Real-Time Non-Rigid Multi-Frame Depth Video Super-Resolution

Kassem Al Ismaeil[1], Djamila Aouada[1], Thomas Solignac[2], Bruno Mirbach[2], Björn Ottersten[1]

[1]Interdisciplinary Centre for Security, Reliability, and Trust,
University of Luxembourg.
{kassem.alismaeil,djamila.aouada,bjorn.ottersten}@uni.lu
[2] Advanced Engineering Department, IEE S.A.
{thomas.solignac,bruno.mirbach}@iee.lu

Abstract

This paper proposes to enhance low resolution dynamic depth videos containing freely non–rigidly moving objects with a new dynamic multi–frame super–resolution algorithm. Existent methods are either limited to rigid objects, or restricted to global lateral motions discarding radial displacements. We address these shortcomings by accounting for non–rigid displacements in 3D. In addition to 2D optical flow, we estimate the depth displacement, and simultaneously correct the depth measurement by Kalman filtering. This concept is incorporated efficiently in a multi–frame super–resolution framework. It is formulated in a recursive manner that ensures an efficient deployment in real–time. Results show the overall improved performance of the proposed method as compared to alternative approaches, and specifically in handling relatively large 3D motions. Test examples range from a full moving human body to a highly dynamic facial video with varying expressions.

1. Introduction

The recent developments in depth sensing technologies, be it time–of–flight (ToF) cameras or structured light cameras, have seen the explosion of their applications in gaming, automotive sensing, surveillance, medical care, and many more. The major problem of these sensors is their high contamination with noise and low spatial resolution. In addition, in the case of large distances between the sensor and the scene of interest, a similar effect is observed even by using a relatively high resolution depth sensor.

In this paper, we consider dynamic depth videos with one or multiple moving objects deforming non–rigidly. This is a very typical scenario encountered in people sensing, cloth deformation, hand gesture, variations of facial expressions, to name a few. Such scenes are more challenging than static scenes. Indeed, in addition to challenges due to noise and outliers, non–rigid deformations in 3D cause occlusions, which result in missing data, and in undesired holes.

This work was supported by the National Research Fund, Luxembourg, under the CORE project C11/BM/1204105/FAVE/Ottersten.

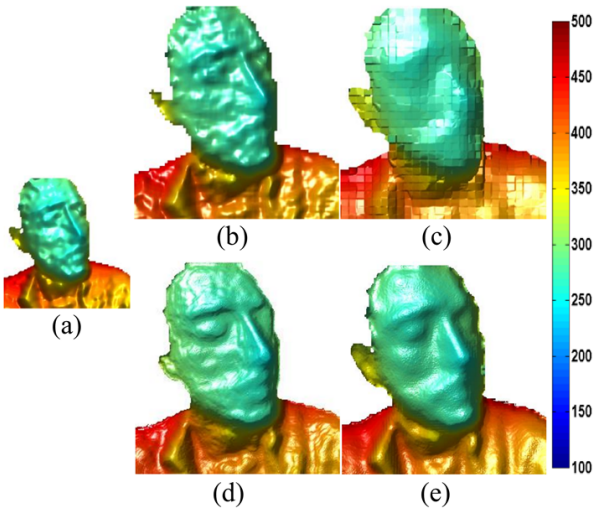

Figure 1. Different super–resolution methods applied to a real low resolution dynamic depth sequence captured with a ToF camera with SR scale factor of $r = 4$. (a) Low resolution depth frame. (b) Bicubic interpolation. (c) Patch Based Single Image Super Resolution (*SISR*) [5]. (d) Upsampling for Precise Super Resolution (*UP-SR*) [4]. (e) Proposed algorithm (50 ms per frame).

Super-resolution (SR) algorithms have been proposed as a solution to this problem. Two categories of algorithms may be distinguished; multi–frame SR which use multiple frames in an inverse problem formulation to reconstruct one high resolution frame [16, 7, 4]. The second category is known as single–image SR. It is based on dictionary learning and a heavy training [5, 12].

In [4], we proposed the first dynamic multi-frame depth SR. This algorithm is, however, limited to lateral motions,

and fails in the case of radial deformations. Moreover, it is not practical due to a heavy cumulative motion estimation process applied to a certain number of frames buffered in the memory. Alternatively, a recursive formulation may be thought of as in [15] where an iterative SR was proposed based on a block affine motion model resulting in a relatively efficient processing. This, however, is not applicable to non–lateral motions.
Earlier attempts for recursive SR approaches have proposed to use a Kalman filter formulation [8, 10, 9, 13, 18]. These methods work only under two conditions: constant translational motion between low resolution frames which represents the system motion model (i.e. transition matrix), and intensity consistency assumption between each pair of images in the video sequence. In the case of dynamic depth videos, these assumptions are not always valid. Indeed, for such videos, individual pixel motions have to be tracked through the video. A local motion model such as a dense 2D optical flow as in [4] is not sufficient, it is necessary to account for the full 3D motion in the SR reconstruction, known as scene flow, or the 2.5D motion, known as range flow.

For a reduced complexity we herein propose to approximate range flow by estimating radial motions on top of the 2D optical flow. Moreover, we propose a recursive depth multi-frame SR algorithm by using multiple Kalman filters. To ensure efficiency, we propose to treat a video as a set of one–dimensional signals. By so doing, we show that we reach an approximation of range flow; which enables us to take radial deformations into account in the SR estimation. To adequately preserve the smoothness properties of the depth surface, and remove noise and blur without over smoothing, we propose to use a multi–level version of the iterative bilateral total variation regularization given in [11]. In summary, the contribution of this paper is a new mutli–frame depth SR algorithm which has the following properties: 1) Recursive, hence, suitable for real–time applications. 2) Robust to radial motions without explicitly computing range flow. 3) Accurate depth video reconstruction thanks to the proposed multi–level iterative bilateral regularization. An overview of the proposed algorithm is shown in Figure 2.

The remainder of the paper is organized as follows: Section 2 gives the problem for depth video super–resolution. Section 3 explains the proposed concept for handling radial motion within the super–resolution framework. The proposed recursive depth video SR algorithm is presented in Section 4. Quantitative and qualitative evaluations and comparisons with other approaches are given in Section 5. Finally, the conclusion is given in Section 6.
The following notations will be considered: bold small letters correspond to vectors. Bold capital letters denote matrices. Italic letters are scalars. $\mathtt{p}_t$ denotes a pixel position on image plane at instant t, and $\mathtt{m}_t$ denotes the corresponding 2D optical flow at t.

2. Background and Problem Formulation

We briefly review the problem of multi–frame SR of dynamic depth videos and highlight the challenges that remain untackled by existing approaches. Let us consider a video of N observed low resolution (LR) depth frames of a dynamically deforming depth scene $\mathcal{F}$ acquired using a depth sensor, ToF or structured light. The scene is assumed to contain one or multiple moving objects. Each LR frame $\mathbf{g}_t$, $t = 1, \cdots, N$, is represented by a column vector of size $(m \times 1)$ corresponding to the lexicographic ordering of frame pixels. The objective of depth SR is to reconstruct a higher resolution (HR) depth video $\{\mathbf{f}_t, t = 1, \cdots, N\}$, where each frame is of size $(n \times 1)$ with $\frac{n}{m} = r \in \mathbb{N}^*$ being the SR scale factor. The classical multi–frame depth SR problem may be simplified by reconstructing one HR frame at a time, referred to as reference frame, by using the observed video. Therefore, if the reference time is t_0, then the problem is to reconstruct $\mathbf{f}_{t_0}$ using the $N' = (N - t_0 + 1)$ preceding measurements. The operation may be repeated for $t_0 = 1, \cdots, N$. A noisy LR observation is modelled as follows:

$$\mathbf{g}_t = \mathbf{DHM}_{t_0}^t \mathbf{f}_{t_0} + \mathbf{n}_t, \ t_0 \leq t \text{ and } t, t_0 \in [1, N] \subset \mathbb{N}^*, \tag{1}$$

where $\mathbf{D}$ is a known constant downsampling matrix of dimension $(m \times n)$. The system blur is represented by the time and space invariant matrix $\mathbf{H}$. The $(n \times n)$ matrices $\mathbf{M}_{t_0}^t$ correspond to the motion between $\mathbf{f}_{t_0}$ and $\mathbf{g}_t$ before their downsampling. The vector $\mathbf{n}_t$ is an additive white noise at time instant t. Without loss of generality, both $\mathbf{H}$ and $\mathbf{M}_{t_0}^t$ are assumed to be block circulant matrices, so they are commutative. As a result, the estimation of $\mathbf{f}_{t_0}$ may be decomposed into two steps; estimation of a blurred HR image, followed by a deblurring step.

While the *LidarBoost* algorithm [16] is a reference method for multi–frame depth SR, it is only applicable to static scenes for object scanning. The *UP-SR* algorithm in [4] is, so far, the only depth multi–frame SR proposed for dynamic scenes. This algorithm is based on two key components. The first one is to densely upsample the observed LR sequence prior to any operation. This is shown to ensure a more accurate registration of frames. The resulting r–times upsampled image is defined as $\mathbf{g}_t \uparrow = \mathbf{U} \cdot \mathbf{g}_t$, where $\mathbf{U}$ is an $(n \times m)$ upsampling matrix. The second component of *UP-SR* is to use a cumulative motion compensation approach between the reference frame and all observations. This operation starts by estimating the motion between consecutive frames, using classical dense 2D optical flow estimation between the upsamled versions $\mathbf{g}_{t-1} \uparrow$

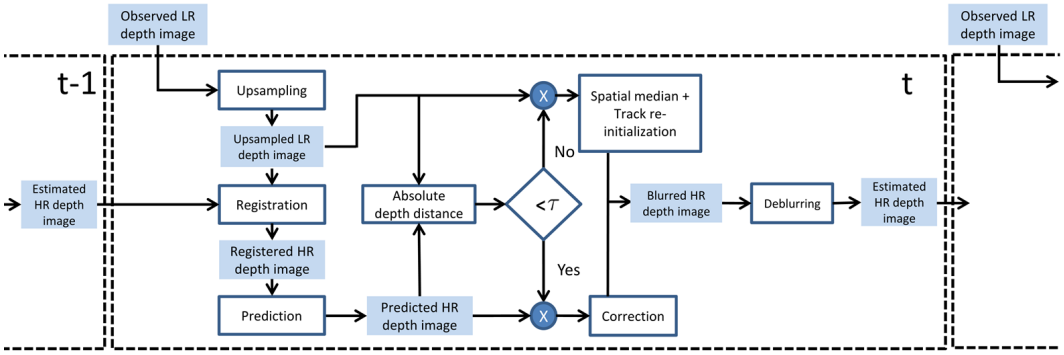

Figure 2. Flow chart of the proposed multi–frame depth super–resolution algorithm for dynamic depth videos containing one or multiple non–rigidly deforming objects.

and $\mathbf{g}_t \uparrow$, namely,

$$\hat{\mathbf{M}}_{t-1}^{t} = \arg\min_{\mathbf{M}} \Psi\left(\mathbf{g}_{t-1}\uparrow, \mathbf{g}_t \uparrow, \mathbf{M}\right), \tag{2}$$

where Ψ is a dense optical flow-related cost function and

$$\mathbf{g}_t \uparrow = \mathbf{M}_{t-1}^{t}\mathbf{g}_{t-1}\uparrow + \boldsymbol{\delta}_t. \tag{3}$$

The vector $\boldsymbol{\delta}_t$ is referred to as the innovation image. It contains novel points appearing, or disappearing due to occlusions or large motions. This innovation is assumed in [4] to be negligible. In addition, similarly to [8], for analytical convenience, it is assumed that all pixels in $\mathbf{g}_t \uparrow$ originate from pixels in $\mathbf{g}_{t-1} \uparrow$ in a one to one mapping. Therefore, each row in $\mathbf{M}_{t-1}^{t}$ contains 1 for each position corresponding to the address of the source pixel in $\mathbf{g}_{t-1} \uparrow$. This assumption of bijectiveness implies that the matrix $\hat{\mathbf{M}}_{t-1}^{t}$ is assumed to be an invertible permutation, s.t., $[\hat{\mathbf{M}}_{t-1}^{t}]^{-1} = \hat{\mathbf{M}}_{t}^{t-1}$. Furthermore, its estimate leads to the following registration to $\mathbf{g}_{t-1} \uparrow$:

$$\overline{\mathbf{g}}_{t}^{t-1}\uparrow = \hat{\mathbf{M}}_{t}^{t-1}\mathbf{g}_t \uparrow. \tag{4}$$

Using a cumulative motion compensation approach, the registration of a non–consecutive frame $\mathbf{g}_t \uparrow$ to the reference $\mathbf{g}_{t_0} \uparrow$ is achieved as follows:

$$\overline{\mathbf{g}}_{t}^{t_0}\uparrow = \hat{\mathbf{M}}_{t}^{t_0}\mathbf{g}_t \uparrow = \underbrace{\hat{\mathbf{M}}_{t_0+1}^{t_0}\cdots\hat{\mathbf{M}}_{t}^{t-1}}_{(t-t_0)\text{ times}}\cdot\mathbf{g}_t \uparrow. \tag{5}$$

Choosing the upsampling matrix $\mathbf{U}$ to be the transpose of $\mathbf{D}$, the product $\mathbf{UD} = \mathbf{A}$ gives a block circulant matrix $\mathbf{A}$ that defines a new blurring matrix $\mathbf{B} = \mathbf{AH}$. Therefore, the estimation of $\mathbf{f}_{t_0}$ starts by estimating its blurred version $\mathbf{h}_{t_0} = \mathbf{B}\mathbf{f}_{t_0}$. The data model in (1) becomes

$$\overline{\mathbf{g}}_{t}^{t_0}\uparrow = \mathbf{h}_{t_0} + \boldsymbol{\nu}_t,\ t_0 \le t \text{ and } t, t_0 \in [1, N] \subset \mathbb{N}^*, \tag{6}$$

where $\boldsymbol{\nu}_t = \hat{\mathbf{M}}_{t}^{t_0}\mathbf{U}\cdot\mathbf{n}_t$ is an additive noise vector of length n. It is assumed to be independent and identically distributed. Using an L_1–norm, the blurred estimate is found by pixel–wise temporal median filtering of the upsampled registered LR observations such as:

$$\hat{\mathbf{h}}_{t_0} = \arg\min_{\mathbf{h}_{t_0}} \sum_{t=t_0}^{N} \|\mathbf{h}_{t_0} - \overline{\mathbf{g}}_{t}^{t_0}\uparrow\|_1 = \text{med}_t\{\overline{\mathbf{g}}_{t}^{t_0}\uparrow\}_{t=t_0}^{N}. \tag{7}$$

Then, as a second step, follows an image deblurring to recover $\hat{\mathbf{f}}_{t_0}$ from $\hat{\mathbf{h}}_{t_0}$. The robustness of the *UP-SR* algorithm in handling large motions is achieved thanks to the cumulative motion approach combined with upsampling, as has been shown experimentally in [4]. However, as described above, the only considered motions are lateral motions using 2D dense optical flow. Radial displacements in the depth direction, often encountered in depth sequences, are therefore not handled. Moreover, the *UP-SR* registration step is based on a heavy cumulative motion estimation which makes this algorithm not suitable for real–time applications.

3. Range Flow Approximation

We argue that the above mentioned challenges may be resolved by incorporating the 2.5D version of dense optical flow [20], known as range flow, in the *UP-SR* framework. The direct computation of range flow can be complex. Instead of its direct computation, we propose an approximation by decomposing range flow into 2D optical flow and a filtered radial motion.

3.1. Flow Decoupling

In order to address the problem of radial motions, it is important to consider the full 3D motion per pixel. At a time instant t, and for a pixel position $\mathrm{p}_t = (x_t, y_t)$ on the sensor image plane, the depth surface $\mathcal{F}$ can be defined as the following mapping:

$$\begin{aligned}\mathcal{F}: \quad \mathbb{R}^2 \times \mathbb{N} &\to \mathbb{R}^3 \\ \mathrm{p}_t &\mapsto (x_t, y_t, z_t(x_t, y_t)).\end{aligned} \tag{8}$$

The deformation of the surface $\mathcal{F}$ from $(t_0 - 1)$ to t_0 takes the point $\mathtt{p}_{t_0-1}$ to a new position $\mathtt{p}_{t_0}$. Given $u_{t_0} = \frac{\partial x_t}{\partial t}\big|_{t_0}$ and $v_{t_0} = \frac{\partial y_t}{\partial t}\Big|_{t_0}$, the vector $\ell = (u_{t_0}, v_{t_0}, 1)^T$ represents the direction of the displacement from $\mathtt{p}_{t_0-1}$ to $\mathtt{p}_{t_0}$. The surface deformation may then be expressed through the derivative of $\mathcal{F}$ following the direction ℓ resulting in a range flow $(u_{t_0}, v_{t_0}, w_{t_0})$ where the lateral displacement is $\mathtt{m}_{t_0} = (u_{t_0}, v_{t_0})$ and the radial displacement in the depth direction is $w_{t_0} = \frac{\partial z_t}{\partial t}|_{t_0}$.

Applying the gradient constraint on the depth total derivative, we find the range flow constraint as first proposed in [20], and defined as follows:

$$u_{t_0} \left.\frac{\partial z_t}{\partial x_t}\right|_{t_0} + v_{t_0} \left.\frac{\partial z_t}{\partial y_t}\right|_{t_0} + w_{t_0} = \frac{dz_t}{dt}. \tag{9}$$

In this work we propose to decouple $\mathtt{m}_{t_0}$ from the radial displacement w_{t_0}. We compute $\mathtt{m}_{t_0}$ using available approaches for 2D optical flow estimation. We compute the 2D optical flow using the low resolution 2D intensity images associated with the considered depth sensor. Note that the intensity (amplitude) images provided by the ToF camera can not be used directly. Their values differ significantly depending on the integration time and object distance from the camera. Thus, in order to guarantee an accurate registration, we apply a standardization step similar to the one proposed in [17] prior to motion estimation, see Figure 3. If the intensity images are not available (e.g. using synthetic data) the 2D optical flow can be directly estimated using the depth images after a preprocessing step with a bilateral filter. The bilateral filter is only used in the preprocessing step while the original depth data is mapped in the registration step. We define the registered depth image from $(t_0 - 1)$ to t_0 as $\bar{z}^{t_0}_{t_0-1}$. Consequently, the radial displacement w_{t_0} may be approximated by the temporal difference between depth values, i.e.,

$$w_{t_0} \approx z_{t_0}(\mathtt{p}_{t_0}) - \bar{z}^{t_0}_{t_0-1}(\mathtt{p}_{t_0}). \tag{10}$$

This first approximation of w_{t_0} is an initial value that requires further refinement directly accounting for the system noise. We propose to do that using tracking with a Kalman filter as detailed in Section 3.2.

3.2. Refinement by Filtering

Let us start by simplifying the notation as $z_t(\mathtt{p}_t) \equiv z_t$. Since, by definition, we have $z_{t-1}(\mathtt{p}_{t-1}) = \bar{z}^t_{t-1}$, then we may write $\bar{z}^t_{t-1}(\mathtt{p}_t) \equiv z_{t-1}$. We consider the following state vector:

$$\mathbf{s}_t = \begin{pmatrix} z_t \\ w_t \end{pmatrix}, \tag{11}$$

where both the depth measurement and the radial displacement are to be filtered. To apply the Kalman filter, one needs

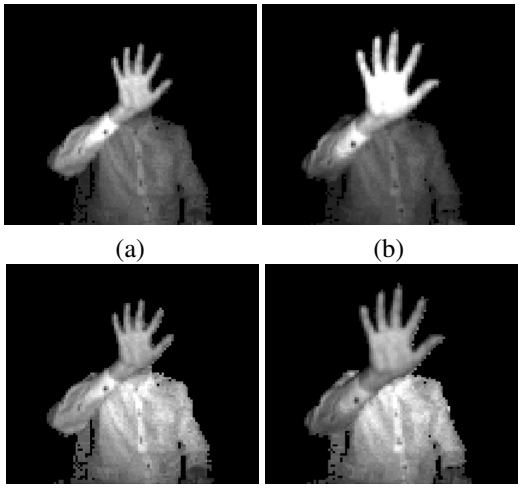

Figure 3. Correcting the amplitude images using a standardization step [17] . We can see in (a) and (b) the original amplitude images for a dynamic scene containing a moving hand towards the camera where the intensity (amplitude) values differ significantly depending on the object distance from the camera. The corrected amplitude images for the same scene are presented in (c) and (d), where the intensity consistency is preserved.

to introduce a Gaussian system; so a noisy depth observation may be modelled as

$$\tilde{z}_t = \mathbf{b} \cdot \mathbf{s}_t + n_t, \tag{12}$$

where the observation vector is $\mathbf{b} = (1, 0)^T$, and the observation noise n_t is Gaussian with the variance σ_n^2, i.e., $n_t \sim \mathcal{N}(0, \sigma_n^2)$. We assume a constant velocity model with an acceleration γ_t following a Gaussian distribution $\gamma_t \sim \mathcal{N}(0, \sigma_a^2)$. The dynamic model is then defined as

$$\begin{cases} z_t = z_{t-1} + w_{t-1}\Delta t + \frac{1}{2}\gamma_t \Delta t^2. \\ w_t = w_{t-1} + \gamma_t \Delta t. \end{cases}, \tag{13}$$

which can be rewritten as:

$$\mathbf{s}_t = \mathbf{K}\mathbf{s}_{t-1} + \boldsymbol{\gamma}_t, \tag{14}$$

where $\mathbf{K} = \begin{pmatrix} 1 & \Delta t \\ 0 & 1 \end{pmatrix}$, and $\boldsymbol{\gamma}_t = \gamma_t \begin{pmatrix} \frac{1}{2}\Delta t^2 \\ \Delta t \end{pmatrix}$ is the process error which is white Gaussian with the covariance

$$\mathbf{Q} = \sigma_a^2 \Delta t^2 \begin{pmatrix} \Delta t^2/4 & \Delta t/2 \\ \Delta t/2 & 1 \end{pmatrix}. \tag{15}$$

Using the standard Kalman equations, the prediction is achieved as

$$\begin{cases} \hat{\mathbf{s}}_{t|t-1} = \mathbf{K}\mathbf{s}_{t-1|t-1}, \\ \hat{\mathbf{P}}_{t|t-1} = \mathbf{K}\mathbf{P}_{t-1|t-1}\mathbf{K}^T + \mathbf{Q}. \end{cases} \tag{16}$$

The error in the prediction of $\hat{\mathbf{s}}_{t|t-1}$ is corrected using the observed measurement $\tilde{z}_t$. This error is considered as the

difference between the prediction and the observation, and weighted using the Kalman gain matrix $\mathbf{G}_{t|t}$ which is calculated as follows:

$$\mathbf{G}_{t|t} = \hat{\mathbf{P}}_{t|t-1}\mathbf{b}^T \left(\mathbf{b}\hat{\mathbf{P}}_{t|t-1}\mathbf{b}^T + \sigma_n^2\right)^{-1}. \quad (17)$$

The corrected state vector $\mathbf{s}_{t|t} = \begin{pmatrix} z_{t|t} \\ w_{t|t} \end{pmatrix}$ and corrected error covariance matrix $\mathbf{P}_t$ are computed as follows:

$$\begin{cases} \mathbf{s}_{t|t} = \hat{\mathbf{s}}_{t|t-1} + \mathbf{G}_{t|t}\left(\tilde{z}_t - \mathbf{b}\hat{\mathbf{s}}_{t|t-1}\right), \\ \mathbf{P}_{t|t} = \hat{\mathbf{P}}_{t|t-1} - \mathbf{G}_{t|t}\mathbf{b}\hat{\mathbf{P}}_{t|t-1}, \end{cases} \quad (18)$$

This per pixel filtering is extended to all the depth frame and incorporated in the SR framework in Section 4.

4. Proposed Recursive Depth Video Super-Resolution

In what follows, we define a recursive multi–frame super–resolution algorithm by incorporating the Kalman filtering framework of Section 3.2 to the dynamic depth video SR problem. In addition to handling radial motions, and in order to properly preserve non–rigidity, we propose to recursively filter each pixel trajectory separately by assuming that all trajectories are independent. This assumption requires a corrective step to bring back the correlation between neighbouring pixels from the original depth surface $\mathcal{F}$. To that end, we use a maximum a posteriori (MAP) estimation where we propose a multi–level iterative bilateral total variation (TV) regularization. The advantage of the processing per pixel is to keep the exact same formulation as in Section 3.2; hence, all the required matrix inversions will be for (2×2) matrices. The burden of traditional Kalman filter–based SR as in [8] will consequently be avoided. For a recursive multi–frame SR algorithm, instead of using the whole video sequence of length N to recover one frame, we use the preceding recovered frame $\hat{\mathbf{f}}_{t-1}$ to estimate $\mathbf{f}_t$ from the current upsampled observation $\mathbf{g}_t \uparrow$.

Similarly to the *UP–SR* algorithm, we estimate $\mathbf{f}_t$ in two steps; first, finding a blurred version $\hat{\mathbf{h}}_t$ as the result of the Kalman filtering, then a deblurred version $\hat{\mathbf{f}}_t$ as the result of the MAP iterative regularization.

4.1. Blurred Estimation

To extend the range flow approximation of Section 3 to a full frame, the point p_t is now considered as an element of a grid constituting a discrete sampling of $\mathbb{R}^2$. We, thus, end up with discrete positions $\mathrm{p}_t^{\mathtt{i}} = (x_t^{\mathtt{i}}, y_t^{\mathtt{i}})$ such that $\mathtt{i} \in [1, n]$. We define the depth image at t as the column vector of all the blurred depth values $z_t(\mathrm{p}_t^{\mathtt{i}})$, and write $\mathbf{h}_t = [z_t(\mathrm{p}_t^{\mathtt{i}})], \forall \mathtt{i}$. The obtained motion vectors are further scaled using the SR factor r. The scaled motion vectors are then used in order to register the depth images $\hat{\mathbf{f}}_{t-1}$ and $\mathbf{g}_t \uparrow$, resulting in $\bar{\mathbf{f}}_{t-1}^t$. The registration step reorders the pixels in order to have a correspondence that enables a direct pixel–wise filtering over time. Moreover, to apply the Kalman filter of Section 3.2, one needs to define a Gaussian system similar to the one defined by (12) and (14). The observation model in (12) is applicable to the SR data model in (6) under the assumption of a zero mean additive white Gaussian noise. The dynamic model in (14) is actually equivalent to the model in (3), and one can prove that the innovation is related to the depth displacement $w_{t-1}^{\mathtt{i}}$ and acceleration uncertainty $\gamma_t^{\mathtt{i}}$ of the pixel $\mathrm{p}_t^{\mathtt{i}}$ by the following equation:

$$\boldsymbol{\delta}_t(\mathtt{i}) = w_{t-1}^{\mathtt{i}}\Delta t + \frac{1}{2}\gamma_t^{\mathtt{i}}(\Delta t)^2. \quad (19)$$

The result of the n joint filters run in parallel is the blurred depth image estimate $\hat{\mathbf{h}}_t$.
Furthermore, in order to separate background from foreground depth pixels, and tackle the problem of flying pixels, especially around edges we define a fixed threshold τ such that:

$$\begin{cases} \text{Continue the track} & \text{if } |\tilde{z}_t - \hat{z}_{t|t-1}| < \tau; \\ \text{New track \& spatial median} & \text{if } |\tilde{z}_t - \hat{z}_{t|t-1}| \geqslant \tau. \end{cases}$$

The choice of the threshold value τ is related to the type of the used depth sensor and the level of the sensor–specific noise. In order to correct the artifacts due to this one–dimensional processing of an image, we propose a multi-level iterative bilateral TV deblurring step as described in the next section.

4.2. Multi–Level Iterative Bilateral TV Deblurring

In order to estimate the deblurred high resolution depth image $\mathbf{f}_t$ from $\hat{\mathbf{h}}_t$, we apply the following MAP deblurring framework:

$$\hat{\mathbf{f}}_t = \underset{\mathbf{f}_t}{\operatorname{argmin}}\Big(\|\mathbf{B}\mathbf{f}_t - \hat{\mathbf{h}}_t\|_1 + \lambda\Gamma(\mathbf{f}_t)\Big), \quad (20)$$

where λ is the regularization parameter, and $\mathbf{B}$ is the blurring matrix. We choose to use a bilateral TV regularizer [11] defined as:

$$\Gamma(\mathbf{f}_t) = \sum_{i=-I}^{i=I}\sum_{j=-J}^{j=J} \alpha^{|i|+|j|} \parallel \mathbf{f}_t - \mathbf{S}_x^i\mathbf{S}_y^j\mathbf{f}_t \parallel_1 . \quad (21)$$

The matrices $\mathbf{S}_x^i$ and $\mathbf{S}_y^j$ are shifting matrices which shift $\mathbf{f}_t$ by i, and j pixels in the horizontal and vertical directions, respectively. The scalar $\alpha \in]0, 1]$ is the base of the exponential kernel which controls the speed of decay [3]. In order to effectively deblur $\hat{\mathbf{h}}_t$ while keeping the details of $\mathbf{f}_t$ without over smoothing, we apply the MAP estimation in (20) where we propose to use a multi–level version in a similar fashion as in [14, 19, 6]. Combined with a

steepest descent numerical solver, the proposed solution is described by the following pseudo–code:

for $l = 1, \cdots, L$
for $k = 1, \cdots, K$

$$\hat{\mathbf{f}}_{k,l} = \hat{\mathbf{f}}_{(k-1),l} - \beta \left\{ \mathbf{B}^T sign\left(\mathbf{B}\hat{\mathbf{f}}_{(k-1),l} - \mathbf{h}_t\right) + \frac{\lambda}{2^l} \sum_{i=-I}^{i=I} \sum_{j=-J}^{j=J} \alpha^{|i|+|j|} [\mathbf{I} - \mathbf{S}_y^{-j}\mathbf{S}_x^{-i}] sign\left(\hat{\mathbf{f}}_{(k-1),l} - \mathbf{S}_x^i \mathbf{S}_y^j \hat{\mathbf{f}}_{(k-1),l}\right) \right\}$$

end for
$\mathbf{h}_t \longleftarrow \hat{\mathbf{f}}_{K,l}$
end for

The parameter β is a scalar which represents the step size in the direction of the gradient, and $\mathbf{I}$ is the identity matrix and $sign(\cdot)$ is the sign function. In our experiments, we used three levels with $L = 3$, and seven iterations per level with $K = 7$.

5. Experimental Results

In this section, we evaluate the performance of the proposed algorithm using: (i) synthetic depth videos, and (ii) real depth videos of dynamic scenes captured by a ToF camera (pmd CamBoard nano). We show the effectiveness of the proposed algorithm as compared to state–of–art methods where we provide quantitative and qualitative evaluations.

5.1. Synthetic Data

In order to provide a quantitative evaluation, we first start with a simple and fully controlled set–up. We use a generated sequence of 20 depth frames of a synthetic hand moving radially with respect to the camera (5 cm difference between each two successive frames, and $\Delta t = 0.1$ seconds). We downsample the sequence with a scale factor of $r = 2$, and $r = 4$. These sequences are further degraded with additive noise with σ varying from 10 to 80 mm. The created LR noisy depth sequences are then super–resolved using the proposed algorithm with, $r = 1$, $r = 2$, and a scale factor of $r = 4$. In the simple case where $r = 1$, the SR resolution problem is merely a denoising one. In other words, the objective is not to increase resolution, and hence there is no blur due to upsampling. In contrast, by increasing the SR factor r more blurring effects occur leading to a higher 3D error in the final reconstructed HR scene Figure 4. In order to evaluate the quality of the filtered depth data and the filtered velocity, we randomly choose one pixel p_t and track its filtered depth value z_t and its filtered velocity $\frac{\Delta z_t}{\Delta t}$ through the super-resolved sequence. We do the same for all SR factors. In Figure 5, we report the tracking results of the

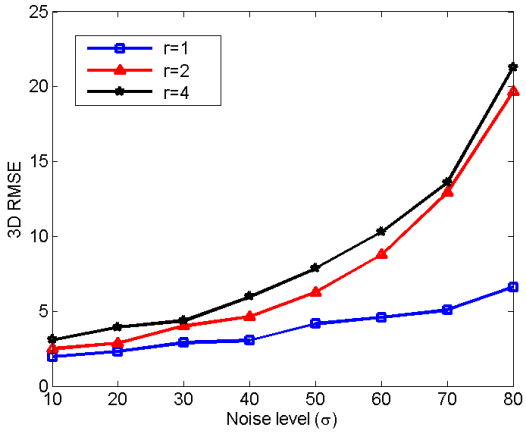

Figure 4. 3D RMSE in mm of the super–resolved hand sequence using the proposed method with different SR scale factors. Increasing the SR factor leads to a higher 3D reconstruction error. This is due to the blurring effects of the upsampling process and the lower resolution of the used LR depth sequence as compared to the one used with $r = 1$.

randomly chosen pixels from the super-resolved sequences with $r = 1$, $r = 2$, and $r = 4$, and a fixed noise level of $\sigma = 50\ mm$. We can see how the depth values are filtered (blue lines) as compared to the noisy depth measurements (red lines) for all scale factors as shown in Figure 5 (a), (b), and (c). Similar behaviour is observed for the corresponding filtered velocities in Figure 5 (d), (e), and (f).

5.2. Publically Available Data

We tested the proposed method using a complex scene with a highly non–rigidly moving object. We use the publicly available "Samba" [1] data. This dataset provides a real sequence of a full 3D dynamic dancing lady scene with high resolution ground truth. This sequence is quite complex where it contains both non–rigid radial motions and self–occlusions, represented by hands and leg movements, respectively. We use the publicly available toolbox V-REP [2] to create from the "Samba" data a synthetic depth sequence with fully known ground truth. We choose to fix a depth camera at a distance of 2 meters from the 3D scene. Its resolution is 1024^2 pixels. The camera is used to capture the depth sequence. Then, similarly to the previous set-up, we downsample the obtained depth sequence with $r = 4$ and further degrade it with additive noise with standard deviation σ varying from 0 to 50 mm. The created LR noisy depth sequence is then super–resolved using state–of–art methods, the conventional bicubic interpolation, *UP-SR* [4], *SISR* [5], and the proposed algorithm. To measure the accuracy of each method, we back project the reconstructed HR depth images to the 3D world using the camera matrix. Then, we calculate the 3D RMSE of each back projected 3D point cloud as compared to the 3D

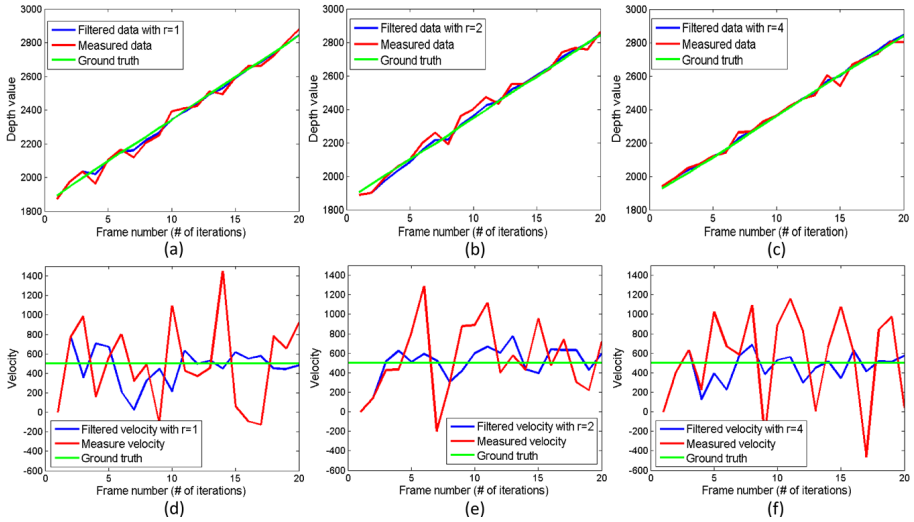

Figure 5. Tracking results for different depth values randomly chosen from the super-resolved sequences with different SR scale factors $r = 1, r = 2$, and $r = 4$, are plotted in (a), (b), and (c), respectively. The corresponding filtered depth displacements are shown in (d), (e), and (f), receptively.

	$\sigma = 25mm$				$\sigma = 50mm$			
	Hand	Torso	Leg	Full body	Hand	Torso	Leg	Full body
Bicubic	10.5	7.5	8.9	8.8	25.2	14.9	13.1	16.5
SISR	**9.0**	5.6	8.4	6.6	14.1	6.9	9.6	9.7
UP-SR	22.2	15.6	9.3	15.9	29.7	17.4	12.8	23.5
Proposed	9.6	**3.6**	**7.5**	**6.3**	**9.9**	**4.8**	**8.1**	**9.5**

Table 1. 3D RMSE in mm for the super–resolved dancing girl sequence using different SR methods. These methods are applied on LR noisy depth sequences with two noise levels. The super–resolution scale factor for this experiment is $r = 4$.

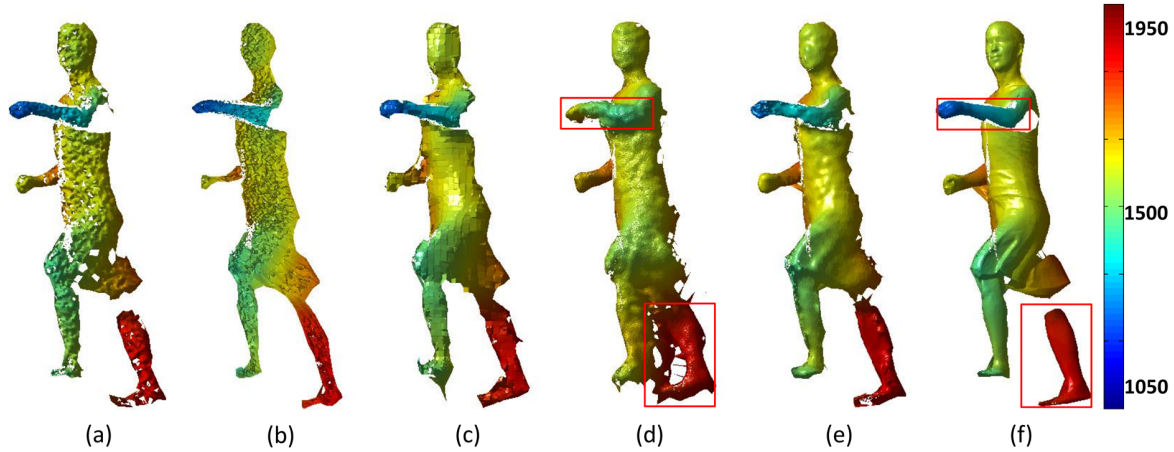

Figure 6. 3D Plotting of one super-resolved depth frame with $r = 4$ using: (b) bicubic interpolation, (c) Patch based single image SR (*SISR*) [5], (d) *UP-SR* [4], (e) our new proposed algorithm. (a) 3D plotting of one LR depth frame. (f) 3D ground truth.

ground truth. Table 1 shows the 3D reconstruction error of the bicubic, *UP-SR* [4], and *SISR* [5] methods as compared to the proposed method versus different noise levels. The comparison is done at two levels: (i) Different parts of the reconstructed 3D body, namely, hand, torso, and the leg, and (ii) full reconstructed 3D body. As expected, by applying the conventional bicubic interpolation method directly on depth images, a large error is obtained. This error is mainly due to the flying pixels around object boundaries. Thus, we run another round of experiments using a modified bicubic

interpolation, where we remove all flying pixels by defining a fixed threshold. Yet, the 3D reconstruction error is still relatively high across all noise levels, see Table 1. This is due to the fact that bicubic interpolation does not profit from the temporal information provided by the sequence. We observe in Table 1 that the proposed method provides, most of the time, better results as compared to state–of–art algorithms. In order to visually evaluate the performance of the proposed algorithm, we plot the super–resolved results of the dancing girl sequence in 3D. We show the results for the sequence at the noise level of $\sigma = 30\ mm$. We note that the proposed algorithm outperforms state–of–art methods by keeping the fine details (e.g. the details of the face) as can be seen in Figure 6 (e). Note that the *UP-SR* algorithm fails in the presence of radial movements and self–occlusions, see red boxes in Figure 6 (d). In contrast, the *SISR* algorithm can handle these cases, but cannot keep the fine details due to its patch–based nature, see Figure 6 (c). In addition, a heavy training phase is required.

5.3. Real Data

Finally, we tested the proposed algorithm on a real sequence captured with a ToF camera (pmd CamBoard Nano). The captured LR depth sequence contains a non rigidly moving face. Samples of the LR captured frames are plotted in the first and second rows of Figure 7. We super–resolve this sequence using the proposed algorithm with an SR scale factor of $r = 4$. Obtained results are given in 3D in the third and fourth rows of Figure 7. The obtained results show the effectiveness of the proposed algorithm in reducing the noise, and further increasing the resolution of the reconstructed 3D face under large non–rigid deformations. To visually appreciate these results as compared to state–of–art methods, we tested the bicubic, *UP-SR*, and *SISR* on the same LR real depth sequence. Obtained results show the superiority of the proposed algorithm as compared to other methods, see Figure 1. In Figure 8, we plot the filtered depth value of a randomly chosen tracked pixel. The blue line shows the filtered trajectory of this pixel as compared to its row noisy measurement in red. The algorithm's run–time on this sequence is 50 ms per frame on a 2.2 GHz i7 processor with 4 Gigabyte ram.

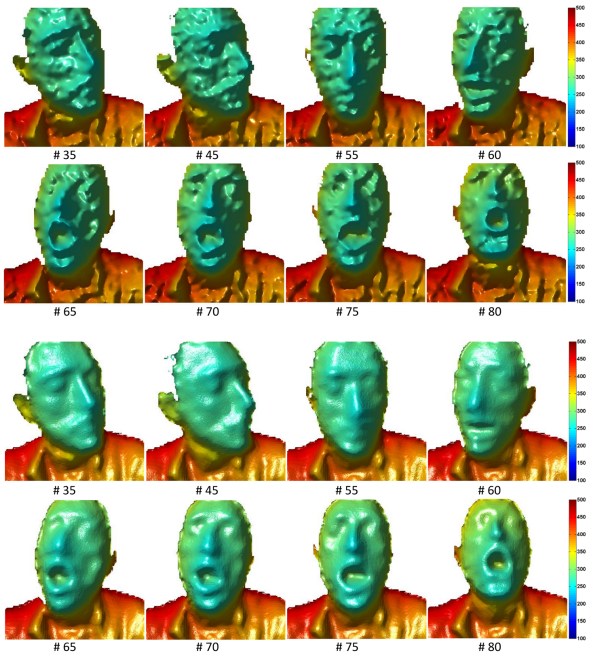

Figure 7. Results of applying the proposed algorithm on a real sequence captured by a LR ToF camera (120×160 pixels) of a non–rigidly moving face. First and second rows contain a 3D plotting of selected LR captured frames. Third and fourth rows contain the 3D plotting of the super-resolved depth frames with $r = 4$.

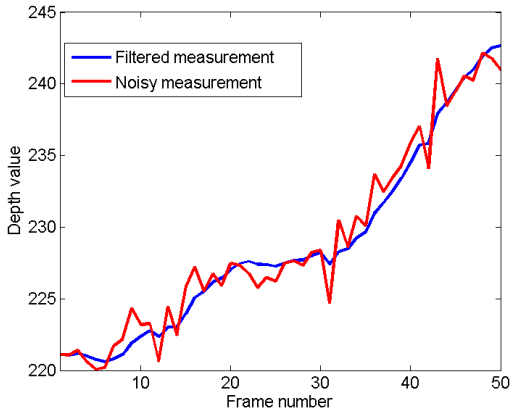

Figure 8. Filtered depth value profile of a tracked pixel through the super-resolved sequence of a real face, with SR scale factor of 4.

6. Conclusion

A new real–time dynamic multi–frame super-resolution algorithm for depth videos has been proposed. It has been shown to be effective in enhancing the resolution and the quality of low resolution dynamic scenes with highly non–rigidly moving objects. Obtained results show the robustness of the proposed algorithm against radial motions. This is handled by first estimating the depth displacement, and simultaneously correcting the depth measurement by Kalman filtering. For the sake of real–time processing, the proposed algorithm is based on per–pixel temporal processing of the depth video sequence such that multiple one–dimensional signals are filtered separately. Each filtered depth frame is further refined using a multi–level iterative bilateral total variation regularization after filtering and before proceeding to the next frame in the sequence. In the case of self-occlusions, the proposed algorithm needs a few number of depth measurements before converging, which is not suitable for some applications. Our future work will focus on increasing robustness to self–occlusions.

References

[1] http://people.csail.mit.edu/drdaniel/mesh_animation/. 6

[2] http://www.k-team.com/mobile-robotics-products/v-rep. 6

[3] K. Al Ismaeil, D. Aouada, B. Mirbach, and B. Ottersten. Bilateral filter evaluation based on exponential kernels. In *Pattern Recognition (ICPR), 2012 20th IEEE International Conference on*, pages 258–261, Nov 2012. 5

[4] K. Al Ismaeil, D. Aouada, B. Mirbach, and B. Ottersten. Dynamic super resolution of depth sequences with non-rigid motions. In *Image Processing (ICIP), 2013 20th IEEE International Conference on*, pages 660–664, Sept 2013. 1, 2, 3, 6, 7

[5] O. M. Aodha, N. Campbell, A. Nair, and G. Brostow. Patch based synthesis for single depth image super-resolution, 2012. 1, 6, 7

[6] M. Charest, M. Elad, and P. Milanfar. A general iterative regularization framework for image denoising. In *Information Sciences and Systems, 2006 40th Annual Conference on*, pages 452–457, March 2006. 5

[7] Y. Cui, S. Schuon, S. Thrun, D. Stricker, and C. Theobalt. Algorithms for 3d shape scanning with a depth camera. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 35(5):1039–1050, May 2013. 1

[8] M. Elad and A. Feuer. Super-resolution reconstruction of image sequences. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 21(9):817–834, Sep 1999. 2, 3, 5

[9] M. Elad and A. Feuer. Superresolution restoration of an image sequence: adaptive filtering approach. *Image Processing, IEEE Transactions on*, 8(3):387–395, Mar 1999. 2

[10] S. Farsiu, M. Elad, and P. Milanfar. Video-to-video dynamic super-resolution for grayscale and color sequences. *EURASIP J. Appl. Signal Process.*, 2006:232–232, Jan. 2006. 2

[11] S. Farsiu, M. Robinson, M. Elad, and P. Milanfar. Fast and robust multiframe super resolution. *Image Processing, IEEE Transactions on*, 13(10):1327–1344, Oct 2004. 2, 5

[12] J. Li, Z. Lu, G. Zeng, R. Gan, and H. Zha. Similarity-aware patchwork assembly for depth image super-resolution. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 3374–3381, June 2014. 1

[13] C. B. Newland, D. A. Gray, and D. Gibbins. Modified kalman filtering for image super-resolution: Experimental convergence results. In *Proceedings of the Ninth IASTED International Conference on Signal and Image Processing*, SIP '07, pages 58–63, Anaheim, CA, USA, 2007. ACTA Press. 2

[14] S. Osher, M. Burger, D. Goldfarb, J. Xu, and W. Yin. An iterative regularization method for total variation-based image restoration. *Simul*, 4:460–489, 2005. 5

[15] V. Patanaviji, S. Tae-O-Sot, and S. Jitapunkul. A robust iterative super-resolution reconstruction of image sequences using a lorentzian bayesian approach with fast affine block-based registration. In *Image Processing, 2007. ICIP 2007. IEEE International Conference on*, volume 5, pages V – 393–V – 396, Sept 2007. 2

[16] S. Schuon, C. Theobalt, J. Davis, and S. Thrun. Lidarboost: Depth superresolution for tof 3d shape scanning. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 343–350, June 2009. 1, 2

[17] M. Sturmer, J. Penne, and J. Hornegger. Standardization of intensity-values acquired by time-of-flight-cameras. In *Computer Vision and Pattern Recognition, 2008. CVPRW 2008. IEEE Workshop on*, pages 660–664, Sept 2013. 4

[18] J. Tian and K.-K. Ma. A new state-space approach for super-resolution image sequence reconstruction. In *Image Processing, 2005. ICIP 2005. IEEE International Conference on*, volume 1, pages I–881–4, Sept 2005. 2

[19] Q. L. Q. S. S. X. Wenshu Li1, Chao Zhao1. A parameter-adaptive iterative regularization model for image denoising, 2012. 5

[20] M. Yamamoto, P. Boulanger, J.-A. Beraldin, and M. Rioux. Direct estimation of range flow on deformable shape from a video rate range camera. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 15(1):82–89, Jan 1993. 3, 4