Efficient Graph Cut Optimization for Shape From Focus

Christophe Ribal, Nicolas Lermé, Sylvie Le Hégarat-Mascle

▶ **To cite this version:**

Christophe Ribal, Nicolas Lermé, Sylvie Le Hégarat-Mascle. Efficient Graph Cut Optimization for Shape From Focus. Journal of Visual Communication and Image Representation, 2018, 55, pp.529 - 539. 10.1016/j.jvcir.2018.06.029 . hal-01704877v2

HAL Id: hal-01704877
https://hal.science/hal-01704877v2

Submitted on 15 Nov 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Efficient Graph Cut Optimization for Shape From Focus

Christophe Ribal[a,*], Nicolas Lermé[a], Sylvie Le Hégarat-Mascle[a]

[a] *Université Paris-Sud,*
Laboratoire SATIE – UMR CNRS 8029,
Rue Noetzlin, 91190 Gif-sur-Yvette, France

Abstract

Shape From Focus refers to the inverse problem of recovering the depth in every point of a scene from a set of differently focused 2D images. Recently, some authors stated it in the variational framework and solved it by minimizing a non-convex functional. However, the global optimality on the solution is not guaranteed and evaluations are often application-specific. To overcome these limits, we propose to globally and efficiently minimize a convex functional by decomposing it into a sequence of binary problems using graph cuts. To illustrate the genericity of such a decomposition-based approach, data-driven strategies are considered, allowing us to optimize (in terms of reconstruction error) the choice of the depth values for a given number of possible depths. We provide qualitative and quantitative evaluation on Middlebury datasets and we show that, according to classic statistics on error values, the proposed approach exhibits high performance and robustness against corrupted data.

Keywords: Shape From Focus, depth map estimation, graph cuts, multi-labels.

1. Introduction

1.1. Context

Retrieving the depth of a scene from a collection of at least one image is a challenging inverse problem that is typically solved using shape-from-X ap-

*Corresponding author `christophe.ribal@u-psud.fr`

proaches (where X denotes the cue to infer the shape, e.g. stereo, motion, shading, focus, defocus, etc) or a mixture of them. This topic gave rise to a huge amount of papers and still represents a great interest for researchers in the computer vision community. Indeed, it has numerous applications, especially in robotics, both for localization and environment analysis, in monitoring or video-surveillance either for security or for medical technical assistance, or in microscopy and chemistry [1].

More specifically, let us remind that stereovision relies on the disparities between matched pixels of an image pair [2], shape-from-shading exploits the variations of brightness of a single image [3, 4] and shape-from-motion deduces depth from matched points of interest [5]. Shape-from-focus (SFF) [6] and shape-from-defocus (SFD) [7] represent alternatives approaches that share the idea of using the focus to estimate the 3D structure of a scene from differently focused images acquired by a monocular camera. Thus, an object appears focused only in a limited range (depth of field) and is progressively blurred as the camera moves away from this range. For both approaches, active and passive sensors exist, depending on whether or not a structured light composed of patterns is projected onto the scene to alleviate ambiguities. In this paper, we will focus on the passive device. In addition to the depth map, both approaches generally also provide an estimation of the all-in-focus image of the scene, i.e. the image obtained by selecting for each pixel, the intensity at which it appears the most focused, or sharp.

Now, SFF and SFD differ on one main point. SFD estimates the depth by measuring the relative blurriness between a reference image and the remaining ones. The blurring process needs to be explicitly modeled, a very few images are usually required and the approach can be applied to dynamic scenes. Similarly, [8, 9] have chosen to solve the inverse problem by precisely modeling the defocusing process with the help of an all-in-focus image. This requires the knowledge of the parameters of the camera to compute the spatially varying point spread function (PSF). In these works, the authors iteratively minimize, using Split Bregman algorithm, a regularized energy computed from the distance between

the observations and the approximated PSFs applied to the all-in-focus images.

SFF only assumes that there is an explicit relationship between the depth of a given pixel and the focal value at which it appears the most focused (or sharp). This implies the choice of an appropriate predefined operator for measuring the amount of sharpness, and a fairly large number of images to expect a good reconstruction quality of the scene. Therefore, SFF is mainly used to analyze static scenes.

In contrast to multi-cameras systems, SFF and SFD approaches allow for a more compact size of the electronic system, decrease its cost and avoid to deal with matching ambiguities. The topic is still of interest as demonstrated by recent works, e.g. [8, 10, 9, 11], including machine learning with convolutional neural networks [11].

1.2. Related work

As previously explained, solving the SFF problem implies the choice of an appropriate sharpness operator for selecting the focus maximizing the pixel sharpness. First among many, Nayar [6] introduces a sharpness operator named Summed Modified LAPlacian (SMLAP) based on second derivatives. Then, we refer the reader to the study [12] that compares a wide variety of sharpness operators in a comprehensive way.

The idea of early approaches (such as [6]) is to compute a sharpness profile over the focus values and take the argument of the maximum of this profile for every pixel. However, whatever the used sharpness operator, an estimation using raw profile is prone to errors in presence of degraded or noisy data so that different filters adapted to the sharpness profile have been proposed. In [6], a Gaussian interpolation is performed around the maximum detected on the raw profile. As an alternative to Gaussian interpolation, [13] proposed to interpolate the sharpness profile by a low-order polynomial. This idea has been then followed in [10], in which an eight-order polynomial is used.

Whatever the sharpness operator and the interpolation method used, blind techniques (i.e. that consider pixels independently of their neighbors) do not

	Our	[15]	[10]
Data term	Convex	Non-convex	Non-convex
Regularization term	Convex	Non-convex	Convex
Functional	Convex	Non-convex	Non-convex
Optimization method	Graph cut	Graph cut	ADMM
Optimality	Globally optimal	Within a known factor of the global minimum [16]	No guaranty of optimality

Table 1: Functional properties between the proposed approach, [15] and [10].

generally allow for accurate recovering the 3D geometry of a whole scene. Indeed, the sharpness operator relies on object borders that produce sharp edges on which reliable and precise depth values may be deduced. In the absence of such elements or of texture, the maximum of sharpness location tends to produce unreliable results. Ambiguities are especially present in textureless, underexposed or overexposed regions. To cope with these problems, some authors [14] proposed to reject the sharpness values being under a threshold, resulting in a globally more reliable, but sparse depth map.

Since the measurements from sharpness operator do not necessarily determine the depth uniquely, the SFF is an ill-posed problem. While formulating this kind of problem in the variational framework is a standard way to tackle it, surprisingly, only very few papers [15, 17, 10] did it. Mathematically, this amounts to the definition of a functional that embeds a data fidelity term and a smoothness (or regularization) term and that has to be (efficiently) minimized.

In [10], the variational formulation uses the negative interpolated contrast measure from Modified LAPlacian (MLAP, i.e. SMLAP restricted to a single pixel) as data fidelity term. As a result, this term is a non-convex but smooth continuous function. The regularization term used is the discrete isotropic Total Variation (TV), discontinuity-preserving, non-smooth but convex. To minimize the resulting non-convex functional, the data term is linearized and an iterative algorithm, namely Alternating Direction Method of Multipliers (ADMM) is applied. According to the authors, this algorithm provably converges toward

a critical point of the functional but no optimality guarantees are mentioned about the solution. Although the proposed algorithm seems to give good results and exhibit good convergence properties, it has been actually evaluated only qualitatively and on few real images.

The work of [15] also uses the sharpness operator MLAP. The data fidelity term is the truncated quadratic difference between the maximum value of sharpness and the tested sharpness. This term is therefore non-convex. The smoothness term is a truncated L^2 norm (then also non-convex) that is discontinuity-preserving. The truncation depends on whether a significant texture is present or not. The algorithm used for the minimization of the resulting non-convex functional is the α-expansion based on graph cuts [16]. Interesting results are obtained but the approach is prone to get easily trapped in local minima of the energy and in [15], the evaluation is limited to application-specific images (optical microscopes).

1.3. Outline of the proposed approach

In this work, we explore a new way to solve the SFF problem by directly minimizing, for a given depth resolution, a *convex* functional. The advantage of such a choice is twofold: (i) The optimality about the solution is easier to guarantee and (ii) the convexity property can be exploited to use fast minimization procedures. Functional properties of the aforementioned approaches against ours are summarized in the Table 1.

Our choice focuses on graph cuts because of their well-founded theoretical background [18] and the existence of a fast maximum-flow/minimum-cut algorithm [19]. While [20] has optimality guarantees for convex priors, the graph construction requires a lot of computational resources (in terms of time and memory). Alternatives, like the α-expansion [16], allow for minimizing the functional iteratively by solving sequentially binary problems until convergence, but without any guaranty relatively to the number of iterations required.

Thanks to a discretization step, the functional can nevertheless be exactly minimized when the data fidelity term is convex, by mapping the original prob-

lem to a deterministic number of independent binary problems (each one solved using graph cuts) [21]. Each subproblem boils down to choosing a split value along the depth dimension and labeling the depth map accordingly. Given a number of binary problems, the dyadic strategy is an usual efficient way to select these split values, but a data-driven splitting strategy allows for lowering the reconstruction error, especially when the fixed number of discrete depth values is low. Another beneficial effect is to balance the sizes of the subproblems, thus reducing the complexity of the divide-and-conquer approach.

Figure 1 gives the outline of our approach. Our optimization algorithm is based on graph cuts (bottom right rectangular box on Fig. 1). Besides data images and regularization parameter λ, it takes as an input the tree of *split values*, i.e. the values that define hierarchically the subproblems. In our case, the split value tree (bottom left diamond box on Fig. 1) is built accordingly to the depth histogram estimated from the sharpness profiles (middle diamond box on Fig. 1), and depends on the selected splitting strategy.

Our contributions are in the definition of a convex functional that is fitted with the graph cut minimization and in the proposition of data-driven strategies for the split value tree estimation. In what follows,

- The chosen sharpness operator, the interpolation, and the proposed convex functional are described in Section 2.
- In Section 3, we provide the algorithm for exactly minimizing such functional for a given split value tree, and secondly we discuss and investigate the selection of the split values.
- Section 4 first compares the respective performances achieved by the variants of sharpness operators with full resolution against [10]. Then, it analyzes the impact of the splitting strategy on the reconstruction error.
- Finally, Section 5 recalls the main results and presents future work.

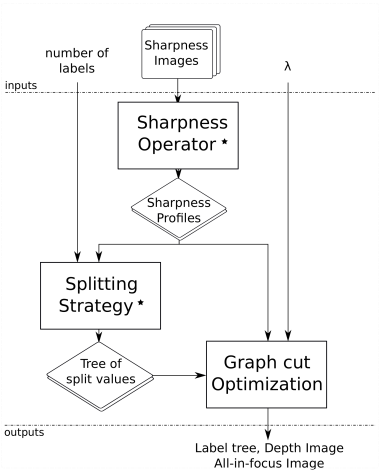


Figure 1: Flowchart representing the approach chosen to solve our problem. In this paper, different implementations are explored for starred boxes.

2. Proposed functional

To take advantage of efficient minimization procedures based on graph cuts, we propose to use a convex functional. Let us first introduce some notations before detailing it.

For a given positive integer $K_0 > 0$, let us denote by Φ_0 the sharpness operator and let us define the finite sets $\mathcal{K}_0 = \{0, \ldots, K_0 - 1\}$ and $\mathcal{L}_0 = \{l_k\}_{k \in \mathcal{K}_0}$ with $l_k \in \mathbb{R}, \forall k \in \mathcal{K}_0$. Moreover, we denote by $\{I_k\}_{k \in \mathcal{K}_0}$ the set of focused images where $I_k : \mathcal{P} \subset \mathbb{Z}^2 \to \mathbb{R}^M$ is a M-channels ($M > 0$) image defined over lattice $\mathcal{P}$ and acquired with focus values l_k, for any $k \in \mathcal{K}_0$.

The proposed approach consists of two steps: (i) A blind estimation of depth is performed for any pixel $p \in \mathcal{P}$ independently of its neighbors and (ii) this estimation is used to setup the data term of the functional that will then be minimized to derive the optimal depth map solution. These steps are detailed in the subsequent sections.

2.1. Sharpness profiles

The model used to interpret the physical process blurring the image is based on geometrical optics. For any given 3D point, moving away the sensor from

it distributes the energy over circular patches (in the image) with a radius increasing with the sensor displacement. This defines the PSF that acts in the frequency domain as a low-pass filtering on the all-in-focus image.

A common choice among sharpness operators is SMLAP since it presents similar performance for a shorter processing time compared to alternative operators [12]. For every pixel $p \in \mathcal{P}$ and every index $k \in \mathcal{K}_0$, this operator is defined as

$$\Phi_0(p,k) = \sum_{\substack{q \in \Omega(p), \\ q=(i,j)}} \left(\left\| \frac{\partial^2 I_k(q)}{\partial i^2} \right\|_1 + \left\| \frac{\partial^2 I_k(q)}{\partial j^2} \right\|_1 \right) = \Phi_0(p, l_k), \tag{1}$$

where $\|.\|_1$ is the L^1 norm in $\mathbb{R}^M$, $\Omega(p) \subset \mathbb{Z}^2$ is the neighborhood of pixel p (typically a small squared window of fixed size), $I_k(q)$ denotes the intensity of image I_k at pixel q and Laplacian operators are approximated by finite differences. Note that MLAP can be deduced from Eq. (1) by restricting $\Omega(p)$ to pixel p. The above operator Φ_0 is used in Section 4.

Once the sharpness operator has been applied to the sequence of focused images $\{I_k\}_{k \in \mathcal{K}_0}$, the resulting measurements are usually filtered. The benefit of interpolating sharpness profiles is twofold: (i) It increases the robustness against potential degradations (noise, contrast, etc.), and (ii) if a high depth resolution is required by the application, it enables us to reduce the discretization step along the depth dimension. Polynomial [10] and Gaussian [6] interpolations are two standard techniques. In this work, to take into account the increase of the depth of field with distance, we propose a Gaussian filtering of the sharpness profile with a standard deviation that linearly depends on the focal value (i.e. the distance of the object plane to the optical center). In Fig. 2, the raw sharpness profiles as well as those filtered by the aforementioned techniques are drawn (right subplot) for three distinct locations depicted on the all-in-focus image. For each profile and each location, the position achieving the maximum sharpness value is also indicated. In textured regions (such as for shown pixel 3), the maxima found are very close to the ideal one and therefore

all filtering techniques perform well. The obtained depth estimates however differ for textureless regions (such as for pixel 1 and pixel 2). It can be observed that the polynomial interpolation (used in [10]) presents some oscillations. It may induce some errors on blind estimates when the profile is flat (e.g. in the absence of texture). Even if, based on these observations, we favor the proposed Gaussian filtering, note that our approach is generic with respect to the sharpness profile (middle diamond box on Fig. 1).

Let us denote by $\Phi(p, .)$ the interpolated sharpness profile of any pixel $p \in \mathcal{P}$, whatever the sharpness operator Φ_0 used. For some integer $K > 0$, let us also define the sets $\mathcal{K} = \{0, \ldots, K-1\}$ and $\mathcal{L} = \{l_k\}_{k \in \mathcal{K}}$ with $l_k \in \mathbb{R}$, $\forall k \in \mathcal{K}$.

Please note that the focus values $\{l_k\}_{k \in \mathcal{K}}$ are not necessarily equally spaced along depth dimension: For targeted applications, some preference can be given to some specific range of values. The blind depth estimates can now be formally defined as

$$\mathbf{v} = \left\{ v_p \,\middle|\, v_p \in \underset{l_k \in \mathcal{L}}{\operatorname{argmax}} \{\Phi(p, l_k)\} \right\}_{p \in \mathcal{P}}. \tag{2}$$

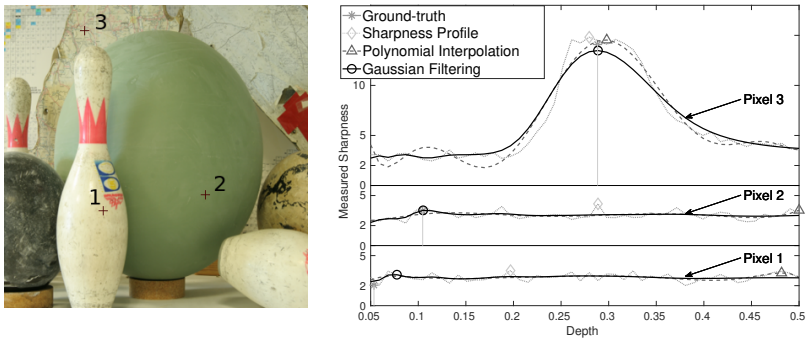


Figure 2: Influence of the filtering techniques on the blind depth estimates for three cross-marked pixels shown on the all-in-focus image (left) and their corresponding sharpness profiles (right).

2.2. Functional

In previous section, $\mathcal{L}$ is the set of depth values on which the sharpness profile is estimated. In the general case, the depth map estimation is performed

on a set of labels that is a subset of $\mathcal{L}$. Then, we introduce the following notations. For any positive integer $\tilde{K} \leq K$, we define the sets of cardinality $\tilde{K}$, $\tilde{\mathcal{K}} = \{k_0, \ldots, k_{\tilde{K}-1}\} \subseteq \tilde{K}$ and $\tilde{\mathcal{L}} = \{l_k \in \mathbb{R}\}_{k \in \tilde{\mathcal{K}}} \subseteq \mathcal{L}$. $\tilde{\mathcal{K}}$ is the set of indices of $\tilde{\mathcal{L}}$ labels in $\mathcal{L}$. Then, given the blind estimate $\mathbf{v} \in \mathcal{L}^{\mathcal{P}}$ obtained using Eq. (2), for any depth map taking its values in $\tilde{\mathcal{L}}$ denoted $\mathbf{x}(\tilde{\mathcal{L}})$ such that $\mathbf{x}(\tilde{\mathcal{L}}) \in \tilde{\mathcal{L}}^{\mathcal{P}}$, the functional to minimize with respect to $\mathbf{x}(\tilde{\mathcal{L}})$ is:

$$E(\mathbf{x}(\tilde{\mathcal{L}})) = \sum_{p \in \mathcal{P}} U_p(x_p) + \lambda \sum_{(p,q) \in \mathcal{N}} V_{p,q}(x_p, x_q), \tag{3}$$

where $\mathcal{N} \subset (\mathcal{P} \times \mathcal{P})$ is the set of adjacent pixel pairs, $U_p(x_p)$ is the data fidelity term measuring the cost of assigning the label x_p to the pixel p with respect to blind estimate $\mathbf{v}$, $V_{p,q}(x_p, x_q)$ is the regularization term that is chosen to penalize the difference of labeling between pixels p and q, and λ is a non-negative weighting parameter determining the balance between both terms.

From Eq. (3), the general problem, that consists of optimizing E with respect to both the label set $\tilde{\mathcal{L}}$ and the label image $\mathbf{x}$, is not straightforward. Thus, it has been split into two subproblems. On the one hand, for a given set $\tilde{\mathcal{L}}$, we have to minimize E with respect to $\mathbf{x}(\tilde{\mathcal{L}})$, i.e. $\mathbf{x}$ pixels taking values only in $\tilde{\mathcal{L}}$. On the other hand, for a given $\tilde{K}$, we have to build the subset of labels $\tilde{\mathcal{L}}$ that minimizes the Root Mean Square Error (RMSE) between $\widehat{\mathbf{x}}(\tilde{\mathcal{L}})$ and $\widehat{\mathbf{x}}(\mathcal{L})$ (where $\widehat{\ }$ denotes a minimizer of the functional). In the following, this second subproblem is referred as the choice of the splitting strategy. For the sake of concision, when dealing with the first subproblem (namely from here to Subsection 3.3), we omit $\tilde{\mathcal{L}}$ (that is actually fixed) in the notation of $\mathbf{x}(\tilde{\mathcal{L}}) = \mathbf{x}$.

In Eq. (3), the data term is defined for any pixel $p \in \mathcal{P}$ as the weighted L^α norm ($\alpha \geq 1$) between blind estimate v_p (see Eq. (2)) and label x_p, i.e.

$$U_p(x_p) = \eta_p |x_p - v_p|^\alpha, \tag{4}$$

where η_p is a coefficient independent of x_p. η_p is computed from the sharpness

profile and the blind estimate:

$$\eta_p \propto \left(\frac{K \left(\Phi(p, v_p) - \Phi(p, z_p) \right)^2}{\sum_{k \in K} \left(\Phi(p, l_k) - \Phi(p, z_p) \right) + \varepsilon} \right) \in [0, K(\Phi(p, v_p) - \Phi(p, z_p))[\,,$$

with $\varepsilon \gtrsim 0$ and z_p is defined for any pixel $p \in \mathcal{P}$ as $z_p = \text{argmin}_{k \in \mathcal{L}} \{\Phi(p, l_k)\}$.

Indeed, the dynamic range of the sharpness profile $\Phi(p, .)$ varies with the pixels depending on the local texture. η_p is proportional to this dynamic range divided by the normalized area under the sharpness profile $\Phi(p, .)$ (ε avoiding division by zero). Therefore, η_p measures the local significance of the difference between x_p and v_p and weights the data fidelity term accordingly.

The regularization term in Eq. (3) corresponds to the anisotropic total variation (TV). For any pixel pair $(p, q) \in \mathcal{N}$, it is expressed as

$$V_{p,q}(x_p, x_q) = w_{p,q}(x_p - x_q)^+, \tag{5}$$

where $a^+ = \max\{a, 0\}$ and $w_{p,q}$ are fixed positive coefficients (see [22]). Despite some undesired behaviors of TV such as "staircase effect", i.e. creation in the depth map of flat regions separated by artifact boundaries, this operator enjoys desirable properties (convexity, discontinuity-preserving of image boundaries, etc.) and it has been successfully applied to numerous applications and problems such as in image restoration when $\alpha = 1$ or $\alpha = 2$ (see [22] and the references therein). The above definition of the TV is general. In particular, it allows us to have $w_{p,q} \neq w_{q,p}$. While neighborhoods taking into account complex relationships between pixels could be considered, we only use the 8-connexity in the experimental results presented in Section 4.

3. Depth estimation using graph cuts

We now describe how the graph cut based approach [21] for exactly minimizing discrete convex functionals (like the functional (3) presented in Section 2.2) can be used for solving our problem, the set of labels $\tilde{\mathcal{L}}$ being given. Firstly,

we remind how this problem can be mapped to a set of independent subproblems which only involve binary variables. Secondly, we recall how each of these subproblems can be efficiently solved using a maximum-flow/minimum-cut algorithm and how their number can be drastically reduced using a divide-and-conquer process with a dichotomous splitting strategy. The most common splitting strategy is the dyadic one. However, since it corresponds to a special case of $\tilde{\mathcal{L}}$, other splitting strategies are investigated, i.e. we discuss the estimation of $\tilde{\mathcal{L}}$ itself.

3.1. Leveled-energy decomposition

In the sequel, we assume that the label values of $\tilde{\mathcal{L}}$ are ordered with respect to their indices in $\tilde{\mathcal{K}}$: $l_{k_i} > l_{k_{i-1}}, \forall i \in \{1, \ldots, \tilde{K}-1\}$. As explained in [21], the data term (see Eq. (4)) and the regularization term (see Eq. (5)) of the functional (see Eq. (3)) can be decomposed as a sum of energies on the level sets of $\mathbf{x}$, with $\mathbf{x} \in \tilde{\mathcal{L}}^{\mathcal{P}}$. Let us denote by χ_p^l the value in pixel p of $\boldsymbol{\chi}^l = \mathbf{1}_{\{x_p \geq l\}} \in \{0,1\}^{\mathcal{P}}$ the l-level set of image $\mathbf{x}$. For any pixel $p \in \mathcal{P}$, the data term can be decomposed as

$$U_p(x_p) = \left(\sum_{0<i<\tilde{K}} \chi_p^{l_{k_i}} \left(U_p(l_{k_i}) - U_p(l_{k_{i-1}}) \right) \right) + U_p(l_{k_0}). \tag{6}$$

Note that the latter equation is consistent whatever $x_p \in \tilde{\mathcal{L}}$. Similarly, for any pixels pair $(p,q) \in \mathcal{N}$, the regularization term can be decomposed as

$$V_{p,q}(x_p, x_q) = \sum_{0<i<\tilde{K}} \underbrace{w_{p,q}(\chi_p^{l_{k_i}} - \chi_q^{l_{k_i}}))^+}_{V_{p,q}(\chi_p^{l_{k_i}}, \chi_q^{l_{k_i}})}. \tag{7}$$

In the latter expression, the sum on i starts from $i = 1$ since $\chi_p^{l_{k_0}} = \chi_q^{l_{k_0}} = 1$, $\forall (p,q) \in \mathcal{N}$. Using Eq. (3), Eq. (6) and Eq. (7), the functional may now be written

$$E(\mathbf{x}) = \sum_{0<i<\tilde{K}} E^{l_{k_i}}(\boldsymbol{\chi}^{l_{k_i}}) + C, \tag{8}$$

where C is a constant that does not depend on $\mathbf{x}$ and the energy $E^{l_{k_i}}$ is defined, for any $0 < i < \tilde{K}$ and any binary image (level set) $\boldsymbol{\chi}^{l_{k_i}} \in \{0,1\}^{\mathcal{P}}$, by

$$E^{l_{k_i}}(\boldsymbol{\chi}^{l_{k_i}}) = \sum_{p \in \mathcal{P}} \chi_p^{l_{k_i}} (U_p(l_{k_i}) - U_p(l_{k_{i-1}})) + \lambda \sum_{(p,q) \in \mathcal{N}} V_{p,q}(\chi_p^{l_{k_i}}, \chi_q^{l_{k_i}}). \tag{9}$$

For any $k, k' \in \tilde{\mathcal{K}} \backslash \{k_0\}$, let us denote by $\widehat{\boldsymbol{\chi}}^{l_k}, \widehat{\boldsymbol{\chi}}^{l_{k'}} \in \{0,1\}^{\mathcal{P}}$ minimizers of E^{l_k} and $E^{l_{k'}}$ respectively. If these minimizers satisfy

$$\widehat{\chi}_p^{l_k} \geq \widehat{\chi}_p^{l_{k'}}, \quad \forall 0 \leq k \leq k' \leq \tilde{K} - 1, \quad \forall p \in \mathcal{P}, \tag{10}$$

i.e. the level sets $\widehat{\boldsymbol{\chi}}^{l_k}$ are nested, then, from Eq. (8), we can check that the level set $\hat{\mathbf{x}} \in \tilde{\mathcal{L}}^{\mathcal{P}}$ defined for all $p \in \mathcal{P}$, by

$$\hat{x}_p = \max \{l_k \in \tilde{\mathcal{L}} \mid \widehat{\chi}_p^{l_k} = 1\}, \tag{11}$$

minimizes Eq. (8). According to [21], if the condition (10) holds for data fidelity term (which is the case here since the data term of Eq. (4) is convex), a minimizer of E can be deduced from all the minimizers of $\{E^{l_k}\}_{k \in \tilde{\mathcal{K}} \backslash \{k_0\}}$. We now present how every binary problem E^{l_k} can be efficiently solved using graph cuts.

3.2. Graph cut minimization

Due to limited resources and algorithm developments, graph cuts remained bounded to binary image restoration for a long time [23]. The emergence of a fast maximum-flow/minimum-cut algorithm [19] coupled to a better characterization of what energies can be minimized [18], was a milestone for solving challenging vision tasks such as segmentation, restoration, stereovision, etc.

In particular, [18] provides a key result about the conditions for the applicability of the approach: Submodularity of pairwise terms is a necessary and sufficient condition for minimizing a functional. In our case, since TV is submodular, this condition is verified for Eq. (3).

For minimizing every E^{l_k} (see Eq. (9)) using graph cuts, we adopt the graph construction detailed in [18]. Let us consider a weighted and oriented graph

$\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where $\mathcal{V} = \mathcal{P} \cup \{s, t\}$ is the set of nodes (s and t are named terminal nodes) and $\mathcal{E} = \mathcal{N} \cup \{(s,p)\}_{p\in\mathcal{P}} \cup \{(p,t)\}_{p\in\mathcal{P}}$ is the set of edges (edges connecting s or t are named t-links while remaining edges are named n-links). Then, we assign a non-negative capacity to any edge $(p,q) \in \mathcal{E}$ as follows:

$$\left\{ \begin{array}{lcll} c_{s,p} & = & (U_p(l_{k_i}) - U_p(l_{k_{i-1}}))^-, & \forall p \in \mathcal{P}, \\ c_{p,t} & = & (U_p(l_{k_i}) - U_p(l_{k_{i-1}}))^+, & \forall p \in \mathcal{P}, \\ c_{p,q} & = & \lambda w_{p,q}, & \forall (p,q) \in \mathcal{N}, \end{array} \right. \tag{12}$$

where $(a)^- = \max\{-a, 0\}$. For any $\mathcal{S} \subseteq \mathcal{P}$, we define the value of the s-t cut $(\mathcal{S} \cup \{s\}, (\mathcal{P} \setminus \mathcal{S}) \cup \{t\})$ in the graph $\mathcal{G}$ by

$$\mathrm{val}_{\mathcal{G}}(\mathcal{S}) = \sum_{\substack{p\in(\mathcal{S}\cup\{s\}) \\ q\in((\mathcal{P}\setminus\mathcal{S})\cup\{t\})}} c_{p,q}.$$

For any $\mathcal{S} \subseteq \mathcal{P}$, we also define $\boldsymbol{\chi}^{\mathcal{S}}$ the binary image such that for every $p \in \mathcal{P}$

$$\chi_p^{\mathcal{S}} = \left\{ \begin{array}{lll} 0 & \text{if} & p \in (\mathcal{S} \cup \{s\}), \\ 1 & \text{if} & p \in ((\mathcal{P} \setminus \mathcal{S}) \cup \{t\}). \end{array} \right.$$

There is a one-to-one correspondence between the sets $\mathcal{S}$ (being elements of the powerset of $\mathcal{P}$) and the elements of $\{0,1\}^{\mathcal{P}}$. Using the edge capacities (12) as well as the definitions (9) and (5), it is straightforward to see that $\mathrm{val}_{\mathcal{G}}(\mathcal{S})$ is equal to $E^{l_k}(\boldsymbol{\chi}^{\mathcal{S}})$, up to a constant that is independent of $\boldsymbol{\chi}^{\mathcal{S}}$. If $(\widehat{\mathcal{S}} \cup \{s\}, (\mathcal{P} \setminus \widehat{\mathcal{S}}) \cup \{t\})$ is a minimum s-t cut (s-t cut of minimum weight) in the graph $\mathcal{G}$, $\boldsymbol{\chi}^{\widehat{\mathcal{S}}}$ is thus a minimizer of E^{l_k}. This minimizer can be efficiently computed using a maximum-flow/minimum-cut algorithm such as [19]. Although it has a pseudo-polynomial worst-case complexity depending on the value of the minimum s-t cut, its near-linear behavior still makes it attractive for many computer vision problems.

More generally, the minimization of the functional Eq. (3) requires the computation of precisely $\tilde{K}$ s-t minimum cuts, which is time-consuming when $\tilde{K}$ is large. Due to the monotone condition (10), binary solutions are nested. The divide-an-conquer process proposed in [21] takes advantage of this property and

via the tree traversal of a tree of split values, defining a set of independent binary problems. It allows us to decrease drastically the number of s-t minimum cuts until $\lfloor \log_2(\tilde{K}) \rfloor$.

3.3. Data-driven decomposition

The key point for reducing the number of used graph cuts (until $\lfloor \log_2(\tilde{K}) \rfloor$ instead of $\tilde{K}$) is the definition of a hierarchical tree having $\tilde{K}$ leaves and whose nodes encodes the binary subproblems. The tree derived from the dyadic splitting provides, for a given $\tilde{K}$, a deterministic number of graph cuts, namely $\lfloor \log_2(\tilde{K}) \rfloor$ that is the depth of the label tree, also later referred to as iteration or cut number. However, it does not necessarily yield the best results in terms of RMSE between $\widehat{\mathbf{x}}(\tilde{\mathcal{L}})$ and $\widehat{\mathbf{x}}(\mathcal{L})$. Here $\widehat{\mathbf{x}}(\mathcal{L})$ is the optimal depth map considering an extended set of labels $\mathcal{L}$ relative to the actually used set $\tilde{\mathcal{L}}$. Then, the mentioned RMSE evaluates the errors between an ideal depth map with the reconstructed one having a limited number of labels. In order to minimize this error for a given $\tilde{K}$, we investigate constructions of the label tree based on automatic data-driven decompositions as alternatives to the dyadic splitting.

The idea is to base the choice of the split values (used at each iteration) on the depth histogram of the considered data. Then, instead of thresholds corresponding to dyadic splitting of the whole depth interval, we will derive adaptive values based on this depth histogram. Let us denote by τ_j^i the split values used at iteration i, with i_{end} the final number of iterations, $i \in \{1, \ldots, i_{end} - 1)\}$, $j \in \left\{1, \ldots, 2^{i-1}\right\}$. At each iteration, the label values are computed in a deterministic way (cf. fourth comment further in the section) within the intervals defined by the τ_j^i set.

Without loss of generality, we consider depth interval equal to $[0, 1]$ (linear transformation is trivial for other interval bounds). Considering the dyadic splitting, extending threshold notation so that $\tau_0^i = 0, \forall i \in \{0, \ldots, i_{end}\}$, the set of τ_j^i values at iteration i is independent of the data, namely $\left\{\tau_j^k + \frac{1}{2^i}, \forall 0 \leq k < i\right\}$. Now, considering a data-driven approach, the most prevalent depths drive the

splitting strategy, in order to provide more details (and therefore a more accurate depth map) for the main objects of the scene.

Two adaptive splitting ways have been considered, namely:

- the median splitting, where the chosen thresholds correspond to the median values of the depth histogram restricted to the interval to split. Specifically, τ_1^1 is the median of the whole histogram interval, τ_1^2 is the median of the histogram restricted to $\left[0, \tau_1^1\right[$ and τ_2^2 is the median of the histogram restricted to $\left[\tau_1^1, 1\right]$, and so on.

- Otsu's splitting, where the chosen split values derive from Otsu algorithm [24] applied to the depth histogram restricted to the interval to split. Similarly to the median splitting case, τ_1^1 is the Otsu threshold of the whole histogram interval, τ_1^2 is the Otsu threshold of the histogram restricted to $\left[0, \tau_1^1\right[$ and so on.

At this stage, five comments have to be made.

- Firstly, as required (for ensuring monotone condition (10)), for any proposed tree of labels (dyadic, median, Otsu), the level sets are nested and thus the graph cut based approach still operates as previously depicted. In the same way, Eq. (11) remains applicable for $\widehat{\mathbf{x}}$ estimation.

- Secondly, since the actual depth histogram is unknown, we use the blind depth map to derive an estimation of the depth histogram. Indeed, even if the blind depth map has numerous errors, we assume that it is sufficiently correct in terms of statistics to allow for the choice of adaptive thresholds more appropriated than the dyadic splitting.

- Thirdly, the choice of Otsu's algorithm stems from the fact that, theoretically, it yields the smallest RMSE between $\widehat{\mathbf{x}}(\tilde{\mathcal{L}})$ and $\widehat{\mathbf{x}}(\mathcal{L})$. However, both due to discretization of the labels and regularization step, the achieved RMSE value between $\widehat{\mathbf{x}}(\tilde{\mathcal{L}})$ and the ground truth cannot be predicted.

- Fourthly, to compute RMSE values, we need the depth value (label) associated to each interval defined by two consecutive split values. To minimize the RMSE, rather than the interval centers, we propose to consider the interval centroids (i.e. mean) values.

- Fifthly, the adaptive splitting is mainly relevant for small values of $\tilde{K}$. Indeed, for large values of $\tilde{K}$, the different sets of leaves (intervals and associated labels $\tilde{\mathcal{L}}$) converge toward the same set, $\mathcal{L}$.

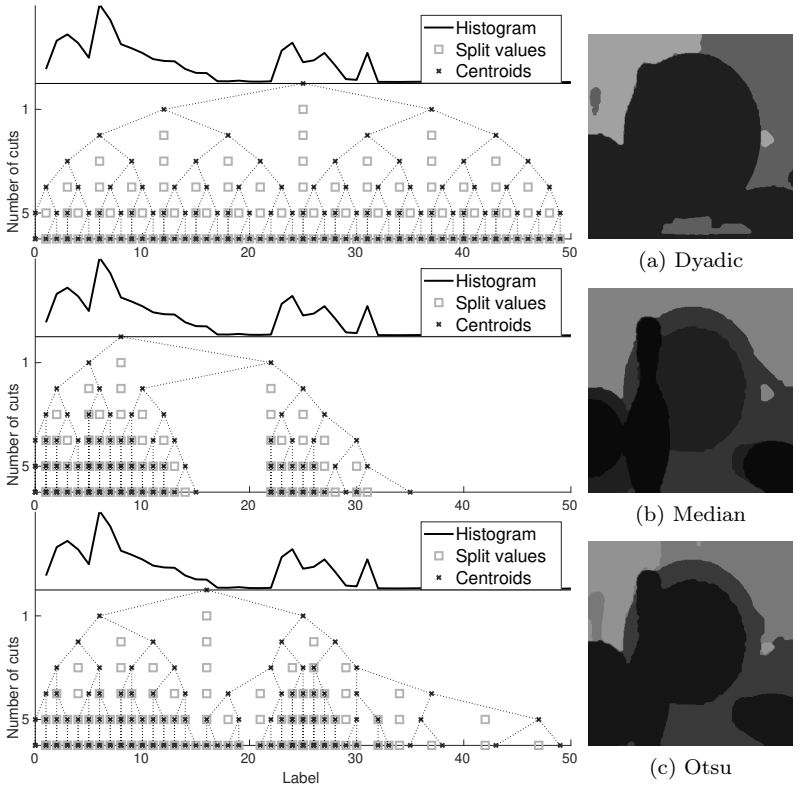


Figure 3: Illustration of the different splitting strategies, for $\lambda = 0.025$; 1^{st} column: Tree of the centroids, also showing the split values versus the iteration number, 2^{nd} column: 4-valued depth map achieved at iteration 2.

Figure 3 illustrates the different splitting strategies: Dyadic, median or Otsu's way as proposed. The trees (below the blind histogram of the image) show the split values (squares) and the centroids (crosses) versus the iterations. For this figure, the different iterations can also be interpreted as different results

when increasing the cardinality of $\tilde{K}$. The centroid set provides the depth values used for depth map estimation. On the right part of the figure, the four-labels image corresponds to the second graph cut iteration. According to this example, we clearly see that Otsu's and median splitting strategies efficiently retrieve the main objects for small values of $\tilde{K}$.

4. Numerical experiments

4.1. Data and evaluation measures

The dataset on which we focused for our experiments is derived from the Middlebury college dataset from 2005 and 2006 [25]. This dataset provides, for various realistic scenes, accurate depth maps as well as colored all-in-focus images (here, $M = 3$), with several available exposures and illumination settings. Among them, we have selected the intermediate exposure, the lowest illumination and smallest image resolution, for both views 1 and 5. The unknown depth values due to occlusions have been estimated by the median value of the surrounding depths.

From this patched dataset, we generate the sequence of K_0 focused images for each scene using the code provided by Pertuz[1] run with default values of parameters: Each pixel of the all-in-focus image is blurred depending on the distance between its actual depth and the image focal plane. In the following experiments, we present the results obtained from datasets simulated with this software adapted to the usage of a depth map and a colored all-in-focus image, with $K_0 = 30$ and $K_0 = 50$ images. We furthermore add noise on the images obtained by adding normally distributed random values (centered on 0, of standard deviation $\sigma \in \{\sigma_0 = 0, \sigma_1 = 0.005, \sigma_2 = 0.01\}$) to the float intensity images scaled to $[0, 1]$. Note that the noise images are uncorrelated along the depth dimension.

To evaluate the performance of our SFF algorithm, we propose to estimate

[1] `https://fr.mathworks.com/matlabcentral/fileexchange/55103-shape-from-focus`

quantitatively the accuracy of our estimation $\mathbf{x} \in \tilde{\mathcal{L}}^{\mathcal{P}}$ against the ground truth $\mathbf{y} \in \mathcal{L}^{\mathcal{P}}$ using four metrics: (i) The RMSE; Then, computed from the histogram of the absolute error values, (ii) the median and (iii) the 90^{th} percentile; And (iv) the Structural Similarity Index (SSIM, [26]). To remove the dependency on the dynamic of the scene (denoted by Δ), we scale RMSE values by Δ:

$$RMSE(\mathbf{x}, \mathbf{y}) = 100 \sqrt{\frac{1}{\sharp\mathcal{P}\Delta^2} \sum_{p \in \mathcal{P}} \left(x_p - y_p\right)^2} \in [0, 100],$$

where $\sharp$ denotes the cardinality of a set.

Then, absolute error distribution provides a complementary evaluation (for instance it is less sensitive to outliers that the RMSE criterion). For these three metrics (i-iii), the lower the achieved values, the better the results are. Finally, the SSIM is also complementary since it evaluates the correlation between estimation and ground truth (ideal estimation). We use the version specified in [26], with $\alpha = \beta = \gamma = 1$ so that SSIM is defined by:

$$SSIM(\mathbf{x}, \mathbf{y}) = \frac{1}{\sharp\mathcal{P}'} \sum_{p \in \mathcal{P}'} \frac{\left(2\bar{x}_{\Omega(p)}\bar{y}_{\Omega(p)} + C_1\right)\left(2\sigma_{xy_{\Omega(p)}} + C_2\right)}{\left(\bar{x}^2_{\Omega(p)} + \bar{y}^2_{\Omega(p)} + C_1\right)\left(\sigma^2_{x_{\Omega(p)}} + \sigma^2_{y_{\Omega(p)}} + C_2\right)} \in [-1, 1]\,,$$

where $\mathcal{P}'$ is the set of the centers p of the used windows $\Omega(p)$ of size 7×7, $\bar{x}_{\Omega(p)}$, $\bar{y}_{\Omega(p)}$ are the means on $\Omega(p)$ of x and y values respectively, and $\sigma_{x_{\Omega(p)}}$, $\sigma_{y_{\Omega(p)}}$, and $\sigma_{xy_{\Omega(p)}}$ are the variances and covariance. Finally, the constants C_1 and C_2 are computed from Δ as $C_1 = (0.01\Delta)^2$ and $C_2 = (0.03\Delta)^2$. For metric (iv), the larger the achieved values, the better the results are.

4.2. Benefit of proposed energy model

The aim of this subsection is to check the usefulness of the regularization process based on the proposed energy model. Assuming that Moeller's work [10][2]

[2]The CUDA/C++ code of the parallelized GPU version is publicly available on the webpage `https://github.com/adrelino/variational-depth-from-focus`

represents the state-of-art (of variational SFF methods), we compare it to three variants of our method with $\tilde{K} = K = K_0$. Faced to the number of algorithm parameters to tune, we run [10] algorithm with its default value parameters (except λ_M parameter that was fitted, see Fig. 5), having checked that these default values provide rather satisfying results. For our approach, three variants of the functional (3) were implemented in OpenCV/C++ with a 8-neighbors graph connexity. Even though the introduced noise is Gaussian, we empirically observe that it yields to a noise likened to impulsive noise on maximal sharpness values and blind estimated depths (see Fig. 4b). Based on this observation, we therefore have chosen $\alpha = 1$ in data fidelity term (Eq. (4)).

The three variants, represented in the box named *Sharpness Operator* in Fig. 1, only differ by the data fidelity term, namely either based on the polynomial interpolation of MLAP sharpness profile (that is also the blind estimate of [10]), or on the Gaussian filtering of MLAP sharpness profile, or on the proposed Gaussian filtering of SMLAP sharpness profile (Eq. (1) with Ω window of size 7×7). These three variants are called 'Graph cut PM', 'Graph cut GM' and 'Graph cut GS', respectively. For the the 'Graph cut GM/GS' variants, the standard deviation of the Gaussian filter is determined empirically according to the relationship $\sigma(k) = 0.2k+1$, where k is the index of the focused image in the sequence. For each of the above variants, the blind estimation is derived considering $\lambda = 0$ whereas the regularized ones correspond to $\lambda > 0$. We vary the parameter λ within a fixed interval. This allows us to observe the behavior of the algorithm with respect to the regularization parameter as well as to get the λ value achieving the minimal RMSE value (denoted by λ^*) in the considered λ interval.

Figure 4 gives a qualitative comparison of some results obtained in the case of the *Art* image example. Specifically, the first column shows the all-in-focus image (last row), the depth ground truth (first row) and 'optimal' result of [10] (middle row). The three following columns allows us to compare the 'Graph cut PM', 'Graph cut GM' and 'Graph cut GS' results with the blind estimations shown on first row, the λ^*-regularized results shown on second line and examples

of over-regularized results on the last line. Main comments are:

- The benefit of the proposed data fidelity term (SMLAP, Gaussian) is visible when comparing Fig. 4b, Fig. 4c and Fig. 4d: Firstly, conversely to polynomial interpolation, Gaussian filtering avoids oscillations and secondly, SMLAP allows for early 2D-spatial filtering of high frequencies of depth map. Besides, Fig. 4f, Fig. 4g and Fig. 4h show that the data fidelity term also impacts the result obtained after regularization.

- The regularization allows for the removal of noise in blind estimation. Using the optimal regularization parameter, the proposed model allows for much better preservation of details and fine structures than with [10] (that may seem a little bit too regularized even though it is the best result achieved when varying the weight of the regularization term).

- Increasing furthermore the regularization parameter (beyond its optimal value), spatial details are wiped out whereas the overall shape of objects in the scene is well preserved and remains visible.

To evaluate quantitatively the usefulness of the proposed method, on Fig. 5, we plot the RMSE values versus the regularization parameter λ. As previously, the considered SFF methods are [10], 'Graph cut PM', 'Graph cut GM' and 'Graph cut GS'. RMSE values are averaged over all the images of the considered dataset. On the first line of Fig. 5, the subgraphs correspond to the two considered datasets with either $K_0 = 30$ or $K_0 = 50$ images with a given noise level (intermediate) whereas on the second line the noise level varies for a given dataset ($K_0 = 30$). In the presented graphs, the λ scale is those used for the models 'Graph cut PM/GM/GS', whereas regularization parameter λ_M of [10] is derived using $\lambda_M = 10^{3+3\times log_{10}(\lambda)}$. From Fig. 5, we observe that:

- The RMSE curves are consistent with the qualitative results depicted in Fig. 4: Whatever the variational method (among the considered ones), increasing λ until λ^* allows for RMSE decrease (by removing blind estimation noise) but when increasing λ beyond λ^*, the RMSE value increases

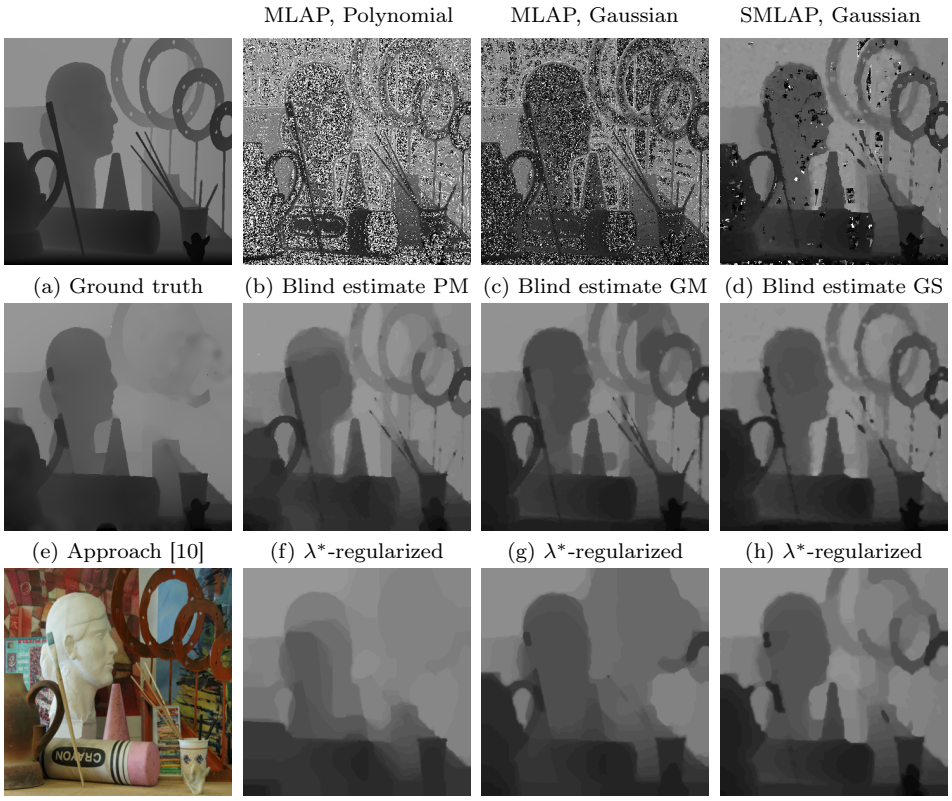


Figure 4: Examples of SFF depth maps associated to all-in-focus (i) and ground truth (a) images; Shown results correspond to: [10] algorithm (e) and our model (see Eq. (3)) when varying the blind estimation (from MLAP polynomial interpolation to proposed SMLAP Gaussian filtering) and the regularization parameter.

(by removing relevant thin structures of the scene).

- The minimal average RMSE value depends on the noise level (see Fig. 5a, 5c and 5d).

- In terms of averaged RMSE, the best results are achieved by the proposed 'Graph cut GS' algorithm (or in one case by its variant 'Graph cut GM') for the three considered noise levels and the two datasets. The curves obtained when varying the dataset (see Fig. 5a and 5b) are quite similar (also for not shown noise levels).

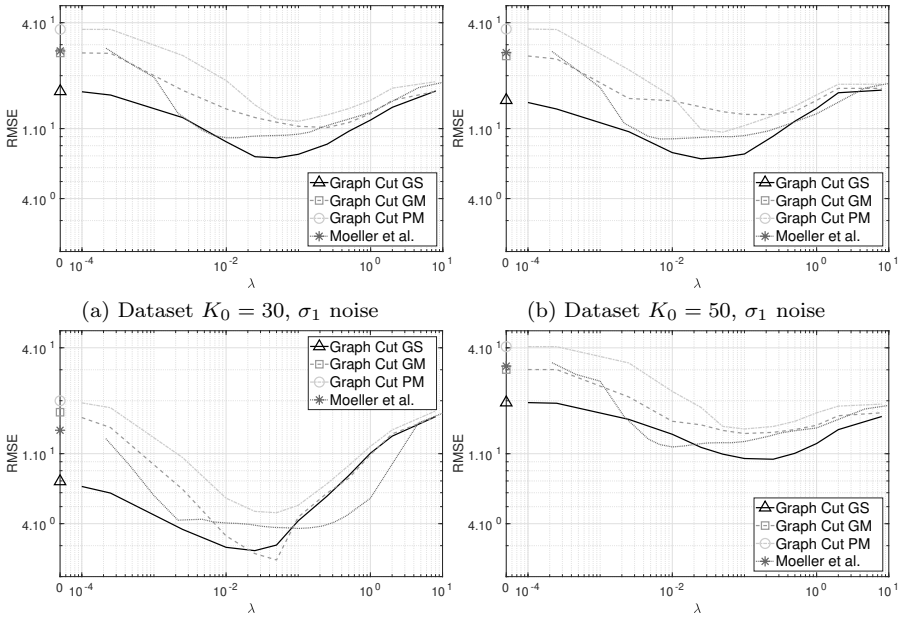
- In the case of Fig. 5c (and visually with the pencils in Fig. 4g), avoiding

the spatial averaging of the SMLAP allows for retrieving thin details in the depth map. However, for noisy data, 'Graph cut GS' outperforms 'Graph cut GM' thanks to noise filter included in SMLAP.

Table 2 shows the values of the four metrics (see Section 4.1), averaged over the two considered datasets with $K_0 = 30$ and $K_0 = 50$ images, for $\lambda^*_{GS} = 0.025$ and $\lambda^*_M = 0.001$. We note that, as seen on Fig. 5, 'Graph Cut GS' achieved smaller RMSE values than [10] and that both mean and standard deviation values increase with noise level. For a given dataset, the histogram of the absolute errors has been computed on all the included images, with ground truth depths scaled in $[0, 100]$. As usual, the noise has a much stronger impact on the 90^{th} percentile values than on the median ones. We notice that the effect of K_0 (number of used images) is more visible on median criterion than on RMSE or 90^{th} ones. Again, according to these two new criteria (ii-iii), our approach outperforms [10]. Finally, considering SSIM, we see that [10] provides slightly better performance than graph cut approach. Indeed, [10] algorithm provides smooth edges and rather homogeneous regions (see Fig. 4e) compared to the obtained results (see Fig. 4f, 4g and 4h). Besides, in [10], handled depths are continuous values whereas our approach considers a finite number of labels.

4.3. Benefit of data-driven decomposition

Let us now investigate the behavior of the proposed data-driven splitting, either according to the median value (of the considered interval) or to Otsu's criterion. For doing so, we mainly compare the results of our different variants, represented in the box named *Splitting Strategy* in Fig. 1, during the first iterations of the algorithms that correspond to small numbers of labels $\tilde{K}$, for which adaptive splitting offer the best appreciation. Figure 6 illustrates some recovered depth maps either after two iterations ($\tilde{K} = 4$, first line) or after three iterations ($\tilde{K} = 8$, three remaining lines). The first column shows the all-in-focus image (*Aloe*, *Flowerpots* and *Moebius* examples of the database); The following columns show the depth maps achieved using the dyadic splitting, the



(a) Dataset $K_0 = 30$, σ_1 noise

(b) Dataset $K_0 = 50$, σ_1 noise

(c) Dataset $K_0 = 30$, σ_0 noise

(d) Dataset $K_0 = 30$, σ_2 noise

Figure 5: RMSE between the recovered depth maps and the ground truths (averaged value on the whole dataset), versus λ (points for $\lambda = 0$ correspond to blind estimation), for the four variants of our approach; Three levels of noise: $\sigma_0 = 0$, $\sigma_1 < \sigma_2$.

median one and Otsu's splitting strategies, respectively. From Fig. 6, we may notice that:

- In the case of the dyadic splitting, there are some unused grey levels (i.e. depth values), e.g. only 3 labels used instead of 4 for *Aloe* at second iteration, or 5 labels actually used instead of 8 for *Flowerpots*. Note that unused labels occur only for specific depth histograms, e.g. empty bins either at the histogram bounds or between main modes.

- Concerning median or Otsu's splitting, the labels distribution follows the histogram features, so that each of the labels (among 4 or 8) represents significant numbers of pixels. In the case of the median strategy, although theoretically the numbers of pixels per label should be equal, practically these numbers only are close due to the regularization and to the quantification from labels discretization.

- The median splitting seems to provide more detailed depth maps in the background whereas the Otsu's splitting gives more details in the foreground. Indeed, Otsu's criterion is sensitive to the global dynamic of the histogram (difference between bounds of the considered depth interval) so that, even a few pixels at interval border can attract the split value. In the presented examples (but the *Flowerpots*), most pixels are located around intermediate depths with few pixels very close to the camera (foreground), hence explaining the different behaviors.

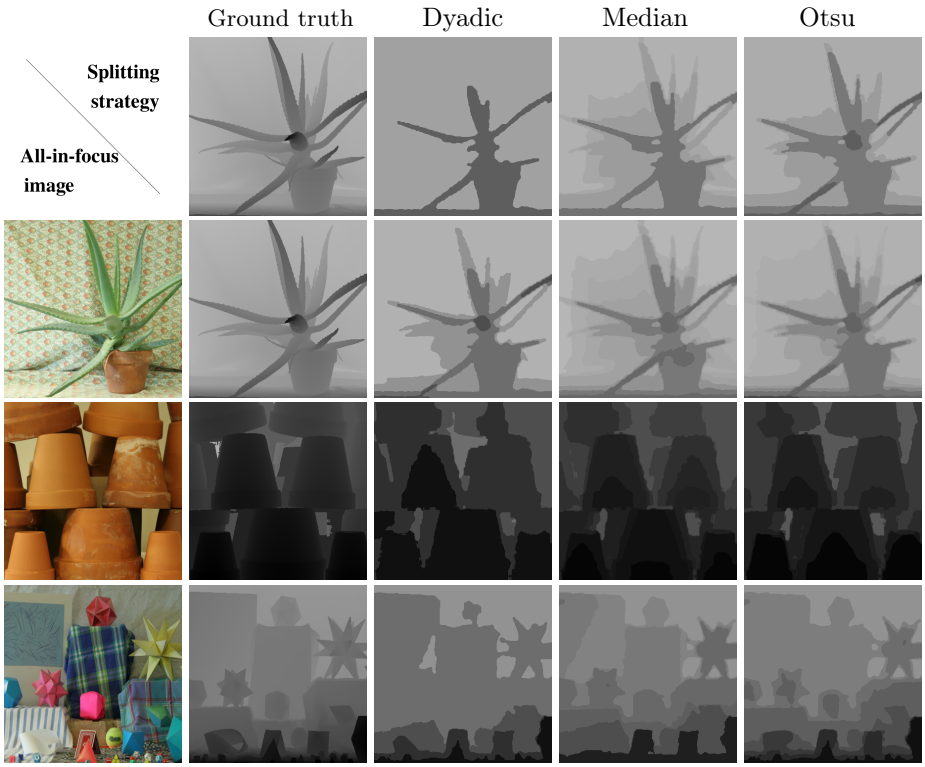


Figure 6: Regularized depth maps obtained after two (top row) or three iterations (three remaining rows) of our approach, each of the three last columns representing a splitting strategy. The second column shows the full resolution ground truth for comparison.

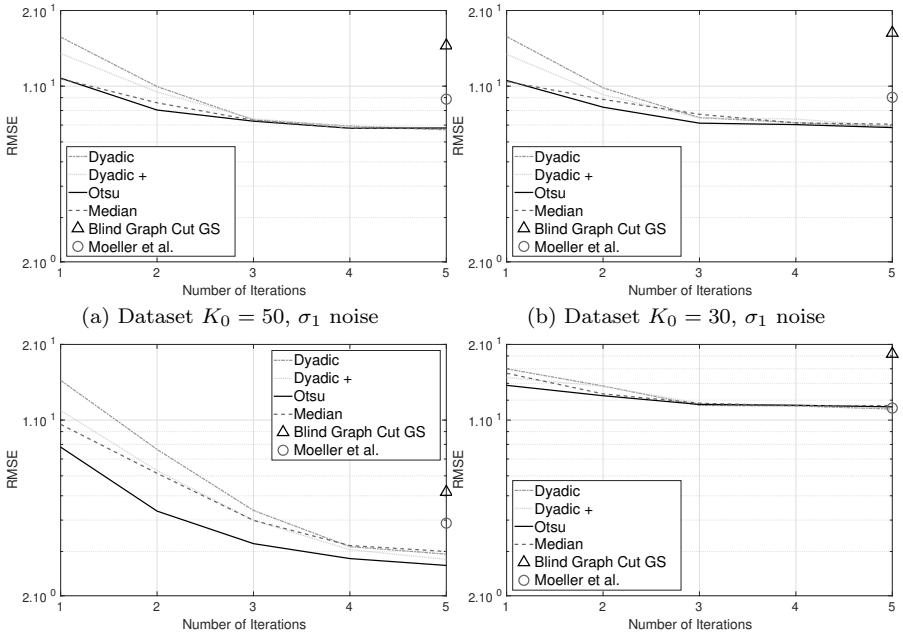
To evaluate the benefit of data-driven splitting, RMSE values (averaged over the considered dataset) are plotted against the number of iterations for three noise levels and two datasets and for all tested splitting strategies in Fig. 7. On

the first line, the noise level is fixed while the considered datasets correspond either to $K_0 = 30$ or $K_0 = 50$. In contrast, the second line shows subgraphs for distinct noise levels and for the dataset with $K_0 = 30$. As expected, when increasing the number of labels, all splitting strategies converge towards depth maps having the same RMSE value and for all splitting strategies, RMSE values grow with the noise level. Additionally, it can be observed that globally (i.e. for a large number of scenes with various depth histograms), the data-driven strategies allow for a lower RMSE for small values of $\tilde{K}$ (small iteration numbers) compared to usual dyadic splitting. On Fig. 7, the curves named Dyadic+ correspond to the dyadic splitting except that the labels (interval centroids represented by crosses on Fig. 3) are estimated as the average values of the depths over the interval (knowing blind histogram) rather than as the interval center. This allows for a decrease of the RMSE values (under dyadic strategy) and a fair comparison with data-driven strategies that also use interval-averaged values for label value estimation. Comparing all strategies for small values of $\tilde{K}$, Otsu's splitting clearly offers the fastest RMSE decrease when noise is absent. In the other cases, all data-driven splitting strategies perform equally well and still outperform dyadic splitting.

5. Conclusion and perspectives

In this paper, we present a new Shape-From-Focus method based on variational formulation using a convex functional. Thanks to its convexity property, the functional can be minimized exactly using graph cuts. More precisely, the multi-label problem is decomposed into a sequence of independent binary subproblems, that can be solved in an efficient way using the graph cut optimization framework. We explore different strategies for decomposition, namely the classic dyadic splitting, and two data-driven strategies, namely either median value splitting or using Otsu's algorithm. Their benefit relies in the reconstruction of the main parts of the scene with a small number of labels.

The proposed data-driven strategies can be applied to several other problems



(a) Dataset $K_0 = 50$, σ_1 noise (b) Dataset $K_0 = 30$, σ_1 noise

(c) Dataset $K_0 = 50$, σ_0 noise (d) Dataset $K_0 = 50$, σ_2 noise

Figure 7: RMSE between the recovered depth maps and the ground truths (average on the whole dataset), versus the number of graph cuts (iteration), for four splitting strategies; Three levels of noise: $\sigma_0 = 0$, $\sigma_1 < \sigma_2$.

designed for the 'divide-and-conquer' approach. In particular, the problems dealing with the estimation of a unknown variable taking values in an ordered set can be formulated in terms of a leveled-energy, for which the data-driven decomposition may be relevant, such as image denoising or half-toning.

References

[1] A. Malik, T.-S. Choi, Comparison of polymers: A new application of shape from focus, IEEE Trans. on Systems, Man, and Cybernetics, Part C 39 (2) (2009) 246–250. doi:10.1109/TSMCC.2008.2001714.

[2] R. A. Hamzah, H. Ibrahim, Literature survey on stereo vision disparity map algorithms, Journal of Sensors 2016 (2) (2016) 1–23. doi:10.1155/2016/8742920.

		RMSE	Median	90^{th} percentile	SSIM
$\lambda = \lambda^*_{GS}$		'Graph cut GS'			
$K_0 = 30$	σ_0	**2.71**	**0.78**	**1.96**	0.26
	σ_1	**5.47**	**1.18**	**9.80**	0.24
	σ_2	**8.51**	**1.57**	**18.0**	**0.22**
$K_0 = 50$	σ_0	**2.46**	**0.39**	**1.57**	0.26
	σ_1	**4.93**	**0.78**	**7.45**	0.24
	σ_2	**7.83**	**0.78**	**15.7**	**0.22**
$\lambda = \lambda^*_M$		Approach [10]			
$K_0 = 30$	σ_0	3.52	1.57	3.92	**0.33**
	σ_1	6.94	1.96	15.7	**0.25**
	σ_2	10.2	2.35	23.5	0.21
$K_0 = 50$	σ_0	3.40	1.57	4.31	**0.33**
	σ_1	7.03	2.35	16.9	**0.26**
	σ_2	9.11	2.75	23.9	0.20

Table 2: Results of the four evaluation metrics, averaged over the two considered datasets with $\lambda^*_{GS} = 0.025$ for 'Graph cut GS' and $\lambda^*_M = 0.001$ for [10]. Best results between the two approaches are shown in bold.

[3] R. Zhang, P.-S. Tsai, J. E. Cryer, M. Shah, Shape-from-shading: A survey, IEEE Trans. on PAMI 21 (8) (1999) 690–706. doi:10.1109/34.784284.

[4] J.-D. Durou, M. Falcone, M. Sagona, Numerical methods for shape-from-shading: A new survey with benchmarks, JCVIU 109 (1) (2008) 22–43. doi:10.1016/j.cviu.2007.09.003.

[5] C. Tomasi, T. Kanade, Shape and motion from image streams under orthography: A factorization method, International J. of Computer Vision 9 (2) (1992) 137–154. doi:10.1007/BF00129684.

[6] S. Nayar, Y. Nakagawa, Shape from focus, IEEE Trans. on PAMI 16 (8) (1994) 824–831. doi:10.1109/34.308479.

[7] G. Surya, M. Subbarao, Depth from defocus by changing camera aperture: A spatial domain approach, in: Proc. of CVPR, 1993, pp. 61–67. doi:10.1109/CVPR.1993.340978.

[8] D. Kumar CH, V. Hi, R. R. Sahay, Shape-from-focus using Total Variation

Prior and Split Bregman algorithm, in: Proc. of ICVGIP, ACM Press, 2014, pp. 1–7. doi:10.1145/2683483.2683563.

[9] P. K. G., R. R. Sahay, Accurate Structure Recovery via Weighted Nuclear Norm: A Low Rank Approach to Shape-from-Focus, in: IEEE International Conf. on Computer Vision Workshop, 2017, pp. 563–574. doi:10.1109/ICCVW.2017.73.

[10] M. Moeller, M. Benning, C. Schonlieb, D. Cremers, Variational Depth From Focus Reconstruction, IEEE Transactions on Image Processing 24 (12) (2015) 5369–5378. doi:10.1109/TIP.2015.2479469.

[11] C. Hazirbas, L. Leal-Taixé, D. Cremers, Deep Depth From Focus, arXiv preprint arXiv:1704.01085.

[12] S. Pertuz, D. Puig, M. A. Garcia, Analysis of focus measure operators for shape-from-focus, Pattern Recognition 46 (5) (2013) 1415–1432. doi:10.1016/j.patcog.2012.11.011.

[13] M. Subbarao, T. Choi, Accurate recovery of three-dimensional shape from image focus, IEEE Trans. on PAMI 17 (3) (1995) 266–274. doi:10.1109/34.368191.

[14] H. Nair, C. Stewart, Robust focus ranging, in: Proceedings of CVPR, 1992, pp. 309–314. doi:10.1109/CVPR.1992.223258.

[15] V. Gaganov, A. Ignatenko, Robust shape from focus via Markov random fields, in: Proceedings of GraphiCon'2009, 2009, pp. 74–80.

[16] Y. Boykov, O. Veksler, R. Zabih, Fast approximate energy minimization via graph cuts, IEEE Transactions on Pattern Analysis and Machine Intelligence 23 (11) (2001) 1222–1239. doi:10.1109/34.969114.

[17] M. T. Mahmood, Shape from focus by total variation, in: IVMSP Workshop, 2013 IEEE 11th, IEEE, 2013, pp. 1–4. doi:10.1109/IVMSPW.2013.6611940.

[18] V. Kolmogorov, R. Zabih, What energy functions can be minimized via graph cuts?, IEEE Trans. on PAMI 26 (2) (2004) 147–159. doi:10.1109/TPAMI.2004.1262177.

[19] Y. Boykov, V. Kolmogorov, An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision, IEEE Trans. on PAMI 26 (9) (2004) 1124–1137. doi:10.1109/TPAMI.2004.60.

[20] H. Ishikawa, Exact optimization for Markov random fields with convex priors, IEEE Trans. on PAMI 25 (10) (2003) 1333–1336. doi:10.1109/TPAMI.2003.1233908.

[21] J. Darbon, M. Sigelle, Image restoration with discrete constrained total variation part I: Fast and exact optimization, J. of Mathematical Imaging and Vision 26 (3) (2006) 261–276. doi:10.1007/s10851-006-8803-0.

[22] D. Goldfarb, W. Yin, Parametric maximum flow algorithms for fast total variation minimization, SIAM J. on Scientific Computing 31 (5) (2009) 3712–3743. doi:10.1137/070706318.

[23] D. Greig, B. T. Porteous, A. Seheult, Exact maximum a posteriori estimation for binary images, J. of the Royal Statistical Society, Series B 51 (2) (1989) 271–279.

[24] N. Otsu, A threshold selection method from gray-level histograms, IEEE Trans. on Systems, Man, and Cybernetics 9 (1) (1979) 62–66.

[25] D. Scharstein, C. Pal, Learning conditional random fields for stereo, in: Proc. of CVPR, IEEE, 2007, pp. 1–8. doi:10.1109/CVPR.2007.383191.

[26] Z. Wang, H. R. Sheikh, Image Quality Assessment: From Error Visibility to Structural Similarity, IEEE Trans. On Image Processing 13 (4) (2004) 600–612. doi:10.1109/TIP.2003.819861.