Blob Reconstruction Using Unilateral Second Order Gaussian Kernels with Application to High-ISO Long-Exposure Image Denoising

Gang Wang[1], Carlos Lopez-Molina[1,2], Bernard De Baets[1]
[1]KERMIT, Department of Mathematical Modelling, Statistics and Bioinformatics, Ghent University, Ghent, Belgium
[2]Dpto. Automatica y Computacion, Universidad Publica de Navarra, Pamplona, Spain
{gang.wang, bernard.debaets}@ugent.be, carlos.lopez@unavarra.es

Abstract

Blob detection and image denoising are fundamental, and sometimes related, tasks in computer vision. In this paper, we propose a blob reconstruction method using scale-invariant normalized unilateral second order Gaussian kernels. Unlike other blob detection methods, our method suppresses non-blob structures while also identifying blob parameters, i.e.*, position, prominence and scale, thereby facilitating blob reconstruction. We present an algorithm for high-ISO long-exposure noise removal that results from the combination of our blob reconstruction method and state-of-the-art denoising methods,* i.e.*, the non-local means algorithm (NLM) and the color version of block-matching and 3-D filtering (CBM3D). Experiments on standard images corrupted by real high-ISO long-exposure noise and real-world noisy images demonstrate that our schemes incorporating the blob reduction procedure outperform both the original NLM and CBM3D.*

1. Introduction

In the computer vision community, the term blob is used to refer to a small structure that is either brighter or darker than the surrounding background in images [6, 9, 22]. It is an important low-level image feature and plays a prominent role in many computer vision tasks, such as bioimagery analysis [14, 23], medical image diagnosis [25], object tracking [19, 26] and autonomous driving [8]. Numerous methods, like Laplacian of Gaussian (LoG) [15], Difference-of-Gaussians (DoG) [15], Determinant of Hessian (DoH) [11], generalized Laplacian of Gaussian (gLoG) [7] and Hessian-based Difference of Gaussians (HDoG) [24], have been proposed over the past several decades. Conventional methods produce significant responses for lines, terminations, corners and edges, hindering the discrimination of the blobs. Moreover, very few researchers have focused on blob reconstruction and reduction. However, on some occasions, blob reduction might be the key to solve some computer vision tasks like the removal of blob noise caused by high-International Organization for Standardization (ISO) sensitivity and long exposure in digital single-lens reflex (DSLR) camera images.

Since it is a fundamental step for improving the final performance in many applications, image denoising has received great attention [20]. A multitude of noise reduction algorithms, such as NLM [1], CBM3D [4], deep Gaussian conditional random field [21], multi-layer perceptron [2], external patch prior guided internal clustering [3] and Wishart fidelity nonlocal total variation [16], have achieved considerable performance in removing white Gaussian noise, salt-and-pepper noise, Poisson noise or speckle noise. Unfortunately, to the best of our knowledge, very few approaches were targeted to the noise caused by high-ISO sensitivity and long-exposure sensors [17, 18], though it is common in DSLR camera images.

In digital photography, high ISO sensitivity and long exposure time are necessary for taking photos of objects under low light conditions. However, the combination of high ISO and long exposure, which maintains a high sensitivity and raises the temperature of the sensor, simultaneously leads to random noise and fixed pattern noise. The random noise is conventionally termed as *grain* [17, 18]. The fixed pattern noise is generally less spatially correlated than the random noise. Therefore, in this paper, we term the noise caused by high ISO and long exposure as *blob noise*, which includes the long-exposure noise and most of the fine *grain* noise. Unfortunately, this kind of blob noise can hardly be removed by conventional methods. The first reason is that blob noise has a certain shape and occupies a certain area, and therefore the self-similarity of image is seriously damaged. The conventional denoising algorithms can hardly recognize whether it is a noise blob or a true visual element of the image. A second reason is that, from the perspective of the frequency domain, blob noise has a large number of very low frequency components that are totally mixed with true image content. Consequently, blob detection and reduction should play an important role in the denoising process.

In this paper, after introducing the definition of blobs and second order Gaussian functions in Section 2, we present in Section 3 a blob reconstruction method based on scale-invariant normalized unilateral Gaussian kernels (USOGKs). In Section 4, we apply this method to high-ISO long-exposure image denoising as a preprocessing procedure that can benefit the subsequent state-of-the-art denoising algorithm, specifically the NLM and the CBM3D algorithms. Finally, in Section 5 experiments on both testing images corrupted by real high-ISO long-exposure noise and real-world noisy images are conducted to validate our method.

2. Related Work

2.1. Blob Structure Definition

In this paper, we consider the case of bright blobs. Lindeberg [10] attempted to give a mathematical description of blobs based on the idea that *the blob would extend until it merges with another blob*. Let $\mathbf{x}^T = [\mathrm{x}, \mathrm{y}]^T$ denote the planar coordinate and a 2D continuous signal $f : \mathbb{R}^2 \to \mathbb{R}$ obtain a local maximum at $\mathbf{x}_0 = [\mathrm{x}_0, \mathrm{y}_0]^T$. For any intensity $h < f(\mathbf{x}_0)$, the protuberance surface $\mathcal{S}(\mathbf{x}_0, h)$ is defined as the convex connected component of

$$\mathcal{S}(\mathbf{x}_0, h) = \Big\{(\mathbf{x}, f(\mathbf{x})) \mid h \leq f(\mathbf{x}) \leq f(\mathbf{x}_0)\Big\} \tag{1}$$

that contains $(\mathbf{x_0}, f(\mathbf{x_0}))$. Obviously, for $h^- < h$, we have $\mathcal{S}(\mathbf{x}_0, h) \subseteq \mathcal{S}(\mathbf{x}_0, h^-)$. A subset $\mathcal{P}$ of $\mathcal{S}(\mathbf{x}_0, h)$ is defined as follows: A point $(\mathbf{x}_i, f(\mathbf{x}_i)) \in \mathcal{S}(\mathbf{x}_0, h)$ belongs to $\mathcal{P}$ if there exists a path $\ell_{(\mathbf{x}_0, f(\mathbf{x}_0)),(\mathbf{x}_i, f(\mathbf{x}_i))}$ from point $(\mathbf{x}_0, f(\mathbf{x}_0))$ to point $(\mathbf{x}_i, f(\mathbf{x}_i))$ such that (i) every point on the path belongs to $\mathcal{S}(\mathbf{x}_0, h)$, and (ii) the derivative of ℓ along this path satisfies $\ell'_{(\mathbf{x}_0, f(\mathbf{x}_0)),(\mathbf{x}_i, f(\mathbf{x}_i))} \leq 0$. The base level h_0 of $\mathcal{S}(\mathbf{x}_0, h)$ is defined as the maximum value of $h < f(\mathbf{x}_0)$ such that $\mathcal{P} \neq \mathcal{S}(\mathbf{x}_0, h)$. Thus the blob with the peak point $\mathbf{x}_0$ can be referred to as the set of points $\mathcal{S}(\mathbf{x}_0, h_0)$, and the blob prominence is given by the difference in intensity between the peak point and the base level:

$$c_0 = f(\mathbf{x}_0) - h_0 \ . \tag{2}$$

A visual representation of this blob model is displayed in Fig. 1.

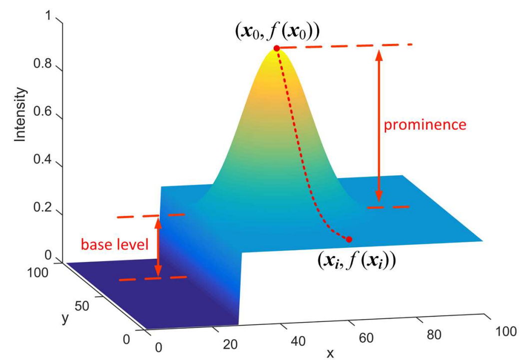


Figure 1. Visual representation of a blob and its measurable characteristics.

2.2. Second Order Gaussian Functions

As described in Lindeberg's work [11, 12], given any n-dimensional signal $f : \mathbb{R}^n \to \mathbb{R}$, its *scale-space representation* $L : \mathbb{R}^n \times \mathbb{R}_+ \to \mathbb{R}$ is given by

$$L(\mathbf{x}; \sigma) = g(\boldsymbol{x}; \sigma) * f(\mathbf{x}) \ , \tag{3}$$

where $*$ denotes the kernel filtering operation, $\boldsymbol{x}$ represents the coordinate of the kernel, the variance σ ($\sigma > 0$) is referred to as the scale parameter and $g(\boldsymbol{x}; \sigma) : \mathbb{R}^n \times \mathbb{R}_+ \to \mathbb{R}$ denotes the Gaussian kernel. Specifically, the 2D Gaussian kernel is defined as:

$$g(\boldsymbol{x}; \sigma) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{\boldsymbol{x}^T\boldsymbol{x}}{2\sigma^2}\right) , \tag{4}$$

where $\boldsymbol{x}^T = [x, y]^T$.

The derivatives of the scale-space representation are then defined based on Eq. (3):

$$L_{x^\alpha, y^\beta}(\mathbf{x}; \sigma) = g_{x^\alpha y^\beta}(\boldsymbol{x}; \sigma) * f(\mathbf{x}) \ , \tag{5}$$

where α and $\beta \in \mathbb{Z}_+$ denote the order of differentiation. Moreover, in order to compensate the magnitude decrease of un-normalized Gaussian derivatives over scales, the γ-normalized derivatives are proposed as

$$L_{x^\alpha, y^\beta, \gamma\text{-norm}}(\mathbf{x}; \sigma) = \sigma^{2\gamma} \cdot g_{x^\alpha y^\beta}(\boldsymbol{x}; \sigma) * f(\mathbf{x}) \ . \tag{6}$$

Based on Eq. (4), the second order Gaussian kernel (SOGK) with respect to x is given by

$$g_{xx}(\boldsymbol{x}; \sigma) = \frac{x^2 - \sigma^2}{2\pi\sigma^6} \exp\left(-\frac{\boldsymbol{x}^T\boldsymbol{x}}{2\sigma^2}\right) . \tag{7}$$

The directional SOGK [13] can be obtained by rotating the SOGK with an orientation of θ:

$$g_{xx}(\boldsymbol{x}; \sigma, \theta) = \frac{\left([\cos\theta, \sin\theta]\boldsymbol{x}\right)^2 - \sigma^2}{2\pi\sigma^6} \exp\left(-\frac{\boldsymbol{x}^T\mathbf{R}_{-\theta}\mathbf{R}_\theta\boldsymbol{x}}{2\sigma^2}\right) , \tag{8}$$

where

$$\mathbf{R}_\theta = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} .$$

Many blob detection methods, including LoG, DoG, gLoG and DoH, employ the SOGK-based Hessian matrix or directional SOGK filters to obtain the blob strength measurement (BSM). However, these methods also produce significant responses for edges, lines, corners and terminations, all of which have a strong spatial structure, as shown in Figs. 2(b), 2(c), 2(d) and 2(e). Moreover, the relationship between the BSM and the blob parameters is implicit so that the blob can hardly be reconstructed.

3. Blob Reconstruction

3.1. Gaussian Blob Modelling

Assuming the blob structure is a Gaussian surface, based on the aforementioned blob definition, we can describe a blob using four parameters: the center position, the scale, the prominence and the base level. Thus a blob in an image can be modelled by a 2D Gaussian function that is formulated as:

$$\mathcal{B}(\mathbf{x};\mathbf{x}_0,\sigma_0,c_0,h_0) = c_0 \cdot \exp\left(-\frac{(\mathbf{x}-\mathbf{x}_0)^T(\mathbf{x}-\mathbf{x}_0)}{2\sigma_0^2}\right) + h_0 \ , \tag{9}$$

where $\mathbf{x}_0$ is the center position, $c_0 \in]0,\ 1]$ is the prominence, $h_0 \in [0,\ 1[$ is the base level and σ_0 denotes the blob scale, respectively.

In fact, as the image intensity at each position is known, we can calculate h_0 based on Eq. (2) once c_0 is obtained. Therefore, the blob can be delimited once these four parameters are determined, and the task of blob detection can also be reconsidered by finding these four parameters.

3.2. Blob Measurement and Reconstruction

Since the conventional blob detection methods can hardly show the relationship between the BSM and the blob parameters, here we will discuss how the second order Gaussian kernel works. Afterwards, the scale-invariant normalized SOGK, which can measure the ideal Gaussian blob precisely, will be derived.

Elaborating on Eq. (6), we redefine the γ-normalized SOGK as:

$$g_{xx,\gamma\text{-norm}}(\boldsymbol{x};\eta,a,\sigma) = (-1)^{\eta} \cdot a \cdot \sigma^{2\gamma} \cdot g_{xx}(\boldsymbol{x};\sigma) \ , \tag{10}$$

where $\eta \in \{0,1\}$ is a parameter determined by the blob type (in this paper we employ kernels with $\eta = 1$ since they produce a positive response for bright blobs), a is a constant that will ensure that the convolutional result between SOGK and the ideal Gaussian blob precisely reflects the blob parameters. Thus the signal response of the γ-normalized SOGK and blob model at position $\mathbf{x}_0$ can be calculated as:

$$\begin{aligned}\mathcal{L}(\mathbf{x}_0) &= g_{xx,\gamma\text{-norm}}(\boldsymbol{x};\eta,a,\sigma) * \mathcal{B}(\mathbf{x};\mathbf{x}_0,\sigma_0,c_0,h_0)\mid_{\mathbf{x}=\mathbf{x}_0} \\ &= ac_0\sigma^{2\gamma-2}\sigma_0(\sigma_0^2+\sigma^2)^{-1}\left[1-\sigma_0^2(\sigma_0^2+\sigma^2)^{-1}\right] \ .\end{aligned} \tag{11}$$

In order to calculate the scale of the blob, we need to find a σ that can produce the largest response in the scale space. Hence we calculate the derivative of $\mathcal{L}(\mathbf{x}_0)$ with respect to σ and set it to zero. The solution is given by

$$\sigma = \gamma^{\frac{1}{2}}(2-\gamma)^{\frac{1}{2}}\sigma_0 \ . \tag{12}$$

Substituting the latter σ in Eq. (12), we obtain

$$\mathcal{L}(\mathbf{x}_0) = \frac{ac_0\gamma}{4(2-\gamma)^{\gamma-2}}\sigma_0^{2\gamma-2} \ . \tag{13}$$

To make sure the response $\mathcal{L}(\mathbf{x}_0)$ is irrelevant to σ_0, *i.e.*, the normalized SOGK is scale-invariant, the value of γ should be set to 1. Now Eq. (13) becomes

$$\mathcal{L}(\mathbf{x}_0) = \frac{1}{4}ac_0 \ . \tag{14}$$

Hence we have $a = 4$, so that $\mathcal{L}(\mathbf{x}_0) = c_0$ while $\mathcal{L}(\mathbf{x}_0)$ achieves its maximum at scale $\sigma = \sigma_0$.

From Eq. (10) and the discussion above, we get the scale-invariant normalized SOGK:

$$g_{\text{sogk}}(\boldsymbol{x};\sigma) = -4\sigma^2 \cdot g_{xx}(\boldsymbol{x};\sigma) \ . \tag{15}$$

Therefore, the proposed SOGKs produce a maximum response at $\mathbf{x}_0$ for the scale σ_0. Moreover, the final maximum response can also indicate the blob prominence c_0.

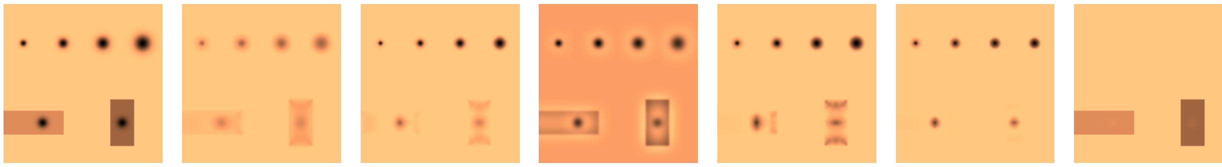
Interestingly, some blobs that are adjacent to other structures do not show ideal appearance. That is, their local contrast values in each direction are not identical. Thus the orientation should be introduced to calculate the local contrast in each direction. For this reason, the USOGK, which inherits the aforementioned merit of the SOGK, is proposed to obtain the blob prominence, namely, the minimum local contrast defined in Section 2.

From Eqs. (8) and (15), the directional scale-invariant normalized SOGK is generated as:

$$g_{\text{sogk}}(\boldsymbol{x};\sigma,\theta) = 2\frac{\sigma^2-\left([\cos\theta,\sin\theta]\boldsymbol{x}\right)^2}{\pi\sigma^4}\exp\left(-\frac{\boldsymbol{x}^T\mathbf{R}_{-\theta}\mathbf{R}_{\theta}\boldsymbol{x}}{2\sigma^2}\right). \tag{16}$$

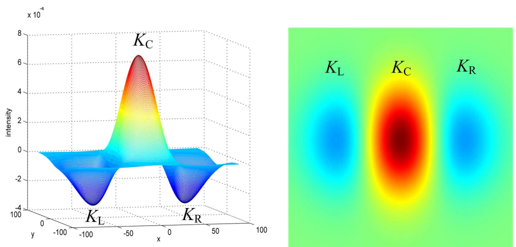
We can see from Eq. (16) and Fig. 3 that a SOGK is spatially divided in three areas, *i.e.*, a central area $K_{\text{C}}(\boldsymbol{x})$ and two symmetrical side-areas $K_{\text{L}}(\boldsymbol{x})$ and $K_{\text{R}}(\boldsymbol{x})$, which are illustrated in Fig. 3. Each of them can be expressed as an individual kernel, leading to:

$$\begin{aligned}K_{\text{C}}(\boldsymbol{x}) &= \begin{cases} g_{\text{sogk}}(\boldsymbol{x}) & \text{, if } g_{\text{sogk}}(\boldsymbol{x}) > 0 \\ 0 & \text{, otherwise} \end{cases} \\ K_{\text{L}}(\boldsymbol{x}) &= \begin{cases} g_{\text{sogk}}(\boldsymbol{x}) & \text{, if } g_{\text{sogk}}(\boldsymbol{x}) < 0 \text{ and } x < -y\tan\theta \\ 0 & \text{, otherwise} \end{cases} \ , \\ K_{\text{R}}(\boldsymbol{x}) &= \begin{cases} g_{\text{sogk}}(\boldsymbol{x}) & \text{, if } g_{\text{sogk}}(\boldsymbol{x}) < 0 \text{ and } x > -y\tan\theta \\ 0 & \text{, otherwise} \end{cases}\end{aligned} \tag{17}$$



(a) (b) (c) (d) (e) (f) (g)

Figure 2. Response produced by LoG (b), DoH (c), gLoG (d), SOGK (e), USOGKs (f) and blob reduction result using our method (g) on a synthetic image (a). The intensity range of each image has been reversed and adjusted for better display.



(a) (b)

Figure 3. Three-dimensional and planar representations of a second order Gaussian kernel in $\mathbb{R}^2$.

where θ is the rotation angle and $[x, y]^T$ denotes the planar coordinate of the kernel. Note that $K_L(\boldsymbol{x})$ and $K_R(\boldsymbol{x})$ have the same sign, while $K_C(\boldsymbol{x})$ has the opposite one.

As a matter of fact, $g_{\text{sogk}}(\boldsymbol{x}; \sigma, \theta)$ measures the difference between the central part and its two side parts, *i.e.* the average of the relative difference on both sides along the orientation determined by θ. As a result, using SOGKs usually leads to a great loss of orientation information, as they always bind the two side parts together. In order to measure the blob prominence described in Section 2, the directional scale-invariant normalized unilateral second order Gaussian kernel is proposed as follows.

For a given scale σ, the USOGK in the direction of θ is defined by

$$g_{\text{usogk}}(\boldsymbol{x}; \sigma, \theta, \mathbf{x}) = K_C(\boldsymbol{x}) + 2\lambda(\mathbf{x})K_L(\boldsymbol{x}) + 2[1 - \lambda(\mathbf{x})]K_R(\boldsymbol{x}) \,, \tag{18}$$

where

$$\lambda(\mathbf{x}) = \begin{cases} 1 & , \text{if } |K_L(\boldsymbol{x}) * \mathcal{N}(\mathbf{x})| > |K_R(\boldsymbol{x}) * \mathcal{N}(\mathbf{x})| \\ 0 & , \text{otherwise} \end{cases} \,, \tag{19}$$

where $\mathcal{N}(\mathbf{x})$ is the image patch centered at position $\mathbf{x}$ with the same size of $g_{\text{usogk}}(\boldsymbol{x}; \sigma, \theta, \mathbf{x})$.

To make the filtering easier to implement, we calculate the filtering result $f_{\text{usogk}}(\mathbf{x}; \sigma, \theta)$ between $g_{\text{usogk}}(\boldsymbol{x}; \sigma, \theta, \mathbf{x})$ and the signal $f(\mathbf{x})$ by

$$\begin{aligned} & f_{\text{usogk}}(\mathbf{x}; \sigma, \theta) \\ & = g_{\text{usogk}}(\boldsymbol{x}; \sigma, \theta, \mathbf{x}) * f(\mathbf{x}) \\ & = f_{\text{LR}}(\mathbf{x})\Big[\Big(K_C(\boldsymbol{x}) + 2K_L(\boldsymbol{x})\Big) * f(\mathbf{x})\Big] + \\ & \Big(1 - f_{\text{LR}}(\mathbf{x})\Big)\Big[\Big(K_C(\boldsymbol{x}) + 2K_R(\boldsymbol{x})\Big) * f(\mathbf{x})\Big] \,, \end{aligned} \tag{20}$$

where

$$f_{\text{LR}}(\mathbf{x}) = \begin{cases} 1 & , \text{if } \Big(-K_L(\boldsymbol{x}) + K_R(\boldsymbol{x})\Big) * f(\mathbf{x}) > 0 \\ 0 & , \text{otherwise} \end{cases} \,. \tag{21}$$

Obviously, the values of $\theta \in [0, \pi[$ are able to cover all possible orientations of the filter. For a given scale σ, the kernel $g_{\text{usogk}}(\boldsymbol{x}; \sigma, \theta, \mathbf{x})$ that produces the minimum response among all the orientations is selected and its corresponding response is retained as the candidate prominence $f_{\text{usogk}}(\mathbf{x}; \sigma)$ at the scale of σ. The final BSM is obtained by selecting the maximum of $f_{\text{usogk}}(\mathbf{x}; \sigma)$ in the scale space. That is, the BSM is selected by

$$f_{\text{BSM}}(\mathbf{x}) = \max_{\sigma \in \boldsymbol{\Sigma}} \min_{\theta \in \boldsymbol{\Theta}} f_{\text{usogk}}(\mathbf{x}; \sigma, \theta) \,, \tag{22}$$

where $\boldsymbol{\Sigma}$ denotes the set of values of σ and $\boldsymbol{\Theta}$ represents the set of orientations.

Subsequently, the estimated blob image $f_{\text{blob}}(\mathbf{x})$ can be reconstructed using the detected position, prominence and scale, all of which are calculated based on Eq. (22).

In order to accommodate digital image processing, the discrete version of the USOGK should be considered. We can get both the discrete Gaussian function and SOGK by sampling the formula in Eq. (16) in the 2D integer coordinate:

$$\begin{aligned} & g_{\text{sogk}}(\boldsymbol{m}; \sigma_j, \theta_w) \\ & = 2\frac{\left(\sigma_j^2 - [\cos\theta_w, \sin\theta_w]\boldsymbol{m}\right)^2}{\pi\sigma_j^4} \exp\left(-\frac{\boldsymbol{m}^T\mathbf{R}_w^T\mathbf{R}_w\boldsymbol{m}}{2\sigma_j^2}\right) , \end{aligned} \tag{23}$$

where

$$\mathbf{R}_w = \begin{bmatrix} \cos\theta_w & \sin\theta_w \\ -\sin\theta_w & \cos\theta_w \end{bmatrix},$$
$$\theta_w = (w-1)\pi/W, \quad w = 1, 2, 3, \ldots, W,$$

where $\boldsymbol{m} = [m_x, m_y]^T \in \mathbb{Z}^2$ represents the 2D image coordinate of the kernel, $\sigma_j \in \boldsymbol{\Sigma}_J = \{\sigma_1, \sigma_2, ..., \sigma_J\}$ with $\sigma_j > 0$ and $\theta_w \in \boldsymbol{\Theta}_W = \{\theta_1, \theta_2, ..., \theta_W\}$ with $\theta_w \in [0, \pi[$ denote the parameter of scale and orientation, respectively. Let $\mathbf{m} = [\mathrm{m}_x, \mathrm{m}_y]^T \in \mathbb{Z}^2$ represent the 2D image coordinate, we easily obtain the discrete USOGK $g_{\text{usogk}}(\boldsymbol{m}; \sigma_j, \theta_w, \mathbf{m})$ using Eqs. (18), (19), (20), (21) and (23).

Next, a family of discrete USOGKs combining all the possible scales and orientations is applied to filter the image. Figure 4 illustrates the kernels generating the USOGK at a specific scale. Consequently, the BSM represented by I_{BSM} is produced based on Eq. (22). At the same time, the corresponding parameters of each blob, *i.e.*, position, scale and prominence, are also obtained.

Figure 2(f) illustrates the blob detection result on a synthetic image using the method based on USOGKs. Comparing with the detection results of LoG, DoH, gLoG and SOGK shown in Figs. 2(b), 2(c), 2(d) and 2(e), one can find that only blob structures are detected in Fig. 2(f).

4. High-ISO Long-Exposure Noise Elimination

4.1. Noise Modelling

Denoising is the process of restoring the original image by reducing the undesirable noise from a corrupted image. Let a 2D image signal $I_n^{(q)}(\mathbf{m})$ ($q \in \{1, 2, 3\}$) denote the q^{th} channel of the color space in a noisy color image $I_n(\mathbf{m})$:

$$I_n^{(q)}(\mathbf{m}) = F^{(q)}\Big(I_t^{(q)}(\mathbf{m})\Big) + n_b^{(q)}(\mathbf{m}) \ , \tag{24}$$

where $I_t^{(q)}$ is the q^{th} channel of the true image, $n_b^{(q)}$ denotes the independent additive blob noise and $F^{(q)}$ is a degradation function caused by other noise. Here, we model the blob noise by spatially mixed Gaussian functions as follows:

$$n_b^{(q)}(\mathbf{m}) = \sum_{\mathbf{m}_i \in I_t^{(q)}} H^{(q)}(u - u_0)\tilde{g}^{(q)}(\mathbf{m}; \mathbf{m}_i, \tilde{\sigma}_i, \tilde{c}_i) \ , \tag{25}$$

where $H^{(q)}$ is the Heaviside step function with u uniformly distributed on the unit interval $[0, 1]$, $u_0 \in [0, 1]$ a constant and $\tilde{g}^{(q)}$ a Gaussian function defined by:

$$\tilde{g}^{(q)}(\mathbf{m}; \mathbf{m}_i, \tilde{\sigma}_i, \tilde{c}_i) = \tilde{c}_i \cdot \exp\left(-\frac{(\mathbf{m} - \mathbf{m}_i)^T(\mathbf{m} - \mathbf{m}_i)}{2\tilde{\sigma}_i^2}\right) , \tag{26}$$

where both $\tilde{\sigma}_i$ and $\tilde{c}_i$ are normally distributed. As shown in Fig. 5, the modelled noise can reflect the real high-ISO long-exposure noise well.

4.2. Noise Elimination Based on Blob Reduction

From Eqs. (25) and (26), we find that an approximated $\hat{I}_b \approx n_b$ can be reconstructed once the parameters $\mathbf{m}_i$, $\tilde{\sigma}_i$ and $\tilde{c}_i$ are determined at each position. Thus, the blob reconstruction method proposed in the preceding section is introduced to produce the blob map $\hat{I}_b$. Then the main part of the blob noise can be removed as

$$\hat{I} = I_n - \hat{I}_b \ . \tag{27}$$

Figure 2(g) shows a blob reduction result from which one can see that only the blobs are eliminated.

Finally, the subsequent denoising algorithms, NLM and CBM3D, are employed to eliminate the residual noise and the errors produced in the blob reduction procedure to obtain the final denoising result, respectively.

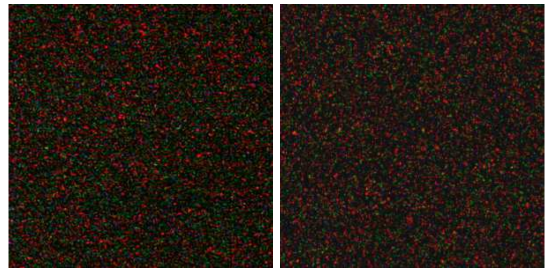


Figure 5. Illustration of noise modelling. The left image shows real high-ISO long-exposure noise, while the right one shows the modelled noise. Please zoom electronically for a better view.

5. Experimental Validation

To evaluate the performance of the proposed methods, we first apply them on standard images to which real high-ISO long-exposure noise is added. In this way, we can obtain both the ground truth and noisy images so that a quantitative evaluation can be performed. Besides, we further apply our methods to real-world images to illustrate the qualitative performance. The proposed methods, which involve the blob reduction procedure summarized in Algorithm 1, are tested in different color spaces, specifically RGB, $\mathrm{YC_rC_b}$, CIE Lab, YUV and $\mathrm{A_{opp}}$ [5]. Surprisingly, the best results were reached for the RGB color space, and so we have kept it as the standard. We also compare our method with the original NLM as well as the CBM3D methods. For a fair comparison, experiments are conducted with either the recommended parameters mentioned in the original papers or optimally tuned ones. Besides, all the experiments are implemented in the Matlab (R2014b) environment on a PC with Intel Core(TM) i5-5200U CPU 2.20GHz $\times$ 2 and RAM 4.00GB.

5.1. Performance on Standard Images

We add real noise, which is created by taking black photos using a DSLR camera with high ISO sensitivity and long exposure time, to all standard testing images including *Baboon*, *Barbara*, *Lena*, *Peppers* and *Sailboat* (a.k.a. *Sailboat*

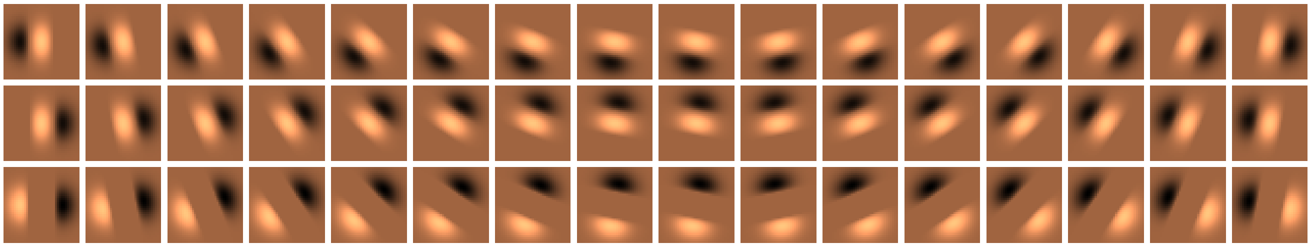


Figure 4. Illustration of kernels generating USOGKs. The top row shows $(K_\mathrm{C} + 2K_\mathrm{L})$, the middle row shows $(K_\mathrm{C} + 2K_\mathrm{R})$ and the bottom row shows $(-K_\mathrm{L} + K_\mathrm{R})$. The intensity range of each patch has been adjusted for better display.

Algorithm 1 Proposed blob reduction method

Input: Noisy image I_n, standard deviation set $\mathbf{\Sigma}_J = \{\sigma_1, \sigma_2, ..., \sigma_J\}$, orientation set $\mathbf{\Theta}_W = \{\theta_1, \theta_2, ..., \theta_W\}$

Output: Blob reduction result $\hat{I}$

for each $I_n^{(q)} \in I_n$ **do**
  for each $\sigma_j \in \mathbf{\Sigma}_J$ **do**
    for each $\theta_w \in \mathbf{\Theta}_W$ **do**
      $I_{\mathrm{usogk}}^{(q)}(\mathbf{m}; \sigma_j, \theta_w) \leftarrow g_{\mathrm{usogk}}(\boldsymbol{m}; \sigma_j, \theta_w, \mathbf{m}) * I_n^{(q)}(\mathbf{m})$
    end for
    $I_{\mathrm{usogk}}^{(q)}(\mathbf{m}; \sigma_j) \leftarrow \min\limits_{\theta_w \in \mathbf{\Theta}_W} I_{\mathrm{usogk}}^{(q)}(\mathbf{m}; \sigma_j, \theta_w)$
  end for
  $I_{\mathrm{BSM}}^{(q)}(\mathbf{m}) \leftarrow \max\limits_{\sigma_j \in \mathbf{\Sigma}_J} I_{\mathrm{usogk}}^{(q)}(\mathbf{m}; \sigma_j)$
  Reconstruct $\hat{I}_b^{(q)}$ using $I_{\mathrm{BSM}}^{(q)}$;
  $\hat{I}^{(q)} \leftarrow (I_n^{(q)} - \hat{I}_b^{(q)})$;
end for
$\hat{I} \leftarrow \{\hat{I}^{(1)}, \hat{I}^{(2)}, \hat{I}^{(3)}\}$.

on lake), all of which are shown in Fig. 6. These testing images reflect a diversity of image content and are extensively used in the field of image processing. Their resolution is $512 \times 512 \times 3$. To attain a quantitative performance evaluation, we adopt the widely used peak signal-to-noise ratio (PSNR) as criterion:

$$\mathrm{PSNR} = 10\log_{10}\left(\frac{N}{\sum_{\mathbf{m}\in I_\mathrm{t}}[I_\mathrm{t}(\mathbf{m}) - I_\mathrm{d}(\mathbf{m})]^2}\right), \tag{28}$$

where I_t is the true image, I_d is the denoising result and N is the number of elements in I_d. Note that both I_t and I_d are transformed into the intensity range $[0, 1]$.

As shown in Fig. 6 and Tab. 1, the proposed approaches, which include the blob reduction procedure, outperform the original NLM and CBM3D methods in all cases, respectively. The acceptable PSNRs indicates that most of the noise has been removed by the proposed methods. Besides, the method combining blob reduction and CBM3D performs the best among all the compared methods at an acceptable computational cost. The average executing time of each method is shown in Tab. 2. However, we can also infer from the denoising results that the proposed method underperforms in areas with plenty of texture structures. For example, all the four approaches show a limited performance on the testing image of *Baboon*. This is because the denoising algorithm has to make a hard compromise between removing more noise and preserving more structured details.

5.2. Performance on Real-world Images

We further evaluate the performance of our schemes on real-world noisy images. Their resolution is $250 \times 250 \times 3$. Figure 7 shows that though a large part of noise can be removed by the original NLM or CBM3D method, the denoising results obtained using methods incorporating the blob reduction procedure appear to be of better quality, although this is subjective. This demonstrates that both the NLM and CBM3D method can benefit a lot from the blob reduction procedure to produce better denoising results in the case of high-ISO long-exposure noise.

6. Conclusion

In this paper we have proposed a novel blob reconstruction method using the scale-invariant normalized unilateral second order Gaussian kernels. The method can suppress non-blob structures and produce significant BSM only for blobs. The BSM can also indicate parameters of blobs, *i.e.*, center position, prominence and scale, and consequently the blobs can be reconstructed. We have applied this method to remove high-ISO long-exposure noise as a preprocessing step to the NLM and CBM3D method, respectively. Experimental results demonstrate that the latter methods can both benefit from the blob reduction procedure and produce better denoising results in the case of high-ISO long-exposure noise.

Acknowledgement

The authors would like to thank Prof. Irina Perfilieva of University of Ostrava for providing part of the experimental images.

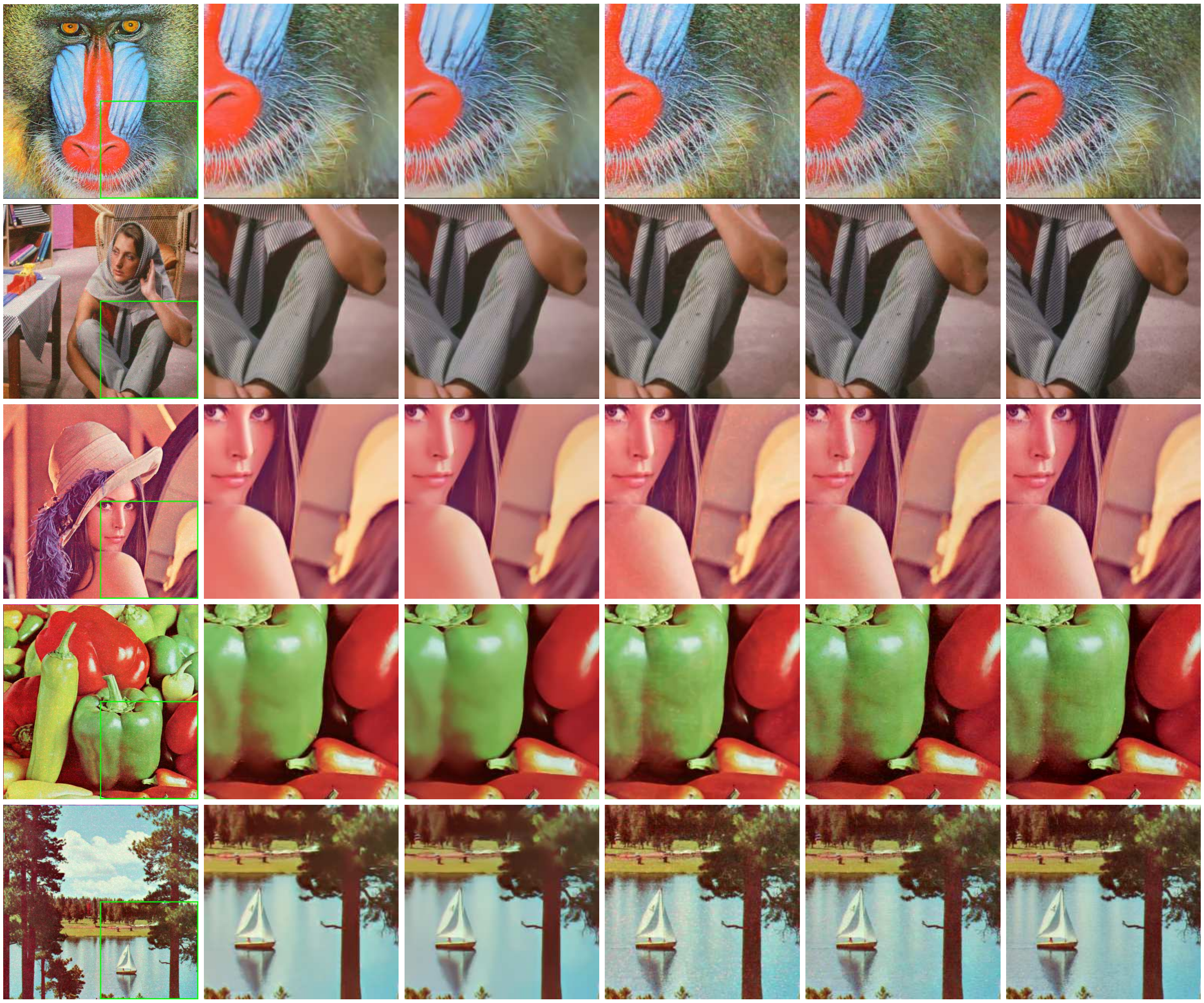


Figure 6. Experimental images and the zoomed-in sections of denoising results. Left Column: Corrupted images using real high-ISO long-exposure noise. Second Column: Results of NLM. Third Column: Results of our method+NLM. Fourth Column: Results of CBM3D. Fifth Column: Results of our method+CBM3D. Right Column: Ground truth. Please zoom electronically for a better view.

Method	*Baboon*	*Barbara*	*Lena*	*Peppers*	*Sailboat*
NLM	24.92	29.69	29.32	27.32	27.34
ours+NLM	25.51	31.52	31.15	30.00	28.20
CBM3D	26.94	30.03	29.46	27.32	28.19
ours+CBM3D	28.55	32.59	32.14	30.85	30.42

Table 1. PSNR (dB) of denoising results.

References

[1] A. Buades, B. Coll, and J. M. Morel. A non-local algorithm for image denoising. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages 60–65, 2005.

[2] H. C. Burger, C. J. Schuler, and S. Harmeling. Image denoising: Can plain neural networks compete with bm3d? In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2392–2399, 2012.

[3] F. Chen, L. Zhang, and H. Yu. External patch prior guided internal clustering for image denoising. In *2015 IEEE International Conference on Computer Vision*, pages 603–611, 2015.

[4] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian. Im-

Method	*Baboon*	*Barbara*	*Lena*	*Peppers*	*Sailboat*
NLM	253.95	262.19	253.95	252.81	254.49
ours+NLM	259.98	267.67	269.10	261.52	259.15
CBM3D	2.88	3.95	4.15	4.18	3.70
ours+CBM3D	9.75	9.74	11.50	13.52	10.21

Table 2. Execution time (s) of different methods on each testing image.

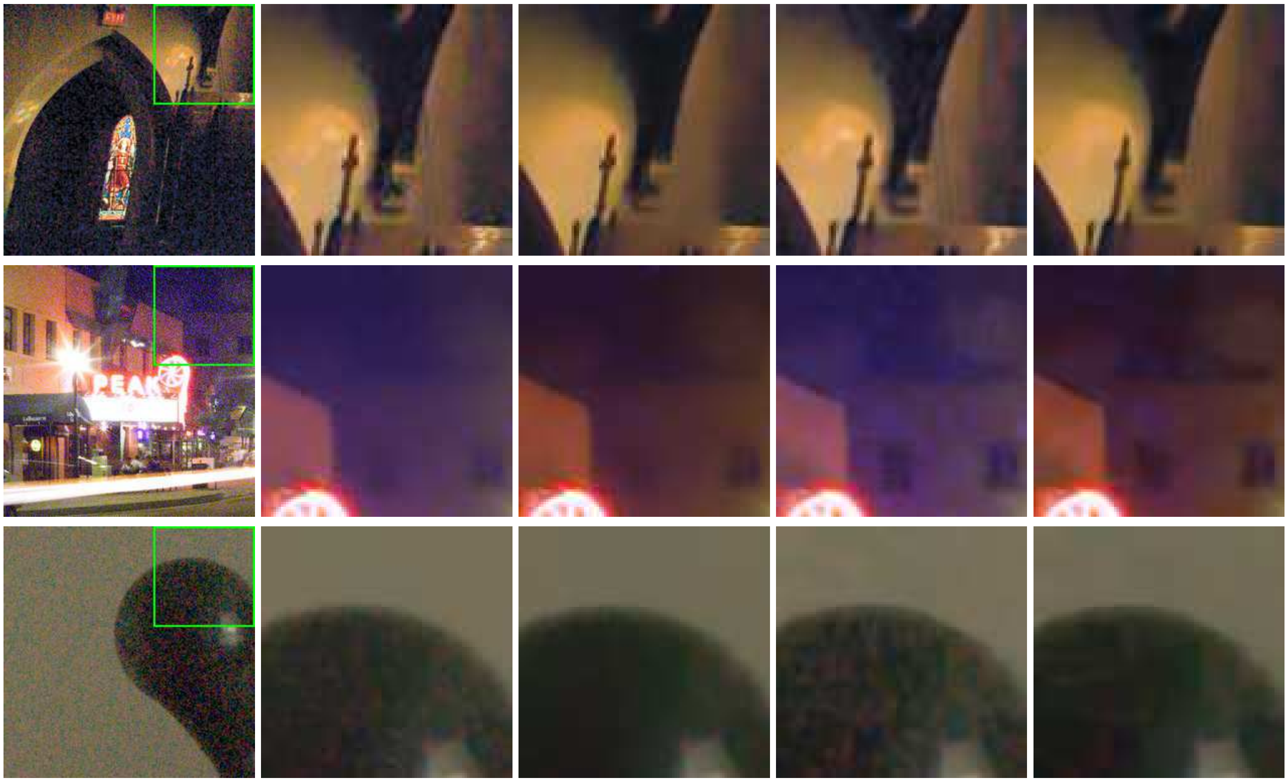


Figure 7. Real-world noisy images and the zoomed-in sections of denoising results. Left Column: Real-world noisy images (Courtesy: Peter K. Burian and Dave Johnson). Second Column: Results of NLM. Third Column: Results of our method+NLM. Fourth Column: Results of CBM3D. Right Column: Results of our method+CBM3D. Please zoom electronically for a better view.

age denoising by sparse 3-D transform-domain collaborative filtering. *IEEE Transactions on Image Processing*, 16(8):2080–2095, 2007.

[5] G. Ghimpeţeanu, T. Batard, M. Bertalmío, and S. Levine. A decomposition framework for image denoising algorithms. *IEEE Transactions on Image Processing*, 25(1):388–399, 2016.

[6] J. J. Koenderink. The structure of images. *Biological Cybernetics*, 50(5):363–370, 1984.

[7] H. Kong, H. C. Akakin, and S. E. Sarma. A generalized laplacian of gaussian filter for blob detection and its applications. *IEEE Transactions on Cybernetics*, 43(6):1719–1733, 2013.

[8] H. Kong, S. E. Sarma, and F. Tang. Generalizing laplacian of gaussian filters for vanishing-point detection. *IEEE Transactions on Intelligent Transportation Systems*, 14(1):408–418, 2013.

[9] Y. Li, S. Wang, Q. Tian, and X. Ding. A survey of recent advances in visual feature detection. *Neurocomputing*, 149:736–751, 2015.

[10] T. Lindeberg. Detecting salient blob-like image structures and their scales with a scale-space primal sketch: A method for focus-of-attention. *International Journal of Computer Vision*, 11(3):283–318, 1993.

[11] T. Lindeberg. Feature detection with automatic scale selection. *International Journal of Computer Vision*, 30(2):79–116, 1998.

[12] T. Lindeberg. Scale selection properties of generalized scale-space interest point detectors. *Journal of Mathematical Imaging and Vision*, 46(2):177–210, 2013.

[13] C. Lopez-Molina, G. Vidal-Diez De Ulzurrun, J. M. Baetens, J. Van Den Bulcke, and B. De Baets. Unsupervised ridge detection using second order anisotropic Gaussian kernels. *Signal Processing*, 116:55–67, 2015.

[14] X. Lou, U. Koethe, J. Wittbrodt, and F. A. Hamprecht. Learning to segment dense cell nuclei with shape prior. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1012–1018, 2012.

[15] D. Marr and E. Hildreth. Theory of edge detection. *Proceedings of the Royal Society of London B: Biological Sciences*, 207(1167):187–217, 1980.

[16] X. Nie, H. Qiao, B. Zhang, and X. Huang. A nonlocal TV-based variational method for PolSAR data speckle reduction. *IEEE Transactions on Image Processing*, 25(6):2620–2634, 2016.

[17] Y. I. Pyo, R. H. Park, and S. Chang. Noise reduction in high-ISO images using 3-D collaborative filtering and structure extraction from residual blocks. *IEEE Transactions on Consumer Electronics*, 57(2):687–695, 2011.

[18] T. Rabie. Adaptive hybrid mean and median filtering of high-ISO long-exposure sensor noise for digital photography. *Journal of Electronic Imaging*, 13(2):264, 2004.

[19] M. Schiegg, P. Hanslovsky, B. X. Kausler, L. Hufnagel, and F. A. Hamprecht. Conservation tracking. In *2013 IEEE International Conference on Computer Vision*, pages 2928–2935, 2013.

[20] L. Shao, R. Yan, X. Li, and Y. Liu. From heuristic optimization to dictionary learning: A review and comprehensive comparison of image denoising algorithms. *IEEE Transactions on Cybernetics*, 44(7):1001–1013, 2014.

[21] R. Vemulapalli, O. Tuzel, and M. Y. Liu. Deep gaussian conditional random field network: A model-based deep network for discriminative denoising. In *2016 IEEE Conference on Computer Vision and Pattern Recognition*, pages 4801–4809, 2016.

[22] G. Wang, Y. Chen, S. Yang, M. Gao, and Y. Dai. Detection of infrared dim small target based on image patch contrast. *Optics and Precision Engineering*, 23(5):1424–1433, 2015.

[23] H. Xu, C. Lu, R. Berendt, N. Jha, and M. Mandal. Automatic nuclei detection based on generalized laplacian of gaussian filters. *IEEE Journal of Biomedical and Health Informatics*, 21(3):826–837, 2017.

[24] M. Zhang, T. Wu, S. C. Beeman, L. Cullen-McEwen, J. F. Bertram, J. R. Charlton, E. Baldelomar, and K. M. Bennett. Efficient small blob detection based on local convexity, intensity and shape information. *IEEE Transactions on Medical Imaging*, 35(4):1127–1137, 2016.

[25] M. Zhang, T. Wu, and K. M. Bennett. Small blob identification in medical images using regional features from optimum scale. *IEEE Transactions on Biomedical Engineering*, 62(4):1051–1062, 2015.

[26] T. Zhang, S. Liu, N. Ahuja, M.-H. Yang, and B. Ghanem. Robust visual tracking via consistent low-rank sparse learning. *International Journal of Computer Vision*, 111(2):171–190, 2015.