DEPTH ESTIMATION FROM A VIDEO SEQUENCE WITH MOVING AND DEFORMABLE OBJECTS

Manuel Martinello, Paolo Favaro

School of EPS, Heriot-Watt University, Edinburgh, UK

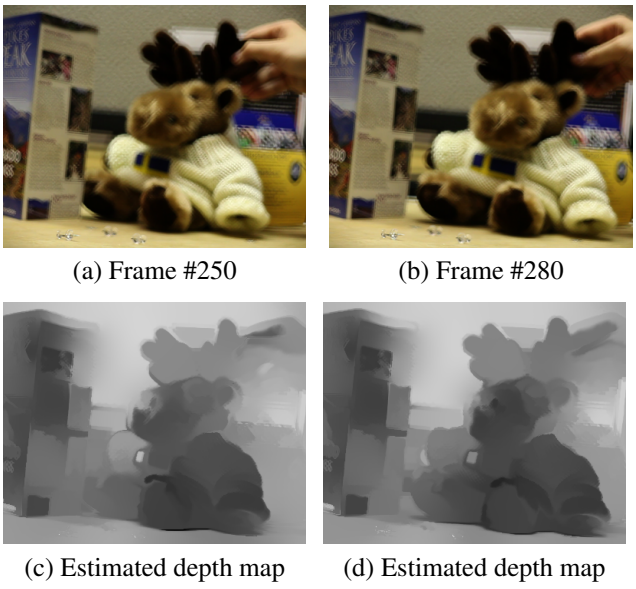
Keywords: coded aperture, depth from video, regularization, non-local means filtering.

Abstract

In this paper we present an algorithm for depth estimation from a monocular video sequence containing moving and deformable objects. The method is based on a coded aperture system (*i.e.,* a conventional camera with a mask placed on the main lens) and it takes a *coded* video as input to provide a sequence of dense depth maps as output. To deal with nonrigid deformations, our work builds on the state-of-the-art single-image depth estimation algorithm. Since single-image depth estimation is very ill-posed, we cast the reconstruction task as a regularized algorithm based on nonlocal-means filtering applied to both the spatial and temporal domain. Our assumption is that regions with similar texture in the same frame and in neighbouring frames are likely to belong to the same surface. Moreover, we show how to increase the computational efficiency of the method. The proposed algorithm has been successfully tested on challenging real scenarios.

1 Introduction

Estimating the three-dimensional (3D) location of objects in the scene is a crucial step for performing tasks such as human-machine interaction, tracking, and autonomous navigation. While 3D structure can be recovered by using multiple cameras or depth sensors, we investigate the use of a single camera, which can reduce the cost of the system, and be combined with other sensors to further improve the overall accuracy of depth estimation. Typical approaches based on single cameras (e.g, optical flow), can be used to estimate depth in the presence of rigid motion, but not with general motion due to deformable or articulated objects. Since no information about the objects and their motion is given, we cannot rely on matching multiple frames; instead, we use the information that is present in each single frame. Furthermore, to make the depth estimates consistent in time and space, we cast the depth estimation problem as a regularized optimization task. In summary, this work provides the following three main contributions: 1) It presents, to the best of our knowledge, the first single-frame video depth estimation algorithm, capable of handling moving and deformable objects; 2) It introduces a novel spatial and temporal depth smoothness constraint, based on nonlocal-means (NLM) filtering: Pixels whose intensities match within a certain spatial and temporal range are likely to share similar depths; 3) The proposed algorithm is robust and accurate on real videos (see Fig. 1).



(a) Frame #250 (b) Frame #280

(c) Estimated depth map (d) Estimated depth map

Figure 1: **Depth estimation with deformable objects**. a) and b) Two input frames in the video sequence; c) and d) Relative depth maps obtained from our method.

2 Related Work

When a scene undergoes rigid motion, depth estimation from a single video can be carried out in several ways. The two most common techniques are optical flow and structure from motion. The former technique consists of finding correspondences between neighbouring frames and measuring the difference of their position: The shift is related to the depth of the scene only when the camera is moving and the scene is rigid [11, 17]. Models for non-rigid structures have been proposed in structure from motion [20, 21, 27], but they assume that feature correspondences are known [27] or occluders are treated as outliers [20, 21, 25] and therefore not reconstructed. Instead, the approach presented in this paper estimates the depth of the whole scene. High-quality depth maps have been obtained in [26] from a video sequence captured with a freely moving camera. However, the method fails when moving or deformable objects are present in most of the area of the scene.

Our algorithm does not rely on matching multiple frames. Since depth information is extracted at each frame, it relates to work on single-image depth estimation. One of the most successful technology in depth estimation is the use

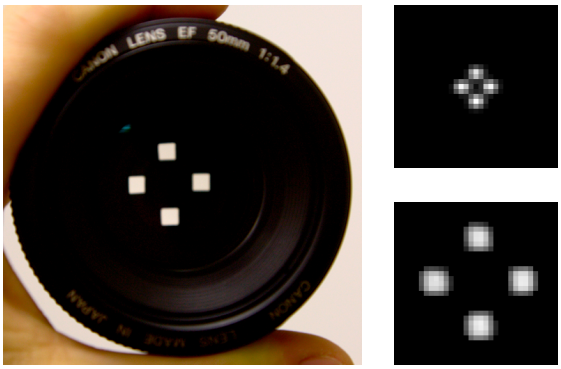
of active illumination: By projecting some structured light into the scene and then measuring the blur of the light pattern [8] or light dots [14] in the captured image has led to good results in depth estimation. Particularly relevant is the recent introduction of Kinect [23], a depth camera based on structured light in the infrared (IR) range. Among passive methods on single image depth estimation, the main contributions are perhaps from [10] and [22], who show that the introduction of a mask in the camera lens can improve the blur identification (and therefore the depth reconstruction). Both works propose a method for estimating both depth and all-in-focus texture from a single coded image. The main drawback is that the methods are based on deconvolution and can deal only with small amounts of blur. More recently, [12] and [13] has shown that depth can actually be recovered from a single image without estimating the radiance. Our algorithm is an extension of the latter work to video sequences. Moreover, we show that we can rewrite the energy minimization in a more efficient way, so that the method can be used to process videos quickly. In our experiments, we use the aperture in Fig. 2(a), since it gives the best performance on depth estimation among all the binary aperture masks proposed in the literature [13].
To regularize our estimation, the concept of non-local mean filters is applied to depth reconstruction: The main idea is to link the depth values of pixels sharing the same colour (or texture). The concept of correlating pixels with similar colour or texture has been shown to be particularly effective in preserving edges in stereopsis [5, 16, 18] and thin structure in depth estimation [6, 17], as well as in image denoising [2, 15, 19].

3 Depth Estimation from Monocular Video

When a part of the scene is brought into focus, objects placed at a different location appear out-of-focus; the amount of defocus depends on their location in the scene: More precisely, it depends on the distance between the objects and the focal plane. Because of this relationship, if we can identify the blur kernel for each object point in the scene, we can reconstruct the relative depth of the items in the scene. The exact distance from the camera can also be recovered from the blur size with a calibration procedure, once the camera setting is known.
In this section, we present the depth estimation algorithm, which takes as input a video sequence captured by a single coded aperture camera. As described in Section 1, we consider videos with moving and deformable objects: Therefore we cannot rely on matching multiple frames.

3.1 Image Formation Model

When we capture a video with a coded aperture camera, we have a set of T coded frames $\boldsymbol{g}_1, \boldsymbol{g}_2, \ldots, \boldsymbol{g}_T$. For each of these frames, the 3D scene, captured at a particular time t, can be decomposed in two entities: a 2D *sharp* frame $\boldsymbol{f}_t$, whose texture is all-in-focus, and a depth map $\boldsymbol{d}_t$, which assigns a depth value (distance from the camera) to each pixel in $\boldsymbol{f}_t$. Our aim is to recover the geometry $\boldsymbol{d}_t$ of the scene



(a) (b)

Figure 2: **Binary aperture mask and corresponding point spread functions (PSF)**. (a) A coded aperture camera is obtained by placing an mask into the camera lens; (b) Examples of point spread functions obtained with the mask in (a) at different depths.

at each time instant t. As previously described, different depths correspond to different blur sizes in the coded image $\boldsymbol{g}_t$. Hence, the blur kernel $\boldsymbol{h}_p$, also called point spread function (PSF), must be allowed to vary at each pixel. Two examples of PSFs of our coded aperture systems are shown in Fig. 2(b). If we consider all the elements ordered as column vectors, we can write $\boldsymbol{g}_t$ as a product of matrices

$$\boldsymbol{g}_t = \underbrace{\begin{bmatrix}\boldsymbol{h}_1 & \boldsymbol{h}_2 & \ldots & \boldsymbol{h}_N\end{bmatrix}}_{\boldsymbol{H}_{d_t}} \cdot \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_N \end{bmatrix} \Bigg\} \boldsymbol{f}_t, \qquad (1)$$

where N is the number of pixels of each frame and $\boldsymbol{H}_{\boldsymbol{d}_t}$ is a symmetric and sparse matrix that contains the information about the depth of the scene.
Since the scene is non-rigid (and hence we cannot rely on matching multiple frames), and since the sharp frames $\boldsymbol{f}_t$ are unknown, we should, in principle, simultaneously estimate both depth and all-in-focus image from $\boldsymbol{g}_t$. However, it has been shown in [7] that this problem can be divided and solved in two separate steps: 1) depth estimation only and 2) image deblurring by using the estimated depth. In this paper, we focus our work on the former step.
We formulate the problem of depth estimation as a minimization of the cost functional

$$\hat{\boldsymbol{d}} = \underset{\boldsymbol{d}}{\operatorname{argmin}} \; E_{data}[d] + \alpha_1 E_{tv}[d] + \alpha_2 E_{nlm}[d] \,, \qquad (2)$$

where α_1 and α_2 are two positive constants. In our approach, the data fidelity term $E_{data}[d]$ is based on depth from a single image (see Section 3.2) and we concentrate more on designing the regularization terms (Section 3.3).

3.2 The Data Fidelity Term: Depth from a Single Frame

The first term is based on the state-of-the-art depth from single coded image algorithm [13]. The method identifies

the blur size (and therefore the depth) at each pixel of a coded image by using projections onto subspaces. In our case, the depth $\boldsymbol{d}_t$ can be extracted from the single frame $\boldsymbol{g}_t$ without deblurring the image $\boldsymbol{f}_t$, by minimizing

$$E_{data}[d] = \sum_{\mathbf{p}} ||\boldsymbol{H}^{\perp}_{d_t(\mathbf{p})} \tilde{\boldsymbol{g}}^{\mathbf{p}}_t||^2_2 \tag{3}$$

where $\tilde{\boldsymbol{g}}^{\mathbf{p}}_t$ indicates the patch of size $\delta \times \delta$ centred at the pixel $\mathbf{p}$ at time t, that has been rearranged as a column vector. The symbol δ denotes the size of the maximum level of defocus considered. The matrix $\boldsymbol{H}^{\perp}_{\boldsymbol{d}_t}$ is built via a learning procedure, described in details in [13], for each depth level d such that

$$\begin{cases} \boldsymbol{H}^{\perp}_{d_i} \boldsymbol{H}_{d_j} \approx 0, & \text{if } d_i = d_j \\ \boldsymbol{H}^{\perp}_{d_i} \boldsymbol{H}_{d_j} \gg 0, & \text{if } d_i \neq d_j \end{cases} \tag{4}$$

for any possible sharp texture $\boldsymbol{f}_t$.
A remarkable fact is that, for the purpose of depth estimation alone, there is no need to know the shape of the mask: In fact, the learning is performed on real coded images of a planar plane (with texture), placed at different distances from the camera.
Since we are processing videos, in Section 4.1 we work out possible solutions to approximate equation (3) in order to increase the efficiency of this algorithm and make it suitable for parallel computation.

3.3 Total Variation and Non-Local Means Filtering

The first regularization term $E_{tv}[d]$ in Equation (2) represents the total variation

$$E_{tv}[d] = \int \|\nabla d(\mathbf{p})\| \, d\mathbf{p} \, , \tag{5}$$

which constrains the solutions to be piecewise constant [4]. However, this term alone tends to misplace the edge location and to remove thin surfaces, since it can combine together pixels that do not belong to the same surface.
To contrast this behaviour, we design a term that links depth values of pixels sharing the same colour (or texture) $E_{nlm}[d]$. Corresponding pixels can belong either to the same frame (Section 3.3.1) or to different frames (Section 3.3.2).

3.3.1 Spatial Smoothness

In this section we briefly analyze how neighbourhood filtering methods establish correspondences between pixels and then extend the concept to a video sequence.
Many depth estimation methods assume that pixels with the same color or texture are likely to share also the same depth value. This can be obtained with a non-local *sigma*-filter [9], based on intensity differences

$$W_1(\mathbf{p}, \mathbf{q}) = e^{-\frac{|g(\mathbf{p}) - g(\mathbf{q})|^2}{\tau_1}} \, , \tag{6}$$

where the weight assigned to $W_1(\mathbf{p}, \mathbf{q})$ represents how strong the link between $\mathbf{p}$ and $\mathbf{q}$ is, or, in other words, how likely they are to be located at the same depth. The symbol τ_1 indicates the bandwidth parameter determining the size of the filter. Loosely speaking, pixels with values much closer to each other than τ_1 are linked together, while the ones with values much larger than τ_1 are not.
This type of filter has been largely used for image denoising, although it generates artifacts at edges and uniform regions [3], probably due to the pixel-based matching being sensitive to noise: To reduce this effect, one could use region-based matching as in the *non-local means* filter [6]:

$$W_1(\mathbf{p}, \mathbf{q}) = e^{-\frac{G_\sigma * |g(\mathbf{p}) - g(\mathbf{q})|^2(0)}{\tau_1}} \tag{7}$$

where G is an isotropic Gaussian kernel with variance σ such that

$$G_\sigma * |g(\mathbf{p}) - g(\mathbf{q})|^2(0) = \int_{\mathbb{R}^2} G_\sigma(\mathbf{x}) |g(\mathbf{p}+\mathbf{x}) - g(\mathbf{q}+\mathbf{x})|^2 |d\mathbf{x}. \tag{8}$$

Now we have obtained a neighbourhood filter for combining pixels in the same frame. However, since we have multiple frames, we can extend the correspondences temporally.

3.3.2 Temporal Smoothness

If objects do not move much between neighbouring frames, we can easily find correspondences (despite deformations of the objects).
Let us consider a pixel $\mathbf{p}$ from a frame $\boldsymbol{g}_{t_0}$ (captured at time t_0). We can rewrite the filter in equation (7) in a more general form where the pixel $\mathbf{q}$ is now free to belong to any frame $\boldsymbol{g}_t$ of the video sequence

$$W_1(\mathbf{p}, t_0, \mathbf{q}, t) = e^{-\frac{G_\sigma * |g_{t_0}(\mathbf{p}) - g_t(\mathbf{q})|^2(0)}{\tau_1}} \, , \tag{9}$$

which included the case when $t = t_0$. Indeed, when considering the frame $\boldsymbol{g}_{t_0}$, the probability to find the same objects (or part of them) in another frame $\boldsymbol{g}_t$ decays moving away from the time t_0. Hence, we can add a filter that implements this likelihood:

$$W_2(t_0, t) = e^{-\frac{|t - t_0|}{\tau_2}} \tag{10}$$

where τ_2 is the bandwidth parameter in the temporal domain. This parameter is very important in deciding the number of frames to consider in the regularization.

We can now combine the spatial filter (equation (7)) and the temporal filter (equation (10)) together to obtain the final filtering weights

$$W(\mathbf{p}, t_0, \mathbf{q}, t) = e^{-\frac{|t - t_0|}{\tau_2}} \; e^{-\frac{G_\sigma * |g_{t_0}(\mathbf{p}) - g_t(\mathbf{q})|^2(0)}{\tau_1}} . \tag{11}$$

Notice that when the temporal term uses only 2 frames, t_0 and t_1, the corresponding pixels given by $W(\mathbf{p}, \mathbf{q}, t_0, t_1)$ include the matchings obtained from optical flow.
Finally, we use the sparse matrix $W(\mathbf{p}, t_0, \mathbf{q}, t)$ to define our neighbourhood regularization term, so that pixels with

similar colors are encouraged to have similar depths value, *i.e.,*

$$E_{nlm}[\boldsymbol{d}] = \int\int W(\mathbf{p}, t_0, \mathbf{q}, t)\ (d_t(\mathbf{q}) - d_{t_0}(\mathbf{p}))^2\ d\mathbf{q}\ dt. \tag{12}$$

where $\mathbf{p}$ and $\mathbf{q}$ represent any pixel in the video sequence. The term E_{nlm} is quadratic in the unknown depth map $\boldsymbol{d}$ and therefore it can be easily minimized.

4 Implementation Details

In this section we first study the data fidelity term equation (2) and find a sound approximation to improve the efficiency of the proposed method (Section 4.1). Secondly, we describe the iterative approach we adopt to minimize the cost functional in equation (2) (Section 4.2).

4.1 Filters Decomposition for Parallel Computation

We focus now on the computation of the data term $E_{data}[d]$. This term can quickly generate a non-regularized depth mapth (also called *raw* depth map), when $\alpha_1 = \alpha_2 = 0$ in Equation (2)). In this section, the subscripts $(_t)$ should be used, but are omitted for simplicity; the patches $\tilde{\boldsymbol{g}}_t^{\mathbf{p}}$ will then be denoted by $\tilde{\boldsymbol{g}}_{\mathbf{p}}$.

Since $\boldsymbol{H}_{\boldsymbol{d}}^{\perp}$ is a projection, we can rewrite equation (3) as

$$E_{data}[d] = \sum_{\mathbf{p}} \tilde{\boldsymbol{g}}_{\mathbf{p}}^T\ \boldsymbol{H}_{d(\mathbf{p})}^{\perp}\ \tilde{\boldsymbol{g}}_{\mathbf{p}}\ . \tag{13}$$

The computation of this term is suitable for parallel computation, since we can obtain a depth value at each pixel $\mathbf{p}$, independently from the other pixels. Also, we have that $\boldsymbol{H}_{\boldsymbol{d}}^{\perp} = \boldsymbol{U_d}\ \boldsymbol{U_d}^T$, where $\boldsymbol{U_d}$ is a matrix with orthonormal column vectors by construction [7]. Then, equation (13) can be computed and represented (in memory) more efficiently as:

$$E_{data}[d(\mathbf{p})] = \|\tilde{\boldsymbol{g}}_{\mathbf{p}}^T\ \boldsymbol{U}_{d(\mathbf{p})}\|^2. \tag{14}$$

When using equation (14) as fidelity term, a raw depth map of size 500×600 pixels can be obtained in about 200 s.

We look now into the set of matrices $\boldsymbol{U_d}$ to check if there are possible approximations that can be adopted. The matrix $\boldsymbol{U_d} = [\boldsymbol{u}_{1,d}\ \boldsymbol{u}_{2,d}\ \ldots\ \boldsymbol{u}_{M,d}]$ has size $\delta^2 \times M$, and its columns are *orthonormal* filters [7]. Therefore, equation (14) can be thought as a series of 2D convolutions between the whole image $\boldsymbol{g}$ and each column of $\boldsymbol{U_d}$ (both reshaped to 2D). This is done for each depth level d: we can then say that, to estimate the depth map for each frame of the video sequence, we have to compute $M \times N_d$ 2D-convolutions, where N_d is the number of depth levels being considered. Just to have an idea of the dimensions we are dealing with, in our experiments we have $M \simeq 150$ and $N_d = 30$.

Since the total numbers of filters we use for each mask is much bigger than the size of each filter itself ($\delta \times \delta$, with $\delta = 33$), we can express each orthonormal filter $\boldsymbol{u}_{k,d}$ as a linear combination of a common basis $\boldsymbol{B}$:

$$\boldsymbol{u}_{k,d} = \underbrace{\begin{bmatrix}\boldsymbol{b}_1 & \boldsymbol{b}_2 & \ldots & \boldsymbol{b}_L\end{bmatrix}}_{\boldsymbol{B}} \cdot \boldsymbol{a}_{k,d}, \tag{15}$$

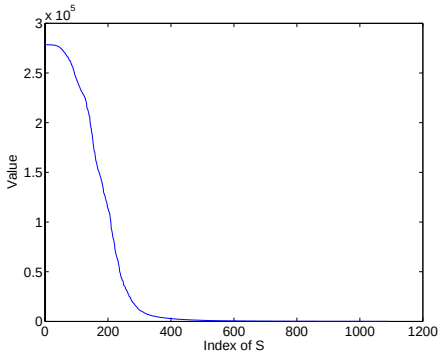


Figure 3: **Eigenvalues of S.** The graph shows the values along the diagonal of $\boldsymbol{S}$; such values correspond to the eigenvalues of the matrix $\tilde{\boldsymbol{U}}$.

where $\boldsymbol{a}_{k,d}$ is a column vector containing the coefficients for the k-th filter at the depth d. By substituting equation (15) in equation (14), we can rewrite the fidelity term as

$$E_{data}[d(\mathbf{p})] = \left\|\tilde{\boldsymbol{g}}_{\mathbf{p}}^T\ \boldsymbol{B}\ \boldsymbol{A}_{d(\mathbf{p})}\right\|^2. \tag{16}$$

with $\boldsymbol{A}_{d(\mathbf{p})} = [\boldsymbol{a}_{1,d}\ \boldsymbol{a}_{2,d}\ \ldots\ \boldsymbol{a}_{M,d}]$.

Notice that with this formulation we have reduced the number of 2D convolutions to the number of columns of $\boldsymbol{B}$; in other words, the complexity corresponds to the number of vectors that compose the common basis (in our experiments there are about 200 vectors). The depth map at each frame (500×600 pixels) can now be estimated in about 4 seconds.

In the following two sections we illustrate how to estimate the common basis $\boldsymbol{B}$ and the matrix of coefficients $\boldsymbol{A}$. These steps have to be run just once, right after the learning of $\boldsymbol{H}_{\boldsymbol{d}}^{\perp}$ for a given mask.

4.1.1 Estimating the Common Basis B

We build $\tilde{\boldsymbol{U}}$ (of size $\delta^2 \times M \times N_d$) by joining in the third dimensions the matrices $\boldsymbol{U}_d$ for all possible depth levels, $1 < d < N_d$. We then perform the singular value decomposition (SVD) of $\tilde{\boldsymbol{U}} = \boldsymbol{W}\boldsymbol{S}\boldsymbol{V}^T$: the most important *orthogonal* vectors that are in the left part of the matrix $\boldsymbol{W}$.

The diagonal of $\boldsymbol{S}$ contains the eigenvalues, *i.e.,* the values that indicate the importance of each column of $\boldsymbol{W}$ to generate the space $\tilde{\boldsymbol{U}}$. The values along the diagonal are plotted in Fig. 3.

The basis $\boldsymbol{B}$ is then composed by the most important column of $\boldsymbol{W}$; experimentally, we have seen that the first 200 vectors are a good approximation for generating the space of $\tilde{\boldsymbol{U}}$.

4.1.2 Estimating the Coefficients $a_{k,d}$

Now that we have the common basis $\boldsymbol{B}$, for each filter $\boldsymbol{u}_{k,d}$ we have to estimate the coefficients $\boldsymbol{a}_{k,d}$, such that eq. (15) is satisfied. This can be done via:

$$\boldsymbol{a}_{k,d}^T = \boldsymbol{u}_{k,d}^T \boldsymbol{B}^T (\boldsymbol{B}\boldsymbol{B}^T). \tag{17}$$

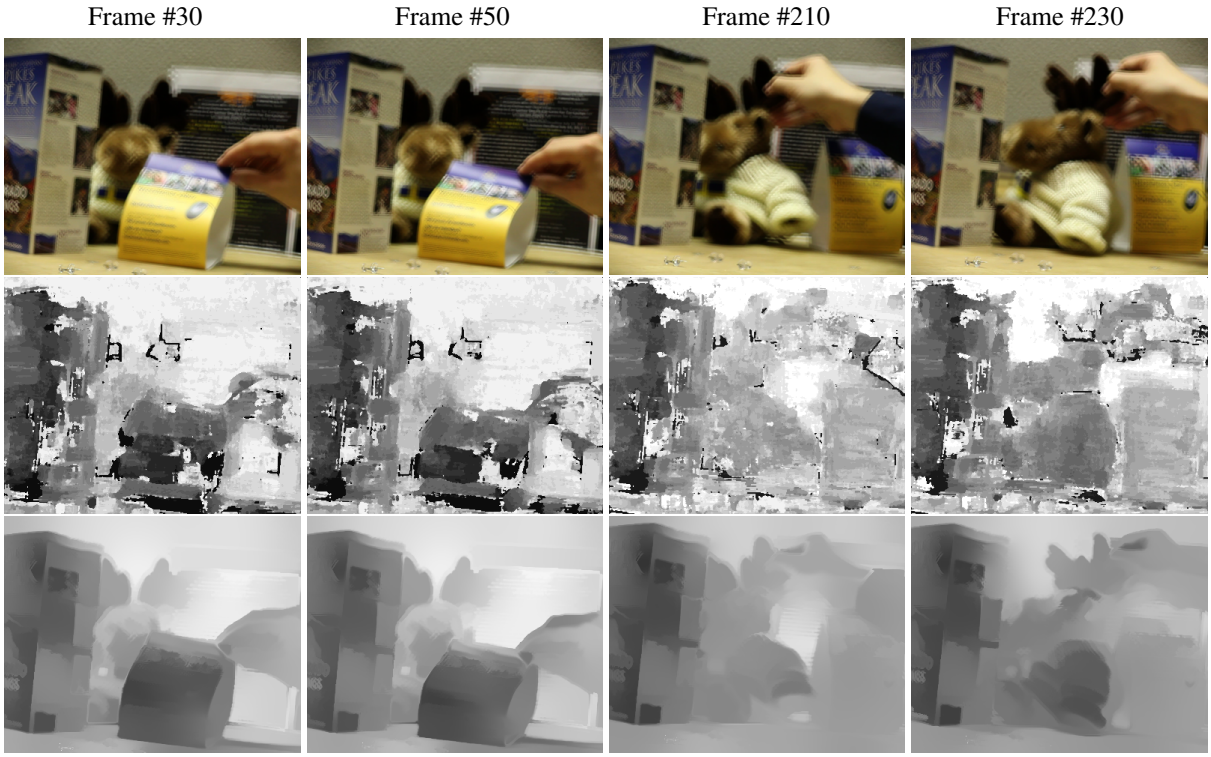


Figure 4: **Table dataset**. **Top row:** Some of the frames of the coded input video; **Central row:** Raw depth maps estimated only with the data fidelity term, and without any regularization ($\alpha_1 = \alpha_2 = 0$) ; **Bottom row:** Final depth maps obtained with our method.

4.2 Iterative Linearization Approach

We solve the Euler-Lagrange equations of the cost functional in equation (2)

$$\nabla E[d] \doteq \nabla E_{data}[d] + \alpha_1 \nabla E_{tv}[d] + \alpha_2 \nabla E_{sm}[d] \quad (18)$$

via iterative linearization [1]. The second and third terms are can be computed easily as

$$\nabla E_{tv}[d] = -\nabla \cdot \left(\frac{\nabla d(\mathbf{p})}{|\nabla d(\mathbf{p})|} \right) \quad (19)$$

and

$$\nabla E_{nlm}[d] = \int \int W(\mathbf{p}, \mathbf{q}, t_0, t) \, (d(\mathbf{p}) - d(\mathbf{q})) \, d\mathbf{q} \, dt \quad (20)$$

while the data fidelity term requires a further analysis. In fact, the energy $E_{data}[d]$ is non convex. Therefore, we expand our energy in Taylor series (stopping at the third term)

$$E_{data}[d] = E_{data}[\boldsymbol{d}_0] + \nabla E_{data}[\boldsymbol{d}_0](d - \boldsymbol{d}_0) \quad (21)$$

$$+ \frac{1}{2}(d - \boldsymbol{d}_0)^T \, \mathbf{H} E_{data}[\boldsymbol{d}_0](d - \boldsymbol{d}_0) \, , \quad (22)$$

where $\mathbf{H}$ indicates the *Hessian*. Now we can compute its derivative with respect to d

$$\nabla E_{data}[d] = \nabla E_{data}[\boldsymbol{d}_0] + \mathbf{H} E_{data}[\boldsymbol{d}_0](d - \boldsymbol{d}_0) \, , \quad (23)$$

where $\boldsymbol{d}_0$ represents the initial depth estimate obtained when setting $\alpha_1 = \alpha_2 = 0$.
Since the conditions for convergence require $\mathbf{H} E_{data}[\boldsymbol{d}_0]$ to be positive-definite, we use $|\mathbf{H} E_{data}[\boldsymbol{d}_0]|$ and make it strictly diagonally dominant [24].

5 Experiments on Real Data

We have captured some videos using our coded aperture camera, a Canon EOS-5D Mark-II, where a mask has been inserted into a 50mm f/1.4 lens (as displayed in Fig. 2(a)). The two datasets shown in this paper, Fig. 4 and Fig. 5, are very challenging scenario for depth estimation using a single camera. For both datasets, we show some coded frames from the video sequence and their corresponding estimated depth maps. Below each input frame there are two depth maps: 1) the raw depth map (central row), obtained by minimizing only the term E_{data} and 2) the final depth map (bottom row) resulting from minimizing the cost in equation (2).
Both videos have been taken when the camera was handheld, and therefore the camera is also moving. The depth estimation, however, it is not affected by this shake. The video shown in Fig. 5 has been captured indoor in a very low light condition; therefore the input video is very noisy (ISO 2000). Nevertheless, the method still outputs impressive results, proving its robustness and consistency. Moreover, the quality of the results in this dataset may suggest that they can be used for tasks such as body pose estimation, or body part recognition.

6 Conclusion

We have presented for the first time a method to estimate depth from a single video with moving and deformable objects. The approach is based on coded aperture technology, where a mask is placed on the lens of a conventional camera. Firstly, we analyze and improve the efficiency of state-

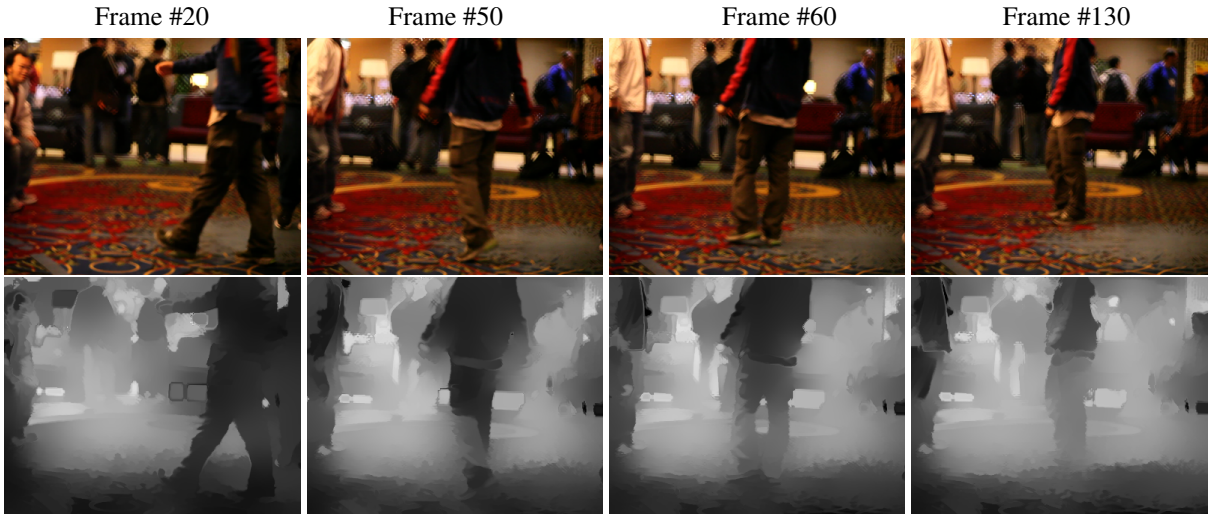


Figure 5: **People Dataset**. **Top row:** Some examples of frames from the coded input video; **Bottom row:** Depth maps reconstructed with our method.

of-the-art depth estimation from a single coded image. Secondly, we introduce a regularization term, based non-local means filtering, that creates at the same time a spatial and temporal neighbourhood of pixels that are likely to share the same depth value. The method is then tested on real data and high-quality depth maps are obtained from very challenging scenes.

References

[1] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert. High accuracy optical flow estimation based on a theory for warping. *ECCV*, 4:25–36, May 2004.

[2] A. Buades, B. Coll, and J.-M. Morel. A non-local algorithm for image denoising. *CVPR*, 2005.

[3] A. Buades, B. Coll, and J.-M. Morel. Nonlocal image and movie denoising. *IJCV*, 76(2):123–139, 2007.

[4] T. F. Chan and J. Shen. *Image processing and analysis: variational, PDE, wavelet, and stochastic methods*. Society for Industrial and Applied Mathematics, 2005.

[5] D.Nister, H. Stewenius, R. Yang, L. Wang, and Q. Yang. Stereo matching with color-weighted correlation, hierachical belief propagation and occlusion handling. *CVPR*, 2:2347–2354, 2006.

[6] P. Favaro. Recovering thin structures via nonlocal-means regularization with application to depth from defocus. *CVPR*, pages 1133 – 1140, Jun 2010.

[7] P. Favaro and S. Soatto. A geometric approach to shape from defocus. *TPAMI*, 27(3):406–417, Mar 2005.

[8] B. Girod and E. H. Adelson. System for ascertaining direction of blur in a range-from-defocus camera. *US Patent No. 4,939,515*, 1990.

[9] J. Lee. Digital image smoothing and the sigma filter. *Computer Vision, Graphics and Image Processing*, 24(2):255–269, Nov 1983.

[10] A. Levin, R. Fergus, F. Durand, and W. T. Freeman. Image and depth from a conventional camera with a coded aperture. *ACM Trans. Graph.*, 26(3):70, Aug 2007.

[11] C. Liu. Beyond pixels: Exploring new representations and applications for motion analysis. *Doctoral Thesis, Massachusetts Institute of Technology*, May 2009.

[12] M. Martinello, T. E. Bishop, and P. Favaro. A bayesian approach to shape from coded aperture. *ICIP*, Sep 2010.

[13] M. Martinello and P. Favaro. Single image blind deconvolution with higher-order texture statistics. *Video Processing and Computational Video*, LNCS7082, 2011.

[14] F. Moreno-Noguer, P. N. Belhumeur, and S. K. Nayar. Active refocusing of images and videos. *ACM Trans. Graph.*, Aug 2007.

[15] J. Salmon and Y. Strozecki. From patches to pixel in non-local methods: weighted-avarage reprojection. *ICIP*, 2010.

[16] B. Smith, L. Zhang, and H. Jin. Stereo matching with non-parametric smoothness priors in feature space. *CVPR*, pages 485–492, 2009.

[17] D. Sun, S. Roth, and M. J. Black. Secrets of optical flow estimation and their principles. *CVPR*, Jun 2010.

[18] H. Tao, H. Sawhney, and R. Kumar. Dynamic depth recovery from multiple synchronized video streams. *CVPR*, 1:118–124, 2001.

[19] C. Tomasi and R. Manduchi. Bilateral filters for gray and color images. *ICCV*, pages 839–846, 1998.

[20] L. Torresani and A. Hertzmann. Automatic non-rigid 3d modeling from video. *ECCV*, pages 299–312, 2004.

[21] L. Torresani, A. Hertzmann, and C. Bregler. Non-rigid structure-from-motion: Estimating shape and motion with hierarchical priors. *PAMI*, 30(5):878–892, May 2008.

[22] A. Veeraraghavan, R. Raskar, A.K. Agrawal, A. Mohan, and J. Tumblin. Dappled photography: mask enhanced cameras for heterodyned light fields and coded aperture refocusing. *ACM Trans. Graph.*, 26(3):69, Aug 2007.

[23] Microsoft Corp. Redmond WA. *Kinect for Xbox 360*.

[24] D. M. Young. *Iterative solution of large linear systems*. Academic Press, 1971.

[25] G. Zhang, J. Jia, W. Hua, and H. Bao. Robust bilayer segmentation and motion/depth estimation with a handheld camera. *PAMI*, pages 603–617, 2011.

[26] G. Zhang, J. Jia, T.-T. Wong, and H. Bao. Consistent depth maps recovery from a video sequence. *PAMI*, pages 974–988, 2009.

[27] S. Zhu, L. Zhang, and B. M. Smith. Model evolution: An incremental approach to non-rigid structure from motion. *CVPR*, pages 1165–1172, 2010.