	(c) Our output

Figure 8: **Binarization results:** (a): Images out of [24]. The best-performing algorithm's output in (b) is improved by using our lowlights maps in (c). Top: [13] Alg. 3 is improved from 92.06% to 93.61%. Our algorithm manages to detect the vertical crease as background. Bottom: [29] is improved from 76.18% to 90.18%. Our algorithm enables better treatment of the noisy region on the right-hand side of the image. Scores are in PF-measure.

Before going into detail, we explain the intuition behind the shadow mask we create. Since our input is an image of a document, we may assume that the background should be locally nearly constant. Therefore, any small group of neighboring pixels with high intensity variation is likely to contain foreground pixels. We aim to detect pixels that are unlikely to be in highly-variating regions. This ensures that only background pixels are detected.

We take advantage of the visibility properties, which tell us how unique a small group of pixels is within a larger region. When transforming the image to a point set (Stage 1, Section 2), regions of absolute high curvature are created from pixels of high varying intensity. Conversely, points that are detected **both** visible (HPR) and occluding (TPO) are guaranteed to reside within regions of small curvature. Consequently, they are likely to be background pixels.

Unlike the previous application, the input here is a color image. We use the L*a*b* color space and process only the L* channel, normalized to $[0, 1]$. To detect the background pixels, we transform the image into a point set, apply both the HPR and the TPO operators and select pixels detected by both. Using only these pixels, we create our *visibility-based shadow mask*, which evaluates the lighting (L*) (or shadows) of the image, disregarding the data (characters). The intensity values of the detected pixels are set to their original intensity, which is purely the document's global lighting. For the rest of the pixels, we use *natural-*

(a) Input

(b) [1]'s shadow map (left) vs. ours (right)
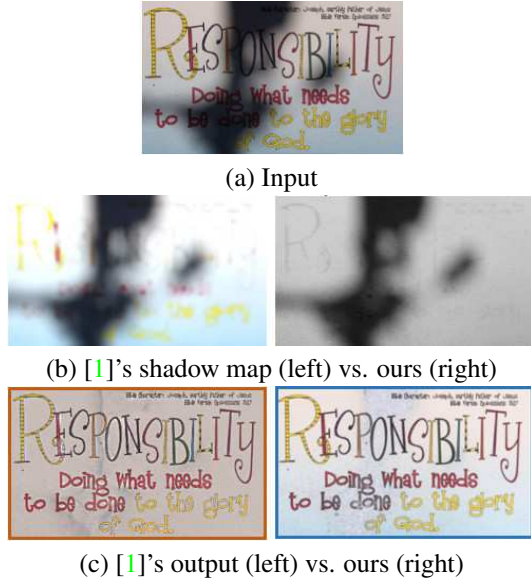

(c) [1]'s output (left) vs. ours (right)

Figure 9: **Shadow masks & unshadowing:** Visible pixels are used to interpolate a shadow mask (b), given an input color image (a). Our mask contains only lighting information, differently from [1]'s. Using the mask, an unshadowed image is produced (c). Because our shadow mask better separates the data from the shadow, the colors and the background lighting are better preserved than in [1].

(a) Input   (b) [1]'s output   (c) Our output

Figure 10: **Unshadowing results:** Details are better preserved using our approach, thanks to more accurate shadow masks. The images represent the 4 categories of the dataset.

neighbor interpolation [28], similarly to what was done for binarization. As can be seen in Figure 9(b), our mask is "clean", containing only the lighting / shadowing component, and no meaningful information is lost.

After our shadow mask is created, the rest of the algorithm follows [1], creating the unshadowed image. Briefly, the input intensity map is divided by the shadow mask pixelwise, turning the value of "visible" (background) pixels to 1 (having the same value in the input and in the shadow mask). As 1 represents full lightness, this results in an over-lit image. It is readjusted by multiplying all intensity values in the over-lit image by a global factor, which is determined by the color of the background. As a final step, a Median filter is applied, to handle artifacts created throughout the unshadowing process, as well as in the input itself.

As the input is a color image, so should the output be. Thus, our intensity map is combined with the original a* and b* channels, and white balancing is applied [9] to regions that were detected as shadowed. As shown in Figure 9(c), the background and the character colors are better preserved and the original colors are recovered.

Results: We evaluated the algorithm on a new and diverse benchmark of 381 artificially-shadowed images from four categories: handwritten documents, printed documents, posters, and fonts. Each category consists of 7-10 docu-

ments and each document is captured with 8-12 variations of shadow intensity, shape and location. It includes the dataset of [1], which contains 81 images created from 11 documents, all having small fonts, little color variance and gentle shadowing (i.e. input is similar to ground truth).

Figure 10 shows qualitative comparisons of typical images from the four categories. The most prominent difference is the color we are able to preserve, restoring a richer and more detailed unshadowed images.

To quantitatively evaluate the performance, it is measured in terms of Structure Similarity Index Measure (SSIM) [30], which is considered to be correlated with the quality perception of the human visual system [12], as well as with Mean Squared Error (MSE) and Peak Signal-to-Noise measure (PSNR) [12]. Table 11 presents the average results for each of the categories. High similarity between an unshadowed image to the ground truth document would result in high SSIM (approaching 1) and PSNR (approaching infinity), while MSE would approach 0. Scores of the input images are shown for reference.

In terms of SSIM and MSE, our results outperform [1] for all four categories. In terms of PSNR, our results are better for two categories and slightly worse than [1]'s for two. This can be explained by the good performance of [1] for mildly-shadowed images, resulting in high PSNR values. Due to the exponential nature of PSNR, successful un-

Category	Method	SSIM	MSE	PSNR
Handwritten Documents	**Input**	0.880	3.966%	18.011
	[Bako]	0.925	0.127%	**30.405**
	Ours	**0.935**	**0.126%**	29.912
Posters and Signs	**Input**	0.871	2.485%	16.218
	[Bako]	0.885	0.497%	23.999
	Ours	**0.914**	**0.180%**	**27.908**
Printed Documents	**Input**	0.925	1.532%	20.642
	[Bako]	0.940	0.148%	**31.422**
	Ours	**0.943**	**0.128%**	30.536
Fonts and Symbols	**Input**	0.883	2.644%	16.508
	[Bako]	0.891	0.618%	24.352
	Ours	**0.922**	**0.227%**	**27.325**

Figure 11: **Quantitative evaluation:** Comparison with [1] on a dataset comprised of 381 images indicates that for most measures and document classes our results outperform those of [1]'s. Input scores appear as a baseline.

shadowing causes the average score over the whole dataset to approach significantly higher values.

Figure 12 provides visual qualitative comparisons for real-world images obtained from the Internet (hence, there is no ground-truth to compare against). It can be seen that even for highly challenging images—both in shadowing and in color—our results are more eye-pleasing.

Parameters: We set γ values in Eq. 3 so as to create a detection rate of $10\tilde{\%}$ for the TPO and $3\tilde{\%}$ for the HPR. For Eq. 2, the point set transformation, we set $a_0 = 1$, $a_1 = 1$. For filtering, we used $window = 3$ for the median filter.

Limitation: Figure 13 is an example where the result of [1] is more similar to the ground truth than ours, since [1] is better tuned to small fonts and little color variance.

5. Conclusion

This paper has proposed a general approach for processing documents of various types. The idea is rather simple. Instead of measuring properties of pixels, such as uniqueness, varying intensity or relativity, we determine the visibility and/or occlusion of their representative 3D points. We show that simple and efficient ($O(n \log n)$) operators reveal properties of the image that are otherwise hidden.

We applied our framework for text image binarization and document unshadowing. We have demonstrated how visibility information helps create a cleaner and clearer input to existing algorithms and improve their own results. Since our modification can be applied to any algorithm, it will likely benefit future developments in the domain.

We intend to study the use of our concept for other prob-

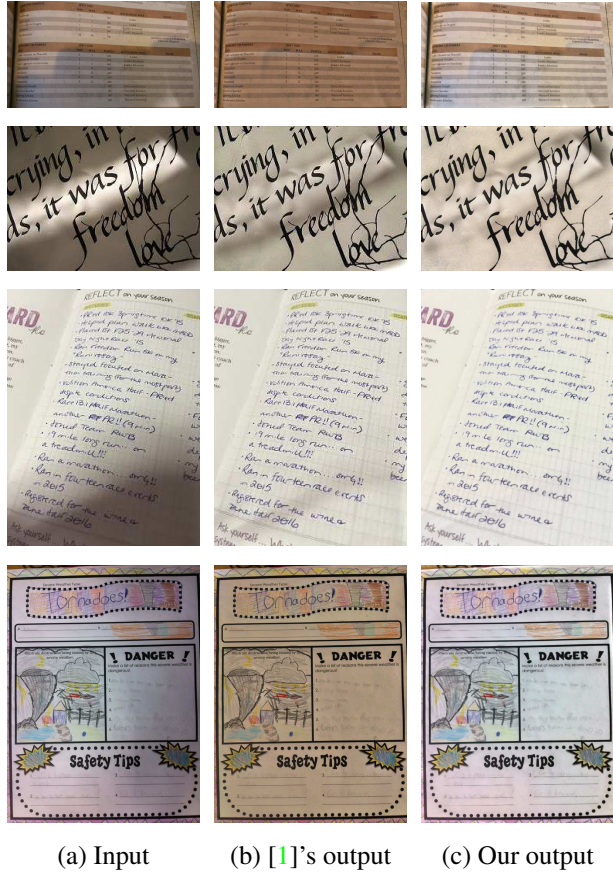(a) Input   (b) [1]'s output   (c) Our output

Figure 12: **Uncontrolled-experiment results:** For real-world challenging images, which are rich in color detail and have multiple shadowing intensities, our method better preserves the color and the artifacts are less eminent.

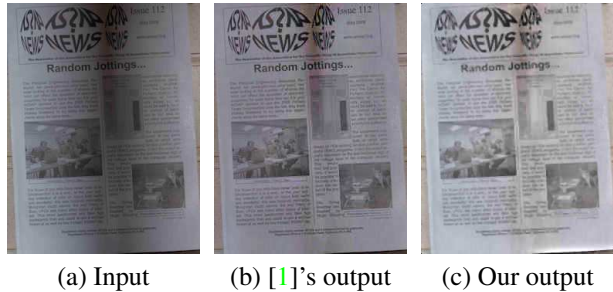(a) Input   (b) [1]'s output   (c) Our output

Figure 13: **Unshadowing limitation:** The result of [1] better matches the ground truth than ours for this example.

lems, both within image processing and for other applications in computer vision. It may prove advantageous even when the feature points are embedded in higher dimensions.

References

[1] S. Bako, S. Darabi, E. Shechtman, J. Wang, K. Sunkavalli, and P. Sen. Removing Shadows from Images. In *Asian Conference on Computer Vision (ACCV)*, 2016. 1, 6, 7, 8

[2] A. F. Bickley. Bickley diary dataset, 1926. Private Journal (unpublished), donated to the Methodist Church Archives, Singapore by Erin Bickley, Jr. 4, 5, 6

[3] J. Bittner and P. Wonka. Visibility in computer graphics. *Environment and Planning B: Planning and Design*, 30(5):729–755, 2003. 1

[4] N. Chaki, S. H. Shaikh, and K. Saeed. *Exploring Image Binarization Techniques*. 2014. 1, 4

[5] D. Cohen-Or, Y. L. Chrysanthou, C. T. Silva, and F. Durand. A survey of visibility for walkthrough applications. *IEEE Transactions on Visualization and Computer Graphics*, 9(3):412–431, 2003. 1

[6] F. Durand, G. Drettakis, and C. Puech. The 3d visibility complex. *ACM Transactions on Graphics (TOG)*, 21(2):176–206, 2002. 1

[7] N. Elmqvist and P. Tsigas. A taxonomy of 3d occlusion management for visualization. *IEEE Transactions on Visualization and Computer Graphics*, 14(5):1095–1109, 2008. 1

[8] D. Fisher-Gewirtzman, A. Shashkov, and Y. Doytsher. Voxel based volumetric visibility analysis of urban environments. *Survey Review*, 45(333):451–461, 2013. 1

[9] H. Garud, A. K. Ray, M. Mahadevappa, J. Chatterjee, and S. Mandal. A fast auto white balance scheme for digital pathology. In *Biomedical and Health Informatics (BHI)*, 2014. 7

[10] B. Gatos, N. Konstantinos, and I. Pratikakis. DIBCO 2009: Document Image Binarization Contest. In *International Conference on Document Analysis and Recognition (ICDAR)*, 2009. 5, 6

[11] H. Gong and D. Cosker. Interactive shadow removal and ground truth for variable scene categories. In *British Machine Vision Conference (BMVC)*, 2014. 6

[12] A. Horé and D. Ziou. Image quality metrics: PSNR vs. SSIM. In *International Conference on Pattern Recognition (ICPR)*, aug 2010. 7

[13] N. R. Howe. Document binarization with automatic parameter tuning. *International Journal on Document Analysis and Recognition*, 16(3):247–258, 2013. 1, 4, 5, 6

[14] S. Katz and A. Tal. Improving the visual comprehension of point sets. In *Computer Vision and Pattern Recognition (CVPR)*, 2013. 3, 4

[15] S. Katz and A. Tal. On the Visibility of Point Clouds. In *International Conference on Computer Vision (ICCV)*, 2015. 3, 4

[16] S. Katz, A. Tal, and R. Basri. Direct visibility of point sets. *ACM Transactions on Graphics*, 26(3):24, 2007. 3

[17] T. Leyvand, O. Sorkine, and D. Cohen-Or. *Ray space factorization for from-region visibility*, volume 22. ACM, 2003. 1

[18] R. Mehra, P. Tripathi, A. Sheffer, and N. J. Mitra. Visibility of noisy point cloud data. *Computers and Graphics*, 34(3):219–230, 2010. 1

[19] R. G. Mesquita, C. A. Mello, and L. H. E. V. Almeida. A new thresholding algorithm for document images based on the perception of objects by distance. *Integrated Computer-Aided Engineering*, 21(2):133–146, 2014. 1, 4

[20] K. Ntirogiannis, B. Gatos, and I. Pratikakis. A combined approach for the binarization of handwritten document images. *Pattern Recognition Letters*, 35(1):3–15, 2014. 1, 4

[21] K. Ntirogiannis, B. Gatos, and I. Pratikakis. H-DIBCO 2014: Handwritten Document Image Binarization Contest. In *International Conference on Frontiers in Handwriting Recognition (ICFHR)*, 2014. 5, 6

[22] D. M. Oliveira, R. D. Lins, and G. d. F. P. Silva. Shading Removal of Illustrated Documents. In *International Conference Image Analysis and Recognition*, 2013. 1, 6

[23] I. Pratikakis, B. Gatos, and K. Ntirogiannis. H-DIBCO 2010: Handwritten Document Image Binarization Contest. In *International Conference on Frontiers in Handwriting Recognition (ICFHR)*, nov 2010. 5, 6

[24] I. Pratikakis, B. Gatos, and K. Ntirogiannis. DIBCO 2011: Document Image Binarization Contest. In *International Conference on Document Analysis and Recognition (ICDAR)*, 2011. 5, 6

[25] I. Pratikakis, B. Gatos, and K. Ntirogiannis. H-DIBCO 2012: Handwritten Document Image Binarization Contest. In *International Conference on Frontiers in Handwriting Recognition (ICFHR)*, 2012. 5, 6

[26] I. Pratikakis, B. Gatos, and K. Ntirogiannis. DIBCO 2013: Document Image Binarization Contest. In *International Conference on Document Analysis and Recognition (ICDAR)*, 2013. 5, 6

[27] I. Pratikakis, K. Zagoris, G. Barlas, and B. Gatos. H-DIBCO 2016: Handwritten Document Image Binarization Contest. In *International Conference on Frontiers in Handwriting Recognition (ICFHR)*, 2016. 5, 6

[28] R. Sibson et al. A brief description of natural neighbour interpolation. *Interpreting multivariate data*, 21:21–36, 1981. 4, 7

[29] B. Su, S. Lu, and C. L. Tan. Robust document image binarization technique for degraded document images. *IEEE Transactions on Image Processing*, 22(4):1408–1417, 2013. 1, 4, 6

[30] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004. 7

[31] W. Yang, W. Guo, K. Peng, and L. Liu. Research on removing shadow in workpiece image based on homomorphic filtering. *Procedia Engineering*, 2012. 6