# Depth-Aware Motion Magnification

Julian F.P. Kooij[1,2](✉) and Jan C. van Gemert[1]

[1] Delft University of Technology, Delft, The Netherlands
{J.F.P.Kooij,J.C.vanGemert}@tudelft.nl
[2] Leiden University Medical Center, Leiden, The Netherlands

**Abstract.** This paper adds depth to motion magnification. With the rise of cheap RGB+D cameras depth information is readily available. We make use of depth to make motion magnification robust to occlusion and large motions. Current approaches require a manual drawn pixel mask over all frames in the area of interest which is cumbersome and error-prone. By including depth, we avoid manual annotation and magnify motions at similar depth levels while ignoring occlusions at distant depth pixels. To achieve this, we propose an extension to the bilateral filter for non-Gaussian filters which allows us to treat pixels at very different depth layers as missing values. As our experiments will show, these missing values should be ignored, and not inferred with inpainting. We show results for a medical application (tremors) where we improve current baselines for motion magnification and motion measurements.

**Keywords:** Motion magnification · Bilateral filter · RGB+D

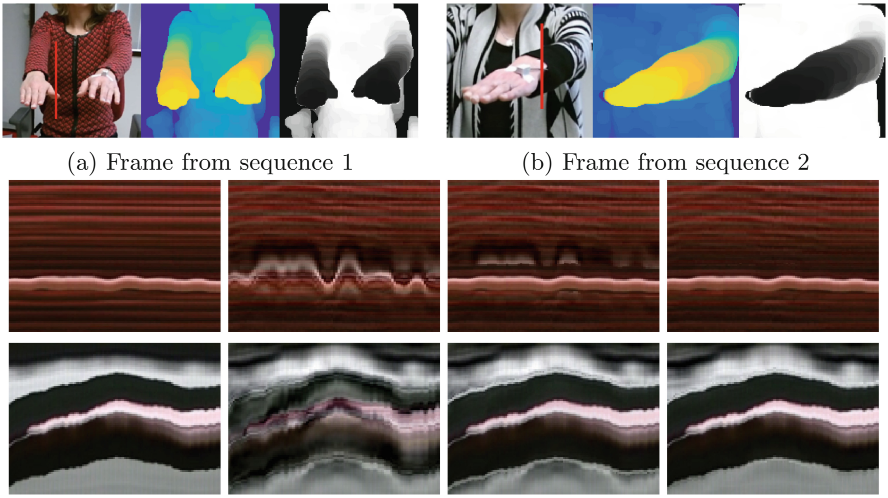
## 1 Introduction

Magnifying tiny motions in video [3,4] opened up a wealth of applications. Examples include: reconstructing speech exclusively from small visual vibrations [5], detecting a heart-beat either from blood flow [4] or from tiny head motions [6], magnifying muscle tremors [7], segmenting blood vessels [8] or estimating material properties by the way it moves [9]. In this paper we propose to only magnify motion at selected depth ranges, which makes motion magnification robust to occlusions and large motions at other depths. Robustness is especially important to open up new applications in the medical domain such as tremor assessment [10–12], where the interaction between doctor and patient should not be disturbed, and prerequisites for video processing should not limit the poses and exercises dictated by the medical protocol.

Currently though, magnifying tiny motions requires that there are no occlusions or large motions [1,3,4]. A recent solution proposes to manually indicate the large motions by drawing a binary pixel mask on the frames of interest [2].

**Electronic supplementary material** The online version of this chapter (doi:10.1007/978-3-319-46484-8_28) contains supplementary material, which is available to authorized users.

B. Leibe et al. (Eds.): ECCV 2016, Part VIII, LNCS 9912, pp. 467–482, 2016.
DOI: 10.1007/978-3-319-46484-8_28



(a) Frame from sequence 1

(b) Frame from sequence 2

(c) original

(d) magnified [1]

(e) CVPR'15 [2]

(f) ours

**Fig. 1.** Comparison of our and baseline magnification approaches when magnifying small motions in the background (here, body) behind moving occluders (here, trembling hands). (a), (b) For two sequences, the input image, depth map, and depth-dependent magnification matte of one frame (black/white is zero/full magnification). (c)–(f) Space-time slices for the red lines in input images. Our approach suppresses unwanted magnification artifacts from the foreground in the magnified background. *See supplementary material for videos.* (Color figure online)

While a mask indicates which pixels should be used, it does not solve how to ignore the motion filter responses on the edge of the mask. Motion filters have a certain spatial extent and they 'leak' across the mask border. Moreover, manually drawing such a mask on a moving target is challenging and time-consuming. We instead exploit depth to automatically define the mask. Furthermore, we prevent the 'leaking' by ignoring motion responses from very different depths whereas filter responses from close-by depth layers are weighted.

Several techniques are available for weighting filter responses [13–16]. These techniques allow weighted Gaussian smoothing or interpolation, for example, on intensity differences resulting in edge-preserving smoothing. However, high-quality motion magnification [1] depends on the complex steerable pyramid [17,18] which consists of non-Gaussian filters for which standard weighting of filter responses [13,14,16] cannot be used. To illustrate, consider a Gaussian derivative filter. Since it integrates to zero, it will give no response on a constant valued input image. Intuitively, the response should not change if some parts of the input are ignored, but reducing some filter weights to zero would now actually yield non-zero output. In other words, the Gaussian derivative cannot be treated as a weighted input aggregate. In this paper we therefore develop filter weighting of non-Gaussian filters, which can ignore input by treating it as missing values.

When images have missing values, there are several advanced inpainting techniques [19–23] available to estimate what is lost. It is not clear, however, how inpainting can be used to infer missing values between multiple depth layers. We propose a different goal. We do not want to recover what is lost: we want to ignore what is there.

In the following sections we first discuss related work, then how to ignore filter responses from different depth layers, and how this allows depth-aware motion magnification. We experimentally compare against inpainting and show example applications in the medical domain on hand tremors.

## 2 Related Work

Motion can be magnified by explicitly tracking feature points with optical flow [24]. The motion is magnified by re-scaling the moving points and adding them back to the video. Optical flow is estimated locally between pairs of frames which is noisy. This noise affects the motion magnification since local motion is represented by a single unique feature point. In contrast to feature point tracking, Eulerian video magnification [4] estimates motion frequency over longer time periods which is more stable. Thus, the method is well-suited for amplifying tiny imperceptible motions. Impressive improvements [1] on the stability of linear motion magnification [4] are made by relying on complex steerable pyramid filters [17,18]. A significant speedup without perceptual decrease in quality can be obtained by approximating the complex pyramid with the Riesz pyramid [25]. While extremely successful for clean video sequences, all these methods assume that there are no occlusions or large motions present. Our method is specifically designed to deal with such cases.

With some help by the user, occlusion or large motions can be manually indicated. Examples of user input on video processing include de-animation [26], blending between face performances [27], video segmentation [28], and video stabilization [29]. For motion magnification a manual drawn mask can specify which pixels to magnify and which pixels to ignore [2]. In this paper we extend this line of reasoning, replacing the manual drawn mask by a weighted mask obtained from depth to ignore filter responses outside a target depth range.

Incorporating weighted responses in a filter is done with the bilateral filter [16]. It applies Gaussian blurring to an image, but locally adapts the Gaussian weights to suppress contributions of neighbourhood pixels with very different intensity levels. The fast bilateral filter [15] offers a significant speedup by approximation. This is achieved by transforming the 2D input image into a 3D sparse matrix, where the 3rd z-dimension is given by a pixel's intensity level. The speedup comes from allowing standard 3D convolutions to obtain intensity-weighted responses. In this paper we begin with the fast bilateral filter [15] due to its speed. However, where the bilateral filter only allows weighted Gaussian smoothing or interpolation we require non-Gaussian filters: the complex steerable pyramid [17,18] as used in high-quality motion magnification [1]. Instead of weighting values, we adapt the bilateral filter so it can handle missing values.

Inferring missing values in images by inpainting typically exploits texture synthesis and pixel consistency [19,30]. Strong step edges can be retained [21] and image statistics through patch-exemplars can give a good prior on what values to infer [23]. Inpainting can be done efficiently [20], making it in principle suitable for video processing. While inpainting could be used to fill in missing values for very different depth layers, it is not clear how to use inpainting to combine closer depth layers. In contrast to inpainting we do not wish to infer what should be present at all depth layers. Our goal is to remove all filter influences from pixels at different depth layers.

## 3 Approach

This section starts with the bilateral filter formulation [16], followed by our non-Gaussian extension. We then apply the developed technique to complex steerable pyramids and use these for occlusion-aware Eulerian motion magnification [1,2,24] and measurement [31]. We note that other image processing tasks could also benefit from the non-Gaussian bilateral filter (see supplementary material for examples), and for instance use intensity, optical flow, or color instead of depth to filter micro-textures, stationaries, surfaces.

### 3.1 Bilateral Filter

The bilateral filter [16] can be used for depth-aware smoothing. Given input image $I$ and corresponding depth image $E$, the bilateral filter computes output image $O$. By defining $y \in N(x)$ as the local a neighbourhood of 2D image locations $x = (u, v)$, and using $O(x)$ as a shorthand for $O(u, v)$, the bilateral filter can be written as a weighted average

$$O(x) = \frac{1}{W(x)} \sum_{y \in N(x)} w(|x - y|, E(x) - E(y))\; I(y) \tag{1}$$

$$w(d_s, d_E) = G(d_s; \sigma_s) \times G(d_E; \sigma_r) \tag{2}$$

where $W(x) = \sum_{y \in N(x)} w(|x-y|, E(x)-E(y))$ is the weight normalization term at $x$, and $G(x; \sigma) = \exp\left(-\frac{x}{2\sigma^2}\right)$ is the Gaussian kernel. The positive weights $w(d_I, d_E)$ approach zero as the spatial distance $d_s$ or the depth distance $d_E$ increases. There are two smoothing parameters, the spatial standard deviation $\sigma_s$, which controls the amount of spatial blurring as is in a normal Gaussian image filter, and the depth standard deviation $\sigma_r$, which controls how strong pixels on different depth layers are weighted.

### 3.2 Bilateral Filter for Non-Gaussian Kernels

Consider some non-Gaussian kernel $F$ with negative values, for instance $F$ is an oriented band-pass filter used in a steerable pyramid [18], or a Gaussian derivative. As with the Gaussian bilateral filter, we would like to apply $F$ to an input image $I$, but obtain filter responses representative of the local spatial neighbourhood with nearby depth values. While the bilateral filter with Gaussian kernel can be seen as a weighted average, we cannot simply replace the kernel by $F$. For instance, the integral of a Gaussian derivative kernel is zero, and would yield a division by zero in normalization. Also, one cannot ignore part of the input by reducing corresponding weights to zero, since this introduces unwanted edge responses as if the input itself partly has zero values; our experiments in Sect. 4.1 will illustrate this point.

Instead, we propose to reduce the influence of regions in distant depth layers by smoothly incorporating the spatial image structure at the local depth layer. Using $\xi = E(z)$ to denote the depth at the output location $z$, the non-Gaussian bilateral filter with output $Q$ is written as

$$Q(z) = \sum_{x \in N(z)} F(|x - z|) O^{+}(x, E(z)) \tag{3}$$

$$O^{+}(x, \xi) = \frac{1}{W^{+}(x, \xi)} \sum_{y \in N(z)} w(|x - y|, \xi - E(y)) I(y) \tag{4}$$

with $w(d_s, d_E)$ again some weight function (which we will define in a moment), and $W^{+}(x, \xi) = \sum_{y \in N(z)} w(|x - y|, \xi - E(y))$. Here the + suffix indicates that a function operates on 3D space by extending the spatial domain with additional depth information from $E$. Throughout this paper we shall use the + suffix notation more frequently, and refer to it as an *extended* representation. Note that $Q(x)$ is not just a convolution with $F$ after applying a bilateral filter, since $O^{+}(x, \xi)$ is not only a function of $x$. But, our formulation does have the regular bilateral filter, Eq. (1), as special case when $F(x) = 1$ iff $x = 0$ and 0 otherwise.

We reformulate our filter to a 3D representation similar to the fast bilateral filter [15]. This has two benefits: One, as with the standard bilateral filter, explicit evaluation of Eq. (3) is inefficient, since filter coefficients need to be reweighted at each spatial location. The 3D representation instead offers a trade off between quality and speed [15]. Two, this 3D representation explicitly keeps filter responses at different depths separated, which will be exploited in our applications to perform depth-aware temporal filtering.

First, an extra dimension $r$ is introduced, representing possible depth values for $E(y)$ in the domain of all depth values $R$. We rewrite Eq. (4) as

$$W^{+}(x, \xi) O^{+}(x, \xi) = \sum_{r \in R} \sum_{y \in N} w(|x - y|, \xi - r) \delta(r, E(y))\; I(y) \tag{5}$$

where $\delta(r, E(y)) = 1$ iff $r = E(y)$ and 0 otherwise. Next, the term $\delta(r, E(y))$ and the 2D input image are used to define equivalent extended representations (indicated by the + suffix) $I^{+}$ for input, and $V^{+}$ to weight the input,

$$V^{+}(y, r) = \delta(r, E(y)) \tag{6}$$

$$I^{+}(y, r) = I(y) \tag{7}$$

$$\delta(r, E(y)) I(y) = V^{+}(y, r) I^{+}(y, r). \tag{8}$$

We see that $I^+(y,r)$ indeed has 3D coordinates $(y,r) = (u_y, v_y, r)$, and similarly, $V^+(y,r)$ constitutes a 3D binary mask indicating which part of the space contains valid input. We write out the terms of (5) as

$$W^+(x,\xi)O^+(x,\xi) = \sum_{r\in R}\sum_{y\in N(z)} w(|x-y|, \xi - r)\, V^+(y,r)I^+(y,r), \tag{9}$$

$$W^+(x,\xi) = \sum_{r\in R}\sum_{y\in N(z)} w(|x-y|, \xi - r)\, V^+(y,r). \tag{10}$$

Now we can recognize the nested summation as a 3D convolution over the extended representations using the weight function $w$ as a 3D kernel, which is the first step in our non-Gaussian filtering method,

$$W^+O^+ = w \otimes V^+I^+ \qquad \text{step 1:} \quad \text{3D Gauss convolution} \tag{11}$$

$$W^+ = w \otimes V^+. \tag{12}$$

If we would use the bilateral filter weight function (2), the 3D convolution will expand the local 3D neighbourhood of a target image location into regions with different depth values. Thereby, increasing depth distances result in less contribution in the convolved result. However, this kernel also blurs the original image values, inadvertently removing details from the input before the filter $F$ is applied in Eq. (3), even at regions with uniform depth that should not be affected.

Therefore, we consider a weighting function

$$w(d_I, d_E) = \begin{cases} \alpha & \text{iff } d_I = 0 \text{ and } d_E = 0, \\ G(d_I; \sigma_s) \times G(d_E; \sigma_r) & \text{otherwise,} \end{cases} \tag{13}$$

such that as $\alpha \to \infty$, the weight of the local image value $I^+(y,r)$ dominates all other weights in (9) and (10) when $y = x, r = \xi$, *and* when $V^+(y,r) = 1$. In other words, the 3D convolution will not blur the actual input values, and not affect the filter response $F$ in uniform depth regions. But the Gaussian weighting is still used in include valid input from the neighbourhood when $V^+(y,r) = 0$, i.e. at regions with missing values in the extended representation. In practice we do not explicitly evaluate (13), but instead produce the result of $\alpha \to \infty$ by applying the normal kernel first to a temporary result $\Theta^+$, and then placing back the original values to obtain the intended result. The remaining steps to apply our filter method are therefore,

$$\Theta^+ = W^+O^+/W^+ \qquad \text{step 2:} \quad \text{element-wise division} \tag{14}$$

$$O^+ = V^+I^+ + (1 - V^+)\Theta^+ \qquad \text{step 3:} \quad \text{restore valid original data} \tag{15}$$

$$Q^+ = F \otimes O^+ \qquad \text{step 4:} \quad \text{apply } F \text{ at all depth layers} \tag{16}$$

$$Q(x) = Q^+(x, E(x)) \qquad \text{step 5:} \quad \text{back-project to 2D} \tag{17}$$

The final step, (17), back-projects the extended space to the original 2D image space, which completes the evaluation of Eq. (3). Following [15], discretizing the depth dimensions $r$ into $D$ depth layers results in a fast approximation, and convolutions on the depth layers can be processed in parallel. Additionally, we downsample the image instead of expanding the spatial Gaussian kernel [15].

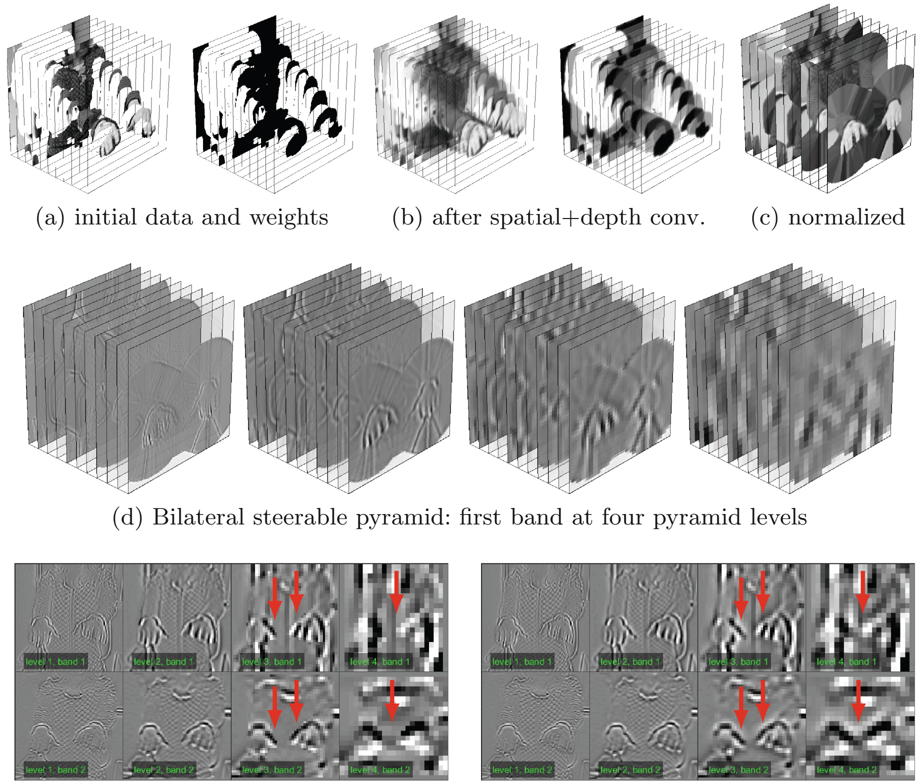
### 3.3 Depth-Aware Video Magnification

For phase-based motion magnification [1], the non-Gaussian complex steerable pyramid is used. The principle behind this approach is that small temporal changes in the spatial offset of edges translates to small temporal changes in the phases of the complex filter responses in the pyramid. Likewise, augmenting temporal phase variations results in magnifying periodic movements in the video. With a magnification factor $M$, phases $\phi_t$ of the pyramid components $p_t$ at time $t$ are augmented with respect to the temporally low-pass filtered phases $\bar{\phi}$ to obtain magnified pyramid phase $\widehat{\phi_t} = (1 + M) \cdot (\phi_t - \bar{\phi}) + \bar{\phi}$.

To exploit the depth information in the complex steerable pyramid, we apply the non-Gaussian bilateral filtering from Sect. 3.2. Figure 2 illustrates the steps to construct a bilateral steerable pyramid from an input grey scale and depth image pair $(I, E)$. First, an extended representation is created following Eq. (11)–(15). This representation is then used in the pyramid construction by applying the low-pass and complex band-pass filters of [18] to each of the depth layers, i.e. Eq. (16). The result is an extended complex steerable pyramid. The bottom row of Fig. 2 illustrates that when the extended pyramid is back-projected using the depth map $E$ (Eq. (17)), the resulting pyramid coefficients are *depth-aware*: filter responses of fore- and background edges are separated as seen by the discontinuities. In the standard pyramid, strong filter responses from the foreground (depicting two hands) 'leak' into the background, especially at higher pyramid levels where the filters have larger spatial extent.

Figure 3, depicts our proposed magnification pipeline. The phase augmentation principle is applied to components in all depth layers of the extended pyramid. However, we adapt the factor per layer with a depth-dependent function, resulting in a spatially varying magnification matte $M(x) = M_{\max} \times \exp\left(-(E(x) - \mu_d)/(2\sigma_d^2)\right)$, parametrized by $(\mu_d, \sigma_d, M_{\max})$. In the last processing step, the magnified pyramids are back-projected to 2D frames, and the matte is used to smoothly blend the magnified results of the discrete depth layers.

The recently proposed method by [2] also considers magnification of subtle motions that occur in videos with large movements, utilizing an opacity matte to blend selected regions of a magnified frame into the original unmagnified frame. For our comparison, we adapted their method to our setting which entails that (a) instead of using a tool to manually select a binary foreground region, the opacity matte $M$ is used, (b) we do not perform initial video stabilization as the camera viewpoint is already static, (c) the motion of matte $M$ itself is not magnified as we do not wish to magnify the motion of the occluding object, but rather that of the occluded region. Figure 3 also shows the difference between both approaches. The baseline introduces the depth information at the last step only. Our approach uses depth from the start to obtain and operate on a depth-aware representation. Section 4.2 shall empirically compare both approaches.



(a) initial data and weights (b) after spatial+depth conv. (c) normalized

(d) Bilateral steerable pyramid: first band at four pyramid levels

(e) Bilateral pyramid (back-projected) (f) Standard steerable pyramid

**Fig. 2.** Constructing a bilateral steerable pyramid on the frame from Fig. 1a with the steps in Sect. 3.2. (a) the input image and depth map are used to construct a 3D image representation image $I^+$ and input weight map $V^+$ by discretizing the depth into multiple layers. (b) *step 1:* Both representations are filtered in 3D (2D and the image coordinates + 1D depth coordinate). (c) *step 2, 3:* The filtered 3D image is normalized using the filtered weights, and valid input is restored. (d) *step 4:* the steerable pyramid is constructed on each discrete depth layer. Here, the result is shown at various levels in the pyramid of a single orientation band. (e) *step 5:* The 3D representation can be back-projected to a normal pyramid using depth map, resulting in an edge-aware steerable pyramid. Note how the responses of edges in the nearby hand remain within the foreground region, resulting in hard edges (e.g. see red arrows). (f) In contrast, a normal steerable pyramid induces soft object edges which 'leak' from the foreground into the surrounding background, especially at the higher pyramid levels (e.g. see red arrows). (Color figure online)

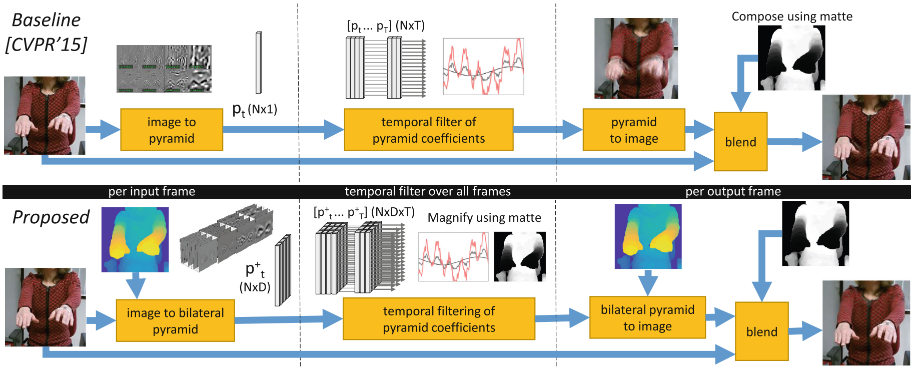

Baseline [CVPR'15]
image to pyramid
$p_t$ (Nx1)
$[p_t \dots p_T]$ (NxT)
temporal filter of pyramid coefficients
pyramid to image
Compose using matte
blend
per input frame
temporal filter over all frames
per output frame
Proposed
image to bilateral pyramid
$p^+_t$ (NxD)
$[p^+_t \dots p^+_T]$ (NxDxT)
Magnify using matte
temporal filtering of pyramid coefficients
bilateral pyramid to image
blend

**Fig. 3.** Video magnification pipelines. (Top) baseline approach from [2], which composes a magnified and unmagnified version of each frame. The composition is based on an opacity matte, based on the depth map. (Bottom) our approach instead uses the depth map directly in the pyramid construction/deconstruction.

### 3.4 Motion Measurements with a Bilateral Pyramid

In addition to magnification, another use for complex steerable filters is to measure subtle periodic motions in the video. In [31], an image is first down scaled and filtered with $B = 2$ bands for the $u$ and $v$ direction, i.e. $b \in \{0°, 90°\}$. Changes in phase can be translated to a local motion estimate $(\Delta u, \Delta v)$, as

$$\Delta u_t(u,v) = -\frac{\partial u}{\phi_t^{0°}(u,v)} \frac{\phi_t^{0°}(u,v)}{\partial t} \qquad \Delta v_t(u,v) = -\frac{\partial v}{\phi_t^{90°}(u,v)} \frac{\phi_t^{90°}(u,v)}{\partial t}. \tag{18}$$

In each equation, the first r.h.s. term is the inverse of a spatial derivative, and the second term is a temporal derivative.

For a depth-aware version, we can use the bands of our bilateral complex pyramid, using $B = 2$ bands, and select a particular layer $l$ for scale. To ensure that the spatial and temporal derivatives are depth-aware, we compute Eq. (18) in the extended space $\phi_t^{l,b+}(u,v,r)$ and obtain $\Delta u_t^+(u,v,r), \Delta v_t^+(u,v,r)$. Only afterwards are these back-projected to 2D motion maps $\Delta u_t$ and $\Delta v_t$.

## 4 Experiments

We first evaluate against inpainting techniques. Then, we introduce a novel RGB+Depth dataset targeting tremor analysis, which is an important medical application [10,12]. On this dataset we compare our depth-aware motion magnification against the state-of-the-art [2], and we show the effect of bilateral filtering on motion measurements in fore- and background.

### 4.1 Filtering Near Missing Values

Consider that we wish to convolve filter $F$ on image $I$ for which we have a binary mask $M$ whose pixel values should be ignored. This situation corresponds to the extreme case of the bilateral filter where foreground and background are far apart such that all weights are either 0 or 1. Our approach of Sect. 3.2 weighs in neighborhood values in ignored regions before applying $F$. Here we compare our approach to image inpainting techniques that intent to reconstruct the regions.

Let $g$ be a filling technique that replaces the values in the masked region, $I_{g,M} = g(I, M)$. The filled image can then be filtered with convolutional filter $F$, resulting in $I^F_{g,M} = I_{g,M} \otimes F$. Ideally the masked pixels are ignored and do not have a response at $\mathcal{R}$, i.e. at the region of pixels just outside $M$ but where the filter still covers masked out pixels. As error measure, we therefore report the L1 norm over the pixels in $\mathcal{R}$. Let $\mathbb{L}_1(g, I, M, F)$ be the norm for a particular technique $g$ on image $I$ and mask $M$ after applying filter $F$, then error($g$) is the total norm over all tested images, masks and filters, i.e.

$$\mathbb{L}_1(g, I, M, F) = \sum_{x \in \mathcal{R}} |I^F_{g,M}(x)| \qquad \text{error}(g) = \sum_{I} \sum_{M} \mathbb{L}_1(g, I, M, F). \tag{19}$$

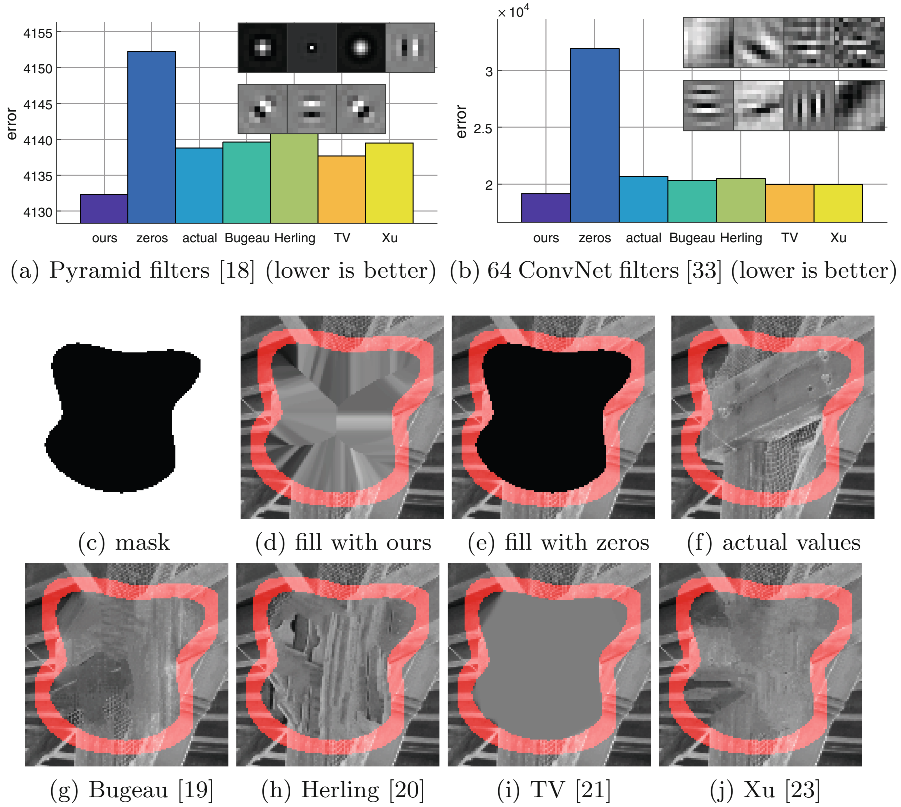
We evaluate on a public inpainting dataset [22], which contains 17 images of 640 × 480 pixels (all images are converted to grayscale), 4 image masks, and also provides on each image-mask pair state-of-the-art inpainting results for *Bugeau* [19], *Herling* [20], *Total Variation (TV)* [21] and *Xu* [23]. We also compare against replacing the missing region with *zeros*, or the *actual* pixel values (an ideal inpainting algorithm). First, we use the 7 filters used in the construction of the bilateral pyramid, and tested varying the spatial parameters $\sigma_s$, but found that $\sigma_s = 1$ performed best on the steerable pyramids features. The results in Fig. 4a show that our proposed approach results in lower errors than the other inpainting techniques. As expected, the naive approach of replacing the masked region with zeros results in strong responses near the mask border, as shown by the error plots. One of the better results is obtained with TV [21], which in fact produces quite bland areas. Indeed, even using an ideal inpainting algorithm (i.e. *actual*) would introduce more unwarranted filter responses.

To test if these results generalize, we also use 64 filters from the trained VGG convolutional neural network [32] (normalized by subtracting DC components divided by norm). Figure 4b shows that similar results were obtained.

### 4.2 Depth-Aware Motion Magnification and Measurements

The next section describes our novel tremor dataset, and then the experiments on motion magnification and motion measurement.[1] Please see the accompanying videos in the supplementary material.

[1] The bilateral pyramid, depth-aware magnification code and dataset (RGB, Depth, skeleton) can be found at https://github.com/jkooij/depthaware-momag.



(a) Pyramid filters [18] (lower is better) (b) 64 ConvNet filters [33] (lower is better)

(c) mask (d) fill with ours (e) fill with zeros (f) actual values

(g) Bugeau [19] (h) Herling [20] (i) TV [21] (j) Xu [23]

**Fig. 4.** (a)–(b) Errors for steerable pyramid and ConvNet features when filtering around a masked region. Insets show visualizations of (some of) these filters (c) one of the masks, black indicates missing values. (d)–(j) Output examples of filling methods (includes inpainting results by [22]). Red shows the evaluation region $\mathcal{R}$ of Eq. (19) where the filters will be affected by missing values. (Color figure online)

**RGB+D Tremor Dataset.** Tremors are manifestations of periodic movements in the body, and assessing their properties (frequency, amplitude) is critical for health monitoring [11,12]. Since in practice only few accelerometers can be placed on the body they are typically placed where the amplitude of the tremor is most clearly visible, e.g. on the hand and arm. Video based measuring and magnification could help discover more subtle occurrences, visualize where tremors originate or how they move trough the body, and even make objective tremor assessment possible without expensive hospital equipment.

We therefore collected a novel dataset with the Microsoft Kinect 2 to study visual tremor assessment using (1) visualization, and (2) by measuring frequency. The dataset contains 4 RGB+Depth sequences of subjects with a simulated tremor in the hand, observed with their arms extended for several seconds. This is a common task in tremor assessment, intended to induce a *postural tremor* (i.e. a tremor which occurs due to subject trying to actively maintain a certain pose) [11]. The subjects are movement scientists, experienced in working with patients of the neurology department at the Leiden University Medical Center.

Using [33], we recorded the high-res RGB video (1920 × 1090, encoded in H.264, 4:2:0 YUV), low-res depth video (512 × 450, lossless H.264, 0–4 m distance mapped to 8-bit greyscale), and the Kinect 2's estimated skeleton data [34], all at 30 fps. Afterwards, the Kinect's mapping API was used to project the recorded depth frames to the RGB image space. The alignment of the depth image with the colour images is not perfect, however: the video and depth camera have slightly different viewpoints, the recording of the depth video loses some quality due to the 8-bit greyscale conversion, and quick motions may result in motion blur in the video that is not observed in the depth. For these reasons, the depth data is pre-processed by first running a 2D median filter to remove noise, and then a 2D max filter. This extends the occluding regions and ensure that foreground in the video is also fully enclosed in the nearby regions of the depth image. The supplementary material demonstrates how depth noise, temporal misalignment, and pre-processing affect the magnification results.

**Motion Magnification Behind Moving Occluder.** On the first three sequences, each 91 frames (= 3 s), we compare our depth-aware video magnification to the the baseline approach from [2], as described in Sect. 3.3. Instead of specifying specific frequencies to magnify [1], we use the mean phase over the whole sequence as the low-pass $\bar{\phi}$ in order to magnify all periodic motion variations, and to avoid tuning temporal bandwidth parameters. The spatial deviation parameter $\sigma_s = 1$, and depth deviation $\sigma_r = 0.1\,\text{m}$.

Examples of the input image region, corresponding depth map, and the used magnification matte (which in all cases has been set to magnify the body's depth range) can be seen in Fig. 1a and b. The results of the various magnification methods are visualized as space-time slices of Fig. 1. Figure 5 shows additional single frame comparisons. On the third sequence the clothing is very dark. Here the intensity channel has been enhanced to more clearly show the details in the body. The figure illustrates that compositing the magnified and original image, as in the baseline [2], results in notable artifacts in both textured on non-textured backgrounds. We conclude that the approach in [2], which is designed to magnify foreground under heavy camera motion, does not properly magnify background behind non-static occluders. Our approach instead suppresses the artifacts.

**Motion Frequency for Overlapping Body Regions.** We applied the bilateral motion measurement on the 4th and longer sequence ($\sim$ 17 s.) to determine the vertical motion in the hand (foreground) and chest region (background) surrounding the hand. In each frame, the measured motion is averaged over a body part mask automatically extracted using the Kinect 2's built-in skeleton estimate, resulting in a single temporal signal for each body region. The time aligned groundtruth data of an accelerometer on the chest demonstrates that this sequence contains two breathing cycles of about 8.5 s., see Fig. 6a. When we

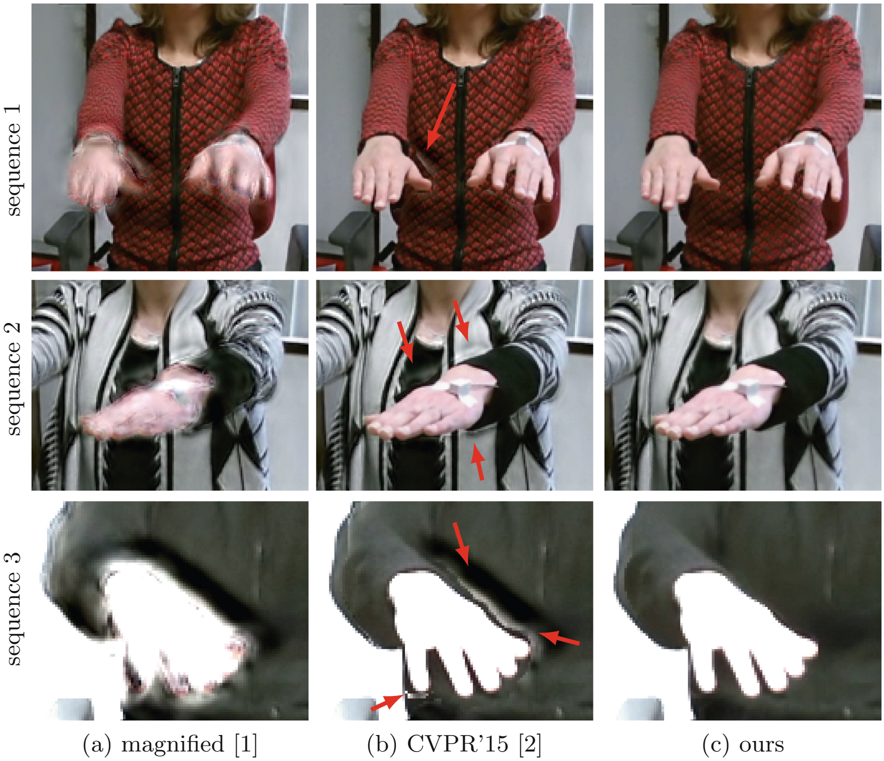
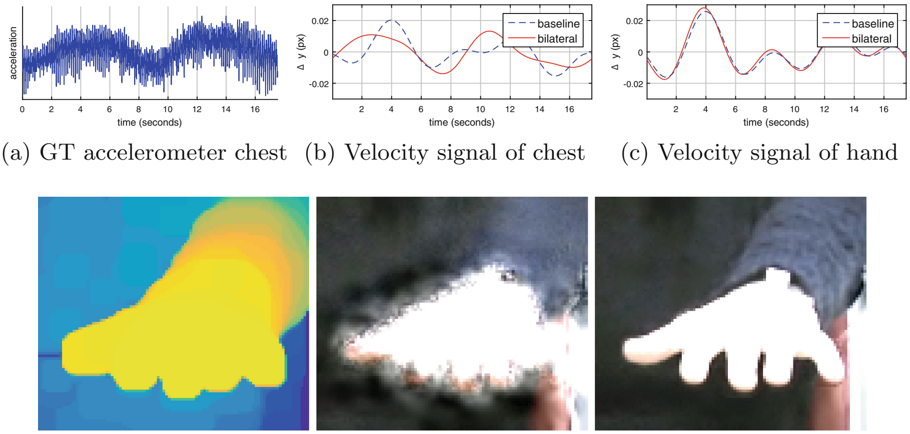


**Fig. 5.** Frames from motion magnification results on three sequences (top-to-bottom $M_{\text{max}} = 10, 3, 5$). The method of [2] blends the standard magnification result [1] with the original frame using an opacity matte, but this does not prevent unwanted artifacts of the foreground occurring the magnified background (see red arrows), even though the (unmagnified) foreground is corrected. Our approach using the bilateral filtered pyramid does avoids such artifacts. (Color figure online)

apply a low-pass filter to only keep frequencies in the 0–0.2 Hz range, we observe that without the bilateral filter the measurements in the chest are virtually the same as those in the hand, see dashed blue lines Figs. 6b and c. With bilateral filtering we obtain the same motion measurements in the foreground, but discover two periodic cycles in the background (see red lines).

The corresponding video magnification results in Fig. 6d again demonstrate that the moving foreground 'leaked' into the background. Our bilateral pyramid yields more robust phase-based measurements in such situations.



(a) GT accelerometer chest (b) Velocity signal of chest (c) Velocity signal of hand

(d) Depth, baseline, and our magnification results on sequence 4

**Fig. 6.** Measuring motion in chest behind moving hand on sequence 4. (a) Accelerometer on chest shows that there are 2 respiration cycles, each taking 9 s. (b) Low-passed velocity measurements on the chest, obtained with standard pyramid [1] (blue), and our bilateral pyramid (red). Our method measures two full up-down cycles of breathing, while the baseline shows the same motion pattern as measured in the occluding hand (c). (d) This effect is also observed when using these pyramids for motion magnification: the baseline (middle) contains movement of the hand in the background, unlike our pyramids (right). (Color figure online)

## 5 Conclusions

Our work exploits depth to make motion magnification robust against moving occluders. To construct depth-aware steerable pyramids, the bilateral filter was adapted to non-Gaussian kernels, such that filter responses can ignore local image values at distant depth layers. We proposed a simple and efficient filling technique that is less prone to introducing additional filter responses than state-of-the-art image inpainting techniques. Depth-aware motion magnification was demonstrated on a novel RGB+D dataset recorded with Microsoft Kinect 2 for tremor assessment, an important application in the medical domain. On this dataset with small motions in the background behind large motions in the foreground, we show improved qualitative motion magnification results with less visual artifacts compared to a state-of-the-art magnification baseline, which only exploits depth information as a final processing step. The bilateral pyramid also resulted in improved phase-based motion measurements.

Future work includes extending the dataset with more subjects, extract more measures used in medical practice, and investigate application to computationally efficient Riesz pyramids [25]. Our aim is to develop the explored methods into cheap and objective techniques to discover, monitor and classify tremors and other movement disorders (e.g. dystonia's) all over the body. Other uses of the non-Gaussian bilateral filter are also considered.

**Acknowledgments.** This work is part of the research programme Technology in Motion (TIM [628.004.001]), financed by the Netherlands Organisation for Scientific Research (NWO).

## References

1. Wadhwa, N., Rubinstein, M., Durand, F., Freeman, W.T.: Phase-based video motion processing. ACM Trans. Graph. **32**(4), 80:1–80:9 (2013). (Proceedings SIGGRAPH)
2. Elgharib, M.A., Hefeeda, M., Durand, F., Freeman, W.T.: Video magnification in presence of large motions. In: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 4119–4127. IEEE (2015)
3. Rubinstein, M., Wadhwa, N., Durand, F., Freeman, W.T.: Revealing invisible changes in the world. Science **339**(6119), 519–519 (2013)
4. Wu, H.Y., Rubinstein, M., Shih, E., Guttag, J., Durand, F., Freeman, W.T.: Eulerian video magnification for revealing subtle changes in the world. ACM Trans. Graph. **31**(4), 65:1–65:8 (2012). (Proceedings SIGGRAPH)
5. Davis, A., Rubinstein, M., Wadhwa, N., Mysore, G., Durand, F., Freeman, W.T.: The visual microphone: passive recovery of sound from video. ACM Trans. Graph. **33**(4), 79:1–79:10 (2014). (Proceedings of SIGGRAPH)
6. Balakrishnan, G., Durand, F., Guttag, J.: Detecting pulse from head motions in video. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3430–3437 (2013)
7. Aziz, N.A., Tannemaat, M.R.: A microscope for subtle movements in clinical neurology. Neurology **85**(10), 920–920 (2015)
8. Amir-Khalili, A., Peyrat, J.-M., Abinahed, J., Al-Alao, O., Al-Ansari, A., Hamarneh, G., Abugharbieh, R.: Auto localization and segmentation of occluded vessels in robot-assisted partial nephrectomy. In: Golland, P., Hata, N., Barillot, C., Hornegger, J., Howe, R. (eds.) MICCAI 2014, Part I. LNCS, vol. 8673, pp. 407–414. Springer, Heidelberg (2014)
9. Davis, A., Bouman, K.L., Chen, J.G., Rubinstein, M., Durand, F., Freeman, W.T.: Visual vibrometry: estimating material properties from small motions in video. In: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 5335–5343. IEEE (2015)
10. Bain, P.G.: Parkinsonism & related disorders. Tremor **13**, S369–S374 (2007)
11. Deuschl, G., Bain, P., Brin, M.: Consensus statement of the movement disorder society on tremor. Mov. Disord. **13**(S3), 2–23 (1998)
12. Schwingenschuh, P., Katschnig, P., Seiler, S., Saifee, T.A., Aguirregomozcorta, M., Cordivari, C., Schmidt, R., Rothwell, J.C., Bhatia, K.P., Edwards, M.J.: Moving toward laboratory-supported criteria for psychogenic tremor. Mov. Disord. **26**(14), 2509–2515 (2011)
13. Fattal, R.: Edge-avoiding wavelets and their applications. ACM Trans. Graph. **28**(3), 22:1–22:10 (2009). (Proceedings SIGGRAPH)
14. He, K., Sun, J., Tang, X.: Guided image filtering. IEEE Trans. Pattern Anal. Mach. Intell. **35**(6), 1397–1409 (2013)
15. Paris, S., Durand, F.: A fast approximation of the bilateral filter using a signal processing approach. Int. J. Comput. Vis. **81**(1), 24–52 (2009)
16. Tomasi, C., Manduchi, R.: Bilateral filtering for gray and color images. In: Sixth International Conference on Computer Vision, pp. 839–846. IEEE (1998)

17. Freeman, W.T., Adelson, E.H.: The design and use of steerable filters. IEEE Trans. Pattern Anal. Mach. Intell. **9**, 891–906 (1991)
18. Simoncelli, E.P., Freeman, W.T.: The steerable pyramid: a flexible architecture for multi-scale derivative computation. In: ICIP, p. 3444. IEEE (1995)
19. Bugeau, A., Bertalmío, M., Caselles, V., Sapiro, G.: A comprehensive framework for image inpainting. IEEE Trans. Image Process. **19**(10), 2634–2645 (2010)
20. Herling, J., Broll, W.: Pixmix: A real-time approach to high-quality diminished reality. In: 2012 IEEE International Symposium on Mixed and Augmented Reality (ISMAR), pp. 141–150. IEEE (2012)
21. Getreuer, P.: Total variation inpainting using split bregman. Image Process. Line **2**, 147–157 (2012)
22. Tiefenbacher, P., Bogischef, V., Merget, D., Rigoll, G.: Subjective and objective evaluation of image inpainting quality. In: 2015 IEEE International Conference on Image Processing (ICIP), pp. 447–451. IEEE (2015)
23. Xu, Z., Sun, J.: Image inpainting by patch propagation using patch sparsity. IEEE Trans. Image Process. **19**(5), 1153–1165 (2010)
24. Liu, C., Torralba, A., Freeman, W.T., Durand, F., Adelson, E.H.: Motion magnification. ACM Trans. Graph. (Proceedings SIGGRAPH) **24**(3), 519–526 (2005)
25. Wadhwa, N., Rubinstein, M., Durand, F., Freeman, W.T.: Riesz pyramids for fast phase-based video magnification. In: 2014 IEEE International Conference on Computational Photography (ICCP), pp. 1–10. IEEE (2014)
26. Bai, J., Agarwala, A., Agrawala, M., Ramamoorthi, R.: Selectively de-animating video. ACM Trans. Graph. **31**(4), 66:1–66:10 (2012). (Proceedings SIGGRAPH)
27. Malleson, C., Bazin, J.C., Wang, O., Bradley, D., Beeler, T., Hilton, A., Sorkine-Hornung, A.: Facedirector: continuous control of facial performance in video. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 3979–3987 (2015)
28. Shankar Nagaraja, N., Schmidt, F.R., Brox, T.: Video segmentation with just a few strokes. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 3235–3243 (2015)
29. Bai, J., Agarwala, A., Agrawala, M., Ramamoorthi, R.: User-assisted video stabilization. Comput. Graph. Forum **33**(4), 61–70 (2014)
30. Efros, A.A., Leung, T.K.: Texture synthesis by non-parametric sampling. In: The Proceedings of the Seventh IEEE International Conference on Computer Vision, vol. 2, pp. 1033–1038. IEEE (1999)
31. Chen, J.G., Wadhwa, N., Cha, Y.J., Durand, F., Freeman, W.T., Buyukozturk, O.: Modal identification of simple structures with high-speed video using motion magnification. J. Sound Vibr. **345**, 58–71 (2015)
32. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition (2014). arXiv preprint arXiv:1409.1556
33. Kooij, J.F.P.: SenseCap: synchronized data collection with Microsoft Kinect2 and LeapMotion. In: Proceedings of the 22nd ACM International Conference on Multimedia. ACM (2016, to appear)
34. Shotton, J., Sharp, T., Kipman, A., Fitzgibbon, A., Finocchio, M., Blake, A., Cook, M., Moore, R.: Real-time human pose recognition in parts from single depth images. Commun. ACM **56**(1), 116–124 (2013)