Efficient Graph Cut Optimization for Shape From Focus

Christophe Ribal, Nicolas Lermé, Sylvie Le Hégarat-Mascle

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 problem of recovering the depth in every point of a scene from a set of differently focused 2D images. Recently, some authors stated this inverse problem in the variational framework and solved it by minimizing a non-convex functional. However, global optimality on the solution is not guaranted and evaluations are either application-specific or incomplete. To overcome these limits, we propose in this paper 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, we investigate several decomposition strategies. Specifically, we focus on data-driven strategies suited to early reconstruction. We provide qualitative and quantitative evaluation on real popular datasets. According to classic statistics on error values, the proposed approach exhibits high performance and robustness against corruped data.

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

*Corresponding author `christophe.ribal@ens-cachan.fr`

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 approaches (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 represent 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 in only a limited range (depth of field) and progressively blurred as we move 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 (note that this paper will focus on the passive device). In addition to the depth map, both approaches also provide an estimation of the all-in-focus (i.e. sharp) image of the scene. Now, about the differences between SFF and SFD, 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. Then, SFF only assumes that there is a direct correspondence between the depth of a given pixel and the focal setting 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 allows for a more compact size of the electronic system, decreases its costs and avoids to deal with matching ambiguities. The cost to pay for such approaches lies however in an increased number of images to acquire and to process.

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 introduces [6] a sharpness operator named Summed Modified LAPlacian (SMLAP) based on second derivatives. We refer the reader to the study [8] that compares a wide variety of sharpness operators in a comprehensive way.

Then, the idea of early approaches (such as [6]) is to compute a sharpness profile over focus values and take the argument of the maximum of this profile for every pixel, independently of its neighbors. However, whatever the used sharpness operator, an estimation using raw profile is prone to errors in presence of corrupted 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, [9] 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 neighbors) therefore do not generally allow for accurately recovering the 3D geometry of a whole scene. Indeed, the sharpness operator relies on objects borders that produce sharp edges on which reliable and precise depth values may be deduced. In the ab-

	Our	[12]	[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 [13]	No guaranty of optimality

Table 1: Functional properties between the proposed approach, [12] and [10].

sence 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 [11] 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-problem. While formulating this kind of problem in the variational framework is a standard way to tackle it, surprisingly, only a very few papers did it [12, 14, 10]. Mathematically, this amounts to the definition of a functional embedding a data fidelity term and a smoothness (or regularization) term 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, the latter 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 (Alternating Direction Method of Multipliers (ADMM)) is applied. According to the authors, this algorithm provably converges to 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

over few real images.

The work of [12] 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 [13]. Interesting results are obtained but the approach is prone to get easily trapped in local minima of the energy and in [12], the evaluation is limited to application-specific images (optical microscopes).

In this work, we explore a new way to solve the SFF problem by directly minimizing a convex functional. The advantage of such choice is twofold: (i) optimality about the solution is easier to guarantee and (ii) the convexity property can be exploited to use fast minimization procedures. For the latter, our choice focused on graph cuts because of their well-founded theoretical background [15] and their efficiency to solve a variety of multi-labels problems with an observed near-linear complexity with the number of pixels using a fast maximum-flow/minimum-cut algorithm [16]. Functional properties of the aforementioned approaches against ours are summarized in the Table 1. While [17] has optimality guarantees for convex priors, the graph construction requires a lot of computational resources (in terms of time and memory), making the approach totally impractical for images with a large number of pixels and/or a large number of focused images. The α-expansion overcomes this limit by iteratively considering and solving independent binary problems over a fixed sequence of labels until convergence. At each iteration, the energy is guaranteed to decrease but the number of iterations is neither fixed nor bounded. The functional can also be exactly minimized without stopping criterion (modulo a quantization step) when the data fidelity term is convex, by mapping the original problem to a fixed number of independent binary problems (each one solved using graph cuts) [18]. In the latter paper, this approach has been successfully applied to

grayscale images restoration but the adaption to color images is not straightforward. The number of binary problems can be drastically diminished using a divide-and-conquer strategy [18]. The idea is to divide a multi-label problem into a set of binary problems. In our case, it boils down to choosing splits along the depth dimension and labeling according to each split. A dyadic dichotomic strategy is an usual way to select these splits [18]. However, a data-driven splitting strategy may be more interesting than the dyadic one when the labels distribution is unbalanced. Indeed, in this case: (i) the size of the subproblems is unbalanced, so that the complexity of a divide-and-conquer process increases and (ii) for the dyadic strategy, the results of intermediate iterates present a higher (than data-driven ones) reconstruction error w.r.t. the all-in-focus image. The latter point is thus of interest when some reconstruction error is targeted by the user.

In what follows, the proposed functional is described in Section 2. In Section 3, we provide the algorithm for exactly minimizing such functional and investigate different splitting strategies. Section 4 analyzes the results obtained on real data and it compares the respective performances achieved by the variants of against [10]. Finally, the contributions of this paper are summarized and future work is discussed in Section 5.

2. Proposed functional

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

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

The approach proposed in this paper 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 radius increasing with the sensor displacement. This phenomenon in the frequency domain acts 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 sharpness operators [8]. For every pixel $p \in \mathcal{P}$ and every index $k \in \mathcal{K}$, this operator is defined as

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

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 intensities of

image I_k at pixel q and Laplacians are approximated by finite differences. Note that MLAP can be deduced for Eq. (1) by restricting $\Omega(p)$ to pixel p. The above operator Φ is used in Section 4.

Once the sharpness operator has been applied to the sequence of focused images $\{I_k\}_{k\in\mathcal{K}}$, resulting measurements are usually filtered. The interest of interpolating sharpness profiles is twofold: (i) it increases the robustness against potential degradations (noise, contrast, etc.), and (ii) it enables us to reduce the discretization step along the depth dimension, when a larger resolution is needed. The interpolation or filtering technique should be carefully selected since the blind estimation relies on it. Polynomial [10] and Gaussian [6] interpolations are two common 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. 1, the raw sharpness profiles as well as those filtered by the aforementioned techniques are drawn on the right for three distinct locations depicted on the left 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 pixel 3), the maxima found are very close to the ideal one and therefore all filtering techniques perform equally 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). Based on these observations, we therefore use the Gaussian filtering in this work.

Let us denote by $\tilde{\Phi}(p,.)$ the interpolated sharpness profile of any pixel $p \in \mathcal{P}$, whatever the sharpness operator Φ used. For some integer $\tilde{K} > 0$, let us also define the sets $\tilde{\mathcal{K}} = \{0,\ldots,\tilde{K}-1\}$ and $\tilde{\mathcal{L}} = \{l_k\}_{k\in\tilde{\mathcal{K}}}$ with $l_k \in \mathbb{R}$, $\forall k \in \tilde{\mathcal{K}}$. Please note that the focus values $\{l_k\}_{k\in\tilde{\mathcal{K}}}$ are not necessary equally spaced along depth dimension: For targeted applications, some preference can be given to some specific range of values. In the following, we only refer to the set $\tilde{\mathcal{L}}$ instead of

$\mathcal{L}$. The blind depth estimates can now be formally defined as

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

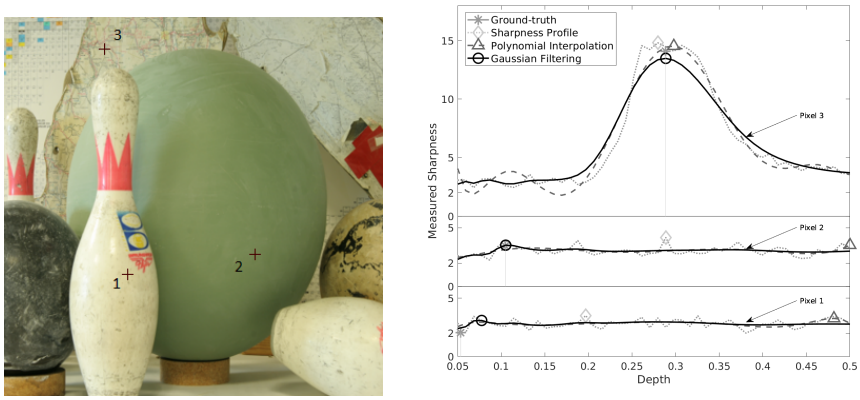


Figure 1: Influence of the filtering techniques on the blind depth estimates for three cross-marked pixels shown on the all-in-focus image (left) and their corresponding sharpness profiles (right). The absence of contrasted texture leads to a poor depth estimate (i.e. too far from the ideal one). The proposed Gaussian filtering offers more accurate results than the polynomial interpolation.

2.2. Functional

Given blind estimates $\mathbf{v} \in \tilde{\mathcal{L}}^{\mathcal{P}}$ obtained using Eq. (2) (see Section 2.1) and for any $\mathbf{x} \in \tilde{\mathcal{L}}^{\mathcal{P}}$, we denote the functional to minimize by

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

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 the sharpness profile, $V_{p,q}(x_p, x_q)$ 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.

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

$$\eta_p \propto \left(\frac{\tilde{K} \left(\tilde{\Phi}(p, v_p) - \tilde{\Phi}(p, z_p) \right)^2}{\sum_{k \in \tilde{\mathcal{L}}} \left(\tilde{\Phi}(p, l_k) - \tilde{\Phi}(p, z_p) \right) + \varepsilon} \right) \in \left[0, \frac{\tilde{K}(\tilde{\Phi}(p, v_p) - \tilde{\Phi}(p, z_p))}{\tilde{K} - 1} \right[,$$

with $\varepsilon \simeq 0$ and z_p is defined for any pixel $p \in \mathcal{P}$ as $z_p = \text{argmin}_{k \in \tilde{\mathcal{L}}} \{\tilde{\Phi}(p, l_k)\}$. In Eq. (4), η_p is independent of x_p and is therefore only computed once. The dynamic range of the sharpness profile $\tilde{\Phi}(p, .)$ is variable across the pixels depending on whether the pixel p lies in a textured region or not. 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 reliability of each pixel p and weights the data fidelity term accordingly.

Now, the regularization term in Eq. (3) corresponds to the anisotropic total variation. 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 [19]). Despite some undesired behaviors of TV such as "staircasing effect" (creation in the depth map of flat regions separated by artifact boundaries), this operator has been well studied in the past, it enjoys desirable properties (convexity, discontinuity-preserving of image boundaries, etc.) and it has been successfully applied to a number of applications and problems such as in image restoration when $\alpha = 1$ or $\alpha = 2$ (see [19] 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 approach [18] for exactly minimizing (modulo a quantization step) convex functionals (like the functional (3) presented in Section 2.2) can be used using graph cuts for solving the SFF problem. 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 (graph cut) and how their number can be drastically reduced using a divide-and-conquer process with a dichotomic splitting. Thirdly, other splitting strategies are investigated.

3.1. Leveled-energy decomposition

In the sequel, we assume that the set $\tilde{\mathcal{L}}$ is ordered and consists of increasing values, i.e. $l_0 < \ldots < l_{\tilde{K}-1}$. As explained in [18], 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 x, with $x \in \tilde{\mathcal{L}}^{\mathcal{P}}$. For doing so, let us denote by $x_p^l = \mathbf{1}_{\{x_p \geq l\}}$ the l-level set of the variable x_p. For any pixel $p \in \mathcal{P}$, the data term can be decomposed as

$$U_p(x_p) = \left(\sum_{k \in \tilde{\mathcal{K}} \setminus \{0\}} x_p^{l_k} \left(U_p(l_k) - U_p(l_{k-1}) \right) \right) + U_p(l_0). \tag{6}$$

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

$$V_{p,q}(x_p, x_q) = \sum_{k \in \tilde{\mathcal{K}} \setminus \{0\}} \underbrace{w_{p,q}(x_p^{l_k} - x_q^{l_k}))^+}_{V_{p,q}(x_p^{l_k}, x_q^{l_k})}. \tag{7}$$

In the latter expression, the sum on k starts from $k = 1$ since $x_p^{l_0} = x_q^{l_0} = 1$, $\forall (p, q) \in \mathcal{N}$. Using Eq. (3), Eq. (6) and Eq. (7), the functional may now be

written

$$E(\mathbf{x}) = \sum_{k \in \tilde{\mathcal{K}} \setminus \{0\}} E^{l_k}(x^{l_k}) + C, \tag{8}$$

where C is a constant that does not depend on $\mathbf{x}$ and the energy E^{l_k} is defined, for any $k \in (\tilde{\mathcal{K}} \setminus \{0\})$ and any binary matrix $\mathbf{w} = \{0,1\}^{\mathcal{P}}$, by

$$E^{l_k}(\mathbf{w}) = \sum_{p \in \mathcal{P}} w_p (U_p(l_k) - U_p(l_{k-1})) + \lambda \sum_{(p,q) \in \mathcal{N}} V_{p,q}(w_p, w_q). \tag{9}$$

For any $k \in \tilde{\mathcal{K}} \setminus \{0\}$, let us denote by $\hat{\mathbf{x}}^{l_k} \in \{0,1\}^{\mathcal{P}}$ a minimizer of E^{l_k}. If these minimizers satisfy

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

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

$$\hat{x}_p = \max \{k \in \tilde{\mathcal{K}} \mid \hat{x}_p^{l_k} = 1\},$$

minimizes Eq. (8). According to [18], 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 $\{E^{l_k}\}_{k \in \tilde{\mathcal{K}} \setminus \{0\}}$. Let us 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 algorithmic developments, graph cuts remained bounded to binary image restoration for a long time [20]. The emergence of a fast maximum-flow/minimum-cut algorithm [16] coupled to a better characterization of what energies can be minimized [15], was a milestone for solving challenging visions tasks such as segmentation, restoration, stereovision, etc. In particular, [15] 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 the functional (3).

For minimizing every E^{l_k} (see Eq. (9)) using graph cuts, we adopt the graph construction detailed in [15]. 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:

$$\left\{ \begin{array}{lcll} c_{s,p} & = & (U_p(l_k) - U_p(l_{k-1}))^-, & \forall p \in \mathcal{P}, \\ c_{p,t} & = & (U_p(l_k) - U_p(l_{k-1}))^+, & \forall p \in \mathcal{P}, \\ c_{p,q} & = & \lambda w_{p,q}, & \forall (p,q) \in \mathcal{N}, \end{array} \right. \tag{11}$$

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

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

For any $\mathcal{S} \subset \mathcal{P}$, we also define for every $p \in \mathcal{P}$

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

There is a one-to-one correspondence between the sets $\mathcal{S}$ and the elements of $\{0,1\}^{\mathcal{P}}$. Using the edge capacities (11) as well as the definitions (9) and (5), it is also straightforward to see that $\mathrm{val}_{\mathcal{G}}(\mathcal{S})$ is equal to $E(\mathbf{x}^{\mathcal{S}})$, up to a constant that is independent of $\mathbf{x}$. If $(\mathcal{S}^* \cup \{s\}, (\mathcal{P} \setminus \mathcal{S}^*) \cup \{t\})$ is a minimum s-t cut (s-t cut of minimum weight) in the graph $\mathcal{G}$, $\mathbf{x}^{\mathcal{S}^*}$ is thus a minimizer of E^{l_k}. This minimizer can be efficiently computed using a maximum-flow/minimum-cut algorithm such as [16]. 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 typical vision tasks.

More generally, the minimization of the functional (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 [18] takes advantage of this property and allows one to drastically decrease the number of s-t minimum cuts until $\lfloor \log_2(\tilde{K}) \rfloor$.

3.3. Data-driven decomposition

The dyadic splitting allows for an optimal decomposition for a given set of labels $\tilde{\mathcal{L}}$, requiring $\lfloor \log_2(\tilde{K}) \rfloor$ different graph cuts instead of the $\tilde{K}$ binary problems. Although dyadic splitting provides a deterministic number of iterations (for a given label resolution), it does not necessarily reach optimal reconstructions for intermediate depth maps. Indeed, intermediate results (derived considering less than $\lfloor \log_2(\tilde{K}) \rfloor$ iterations) may be significantly suboptimal, depending on the actual depth map itself. Then, in this section, we investigate automatic decompositions that are data-driven.

The idea is to base the choice of the split values (used at each iteration) on the depth histogram. Then, instead of thresholds corresponding to dyadic splitting of the whole depth interval, we will derive adaptive values based on the depth histogram. Let us denote by τ_j^i the split values used at iteration i, with i_{end} the final number of iterations, $i \in \{1, \ldots, i_{end} - 1)\}$, $j \in \left\{1, \ldots, 2^{i-1}\right\}$. Note that the label values are 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 $\left\{\tau_j^k + \frac{1}{2^i}, \forall 0 \leq k < i\right\}$. Focusing on a data-driven approach, the most prevalent depths should 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 chosen thresholds correspond to 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 chosen split values correspond to Otsu algorithm [21] 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.

Now, five comments have to be made.

- Firstly, the way to choose the split value do not impact the graph cut minimization algorithm presented in Section 3.2.

- Secondly, since the actual depth histogram is unknown, we use the blind depth map to derive an approximation 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 best results in terms of RMSE (Root Mean Square Error). However, both due to discretization of the labels, iterative process and regularization step, achieved RMSE value 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 RMSE, rather than the interval centers, we propose to consider the interval centroids (i.e. mean) values.

- Fifthly, in the absence of a stopping criterion based on the number of pixels per interval, the number of iterations of the adaptive approaches may be greater than $\lfloor \log_2(\tilde{K}) \rfloor$ since the length of intervals is unbalanced so that some 'small' intervals cannot be further split and some 'large' intervals will require more iterations to reach the chosen depth resolution. Therefore,

adaptive splitting is mainly relevant for low numbers of iterations, while the 'cost' of being optimal for the first iterations is either a stopping test (interval occupancy) or a higher total number of iterations.

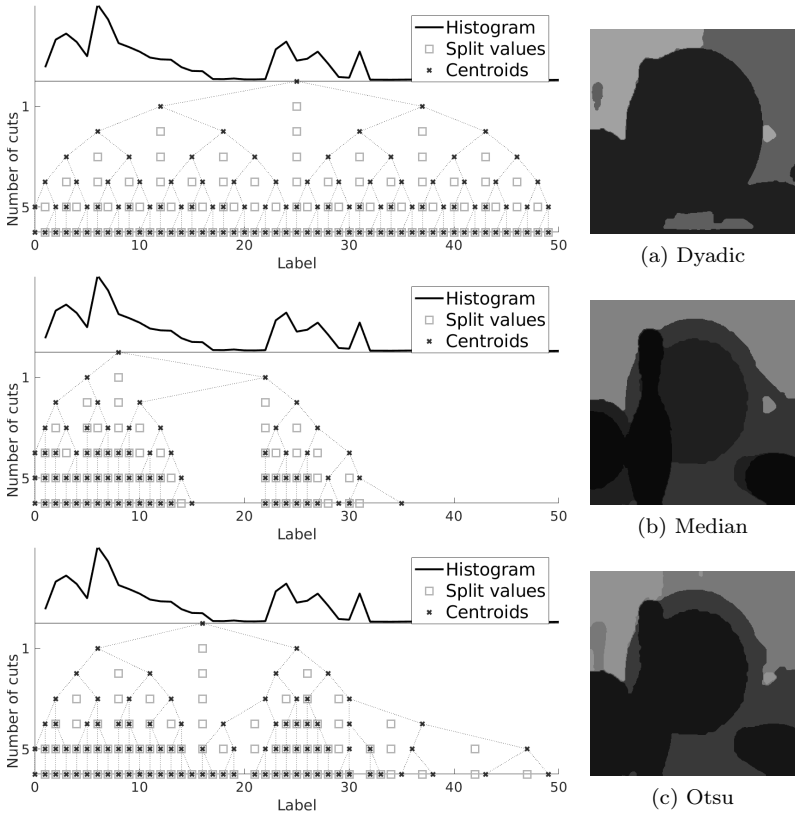


Figure 2: Illustration of the different splitting strategies, for $\lambda = 0.025$; 1^{st} column: tree of the centroids, also showing the split values versus the iteration number, 2^{nd} column: 4-valued depth map achieved at iteration 2. Median or Otsu's splitting strategies outperform the dyadic one in the earliest iterations.

Figure 2 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. 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 in the earliest iterations.

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 [22]. 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 and the lowest illumination and smallest image resolution, for both views 1 and 5. Since the depth maps are recovered from stereo matching, depth information is not available for some pixels due to occlusions. To overcome this problem, the unknown depth values have been estimated by the median value of the surrounding depths.

From this patched dataset, we generate the sequence of focused images for each scene using the code provided by Pertuz [1] with default values of parameters. Each pixel of the all-in-focus image is blurred depending on the distance between the depth map and the image focal plane, thus giving rise to a sequence of K images. 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 $K = 30$ and $K = 50$ 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 $[0, 1]$. Please note that the noise images are uncorrelated along the depth dimension.

To evaluate the performance of our SFF algorithm, we propose to estimate quantitatively the accurateness of our estimation $\mathbf{x} \in \tilde{\mathcal{L}}^{\mathcal{P}}$ against the ground truth $\mathbf{y} \in \tilde{\mathcal{L}}^{\mathcal{P}}$ 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 Universal Quality Index (UQI, [23]). To remove the dependency

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

on the dynamic of the scene (denoted by $\Delta_{\mathbf{x}}$), we scale RMSE values by $\Delta_{\mathbf{x}}$:

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

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

Considering 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, UQI is also complementary since it evaluates the correlation between estimation and ground truth (ideal estimation). It is defined by:

$$UQI(\mathbf{x},\mathbf{y}) = \frac{1}{\sharp\mathcal{P}'}\sum_{p\in\mathcal{P}'}\frac{4\sigma_{x_{\Omega(p)}y_{\Omega}(p)}\bar{x}_{\Omega(p)}\bar{y}_{\Omega(p)}}{\left(\sigma^2_{x_{\Omega(p)}}+\sigma^2_{y_{\Omega(p)}}\right)\left(\bar{x}^2_{\Omega(p)}+\bar{y}^2_{\Omega(p)}\right)} \in [-1,1]\,,$$

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 of $x_{\Omega(p)}$, $y_{\Omega(p)}$ respectively, and $\sigma_{x_{\Omega(p)}}$, $\sigma_{y_{\Omega(p)}}$, and $\sigma_{x_{\Omega(p)}y_{\Omega}(p)}$ are the variances and covariance. For metric (iv), the higher the achieved values, the better the results are.

In what follows, we first illustrate the benefit of the proposed energy model against the state-of-the-art methods. Then, we present the interest of data-driven splittings when seeking for early reconstructions.

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] represents the state-of-art (of variational SFF methods), we compare it to three variants of our method. Faced to the number of algorithm parameters to tune, we run [10] algorithm with its default value parameters (except λ_M parameter

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

that was fitted, see Fig. 4), 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 and $\tilde{K} = K$. 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. 3d). Based on this observation, we therefore have chosen to set $\alpha = 1$ in Eq. (4).

The three variants 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 (see 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 by the relation $\sigma(k) = 0.2k + 1$, 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 where the RMSE values are then sampled. This allows us to observe the behavior of the algorithm versus the regularization parameter as well as to get the λ value achieving the minimal RMSE value (denoted by λ^*) in this sampled interval.

Figure 3 gives a qualitative comparison of some results obtained in the case of the *Art* image example. Specifically, the first line shows the all-in-focus image (last column), the depth ground truth (first column) and 'optimal' result of [10] (middle column). The three following lines allows us to compare the 'Graph cut PM', 'Graph cut GM' and 'Graph cut GS' results with the blind estimations shown on first column, the λ^*-regularized results shown on second column and examples of over-regularized results on last column. We can draw the following comments:

- The benefit of the proposed data fidelity term (SMLAP, Gaussian) is vis-

ible when comparing Fig. 3d, Fig. 3g and Fig. 3j: 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. 3e, Fig. 3h and Fig. 3k 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, 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. 4, 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. 4, the subgraphs correspond to the two considered datasets with either $K = 30$ or $K = 50$ images with a given noise level (intermediate) whereas on the second line the noise level varies for a given dataset ($K = 30$). 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. 4, we observe that:

- The RMSE curves are consistent with qualitative results depicted in Fig. 3: 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 (by removing relevant thin structures of the scene).

- The minimal average RMSE value depends on the noise level (see Fig. 4a,

(a) Ground truth (b) Approach [10] (c) All-in-focus

(d) MLAP, polynomial (e) λ^*-regularized (f) Over-regularized

(g) MLAP, Gaussian (h) λ^*-regularized (i) Over-regularized

(j) SMLAP, Gaussian (k) λ^*-regularized (l) Over-regularized

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

4c and 4d).

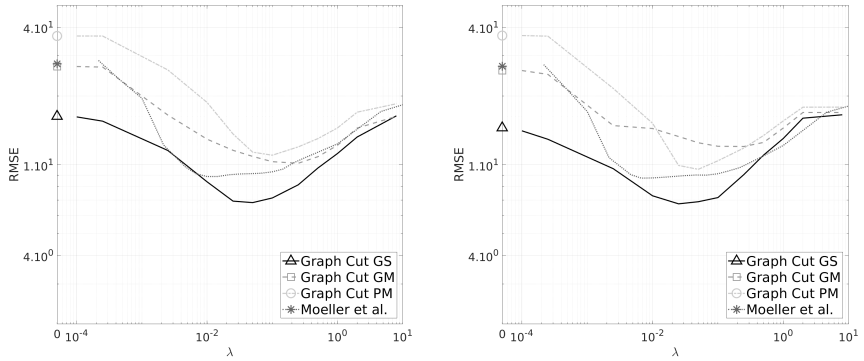
- In terms of averaged RMSE, the best results are achieved by the proposed 'Graph cut GS' algorithm (or in one case by its variant 'Graph cut GM') for the three considered noise levels and the two datasets. The curves obtained when varying the dataset (see Fig. 4a and 4b) are quite similar (also for not shown noise levels).

- In the case of Fig. 4c (and visually with the pencils in Fig. 3h), 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 $K = 30$ and $K = 50$ images, for $\lambda^*_{GS} = 0.025$ and $\lambda^*_M = 0.001$. We note that, as seen on Fig. 4, 'Graph Cut GS' achieved lower 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 have 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 (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 UQI, we see that [10] provides slightly better performance than graph cut approach. Indeed, [10] algorithm provides smooth edges and rather homogeneous regions (see Fig. 3b) compared to the obtained results (see Fig. 3e, 3h and 3k). 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 obtained during the first



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

(c) Dataset $K = 30$, σ_0 noise (d) Dataset $K = 30$, σ_2 noise

Figure 4: RMSE between the recovered depth maps and the ground truths (average 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: $\sigma_0 = 0$, $\sigma_1 < \sigma_2$. In every case, as expected, the curves vary smoothly with a minimum RMSE value achieved for some value of regularization parameter λ. Proposed approach 'Graph Cut GS' outperforms alternative methods.

iterations of the algorithms (since at the end, whatever the splitting criterion, the same full-resolution depth map will be achieved). Figure 5 illustrates some recovered depth maps either after two iterations (first line) or after three iterations (three remaining lines). The first column shows the all-in-focus image (*Aloe*, *Flowerpots* and *Moebius* examples of the database); the following columns shows the depth maps achieved using the dyadic splitting, the median one and Otsu's splitting strategies, respectively. From Fig. 5, we may notice that:

- In the case of the dyadic splitting, there are some unused grey levels (i.e.

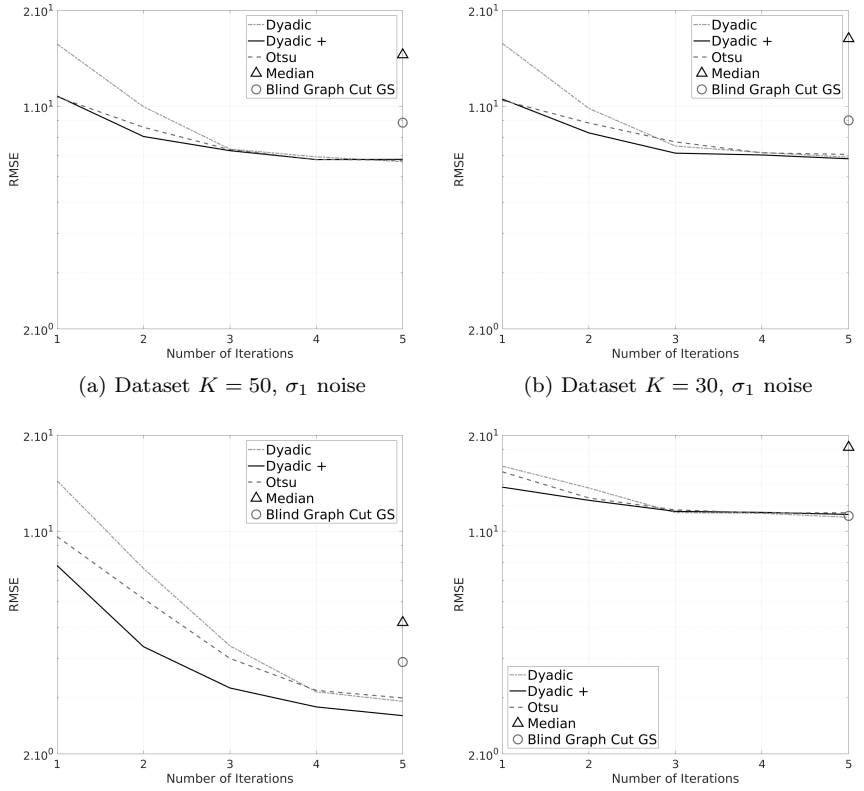
depth values), e.g. only 3 labels used instead of 4 for *Aloe* at second iteration, or 5 labels actually used instead of 8 for *Flowerpots*. Note that unused labels occur only for specific depth histograms, e.g. empty bins either at the histogram bounds or between main modes.

- Concerning median or Otsu's splitting, the labels distribution follows the histogram features, so that each of the labels (either 4 or 8 depending on the iteration) represent 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 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 depth bounds of the considered 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 those behaviors.

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 driven/non-driven splitting strategies in Fig. 6. On the first line, the noise level is fixed while the considered datasets are either for $K = 30$ or $K = 50$. Conversely, the second line shows subgraphs for distinct noise levels and for the dataset with $K = 30$. As expected, 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 allows for a smaller RMSE for small iteration numbers compared to usual dyadic splitting. On Fig. 6, the curves named Dyadic+ correspond to the dyadic splitting followed except that

the labels (interval centroids represented by crosses on Fig. 2) 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 intermediate iterations, 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 outperforms dyadic splitting.



(a) Dataset $K = 50$, σ_1 noise

(b) Dataset $K = 30$, σ_1 noise

(c) Dataset $K = 50$, σ_0 noise

(d) Dataset $K = 50$, σ_2 noise

Figure 6: 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: $\sigma_0 = 0$, $\sigma_1 < \sigma_2$. There is no significant influence of the number of images, whereas the presence of noise induces higher RMSEs. Whatever the noise level, dyadic+ strategy offers intermediate performances among the other splitting strategies. For small iteration numbers, median and Otsu's splitting strategies exhibit the best results, with a clear advantage for the latter when the level of simulated noise is null.

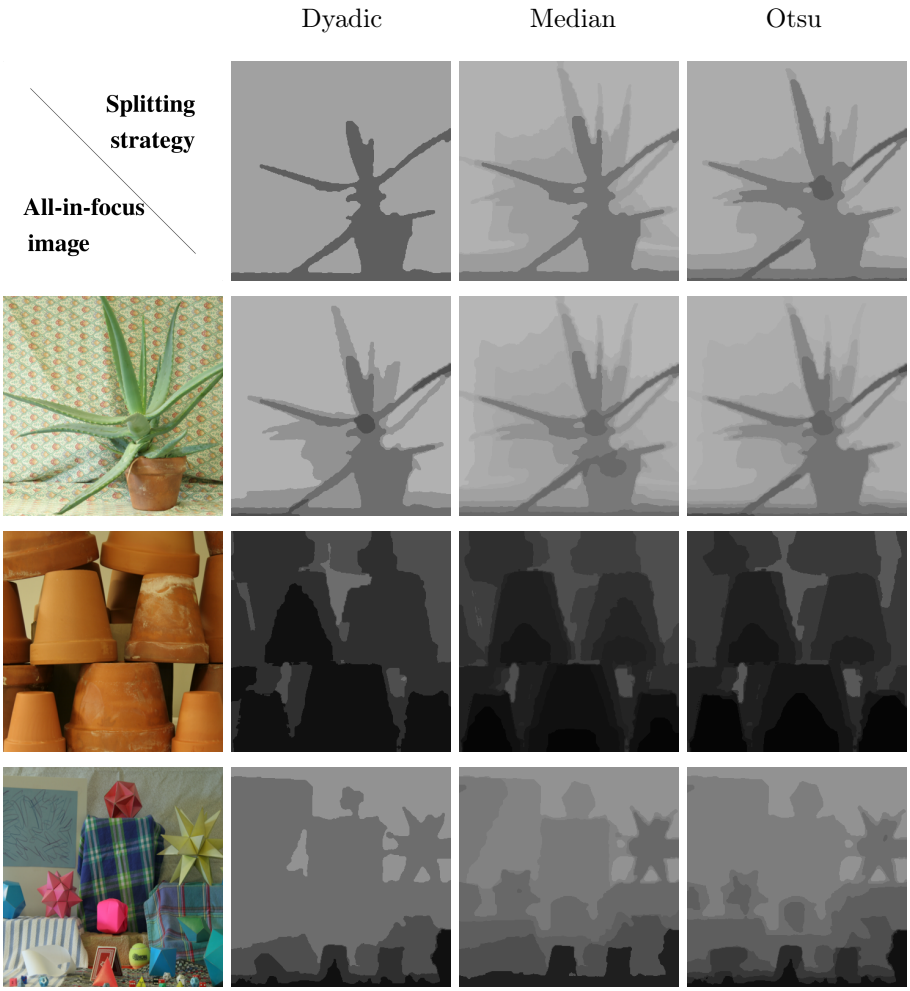


Figure 5: Regularized depth maps obtained after two (top row) or three iterations (three remaining rows) of our approach, each column representing a splitting strategy. It illustrates how the data-driven splitting strategies allow for the maximization of the information recovered (at each step) and how the obtained depth maps focus on different image parts depending on the considered strategy.

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-labels 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 either median value splitting or using Otsu's algorithm. Their benefit relies in the early reconstruction of the main parts of the scene.

The proposed data-driven strategies can be applied to several other problems 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 formulated in terms of a leveled-energy, for which the data-driven decomposition may be relevant, such as image denoising or half-toning.

		RMSE	Median	90^{th} percentile	UQI
$\lambda = \lambda^*_{GS}$		'Graph cut GS'			
$K = 30$	σ_0	**2.71**	**0.78**	**1.96**	0.27
	σ_1	**5.47**	**1.18**	**9.80**	0.22
	σ_2	**8.51**	**1.57**	**18.0**	0.17
$K = 50$	σ_0	**2.46**	**0.39**	**1.57**	0.32
	σ_1	**4.93**	**0.78**	**7.45**	0.28
	σ_2	**7.83**	**0.78**	**15.7**	0.22
$\lambda = \lambda^*_M$		Approach [10]			
$K = 30$	σ_0	3.52	1.57	3.92	**0.37**
	σ_1	6.94	1.96	15.7	**0.29**
	σ_2	10.2	2.35	23.5	**0.21**
$K = 50$	σ_0	3.40	1.57	4.31	**0.37**
	σ_1	7.03	2.35	16.9	0.28
	σ_2	9.11	2.75	23.9	0.22

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. The proposed one behaves better for three metrics out of four.

References

[1] A. Malik, T.-S. Choi, Comparison of polymers: A new application of shape from focus, IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews) 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.

[3] R. Zhang, P.-S. Tsai, J. E. Cryer, M. Shah, Shape-from-shading: A survey, IEEE Transactions on Pattern Analysis and Machine Intelligence 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, Journal of Computer Vision and Image Understanding 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 Journal of Computer Vision 9 (2) (1992) 137–154. doi:10.1007/BF00129684.

[6] S. Nayar, Y. Nakagawa, Shape from focus, IEEE Transactions on Pattern Analysis and Machine Intelligence 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: Proceedings of CVPR, 1993, pp. 61–67. doi:10.1109/CVPR.1993.340978.

[8] 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.

[9] M. Subbarao, T. Choi, Accurate recovery of three-dimensional shape from image focus, IEEE Transactions on Pattern Analysis and Machine Intelligence 17 (3) (1995) 266–274. doi:10.1109/34.368191.

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

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

[13] 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.

[14] 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.

[15] V. Kolmogorov, R. Zabih, What energy functions can be minimized via graph cuts?, IEEE Transactions on Pattern Analysis and Machine Intelligence 26 (2) (2004) 147–159. doi:10.1109/TPAMI.2004.1262177.

[16] Y. Boykov, V. Kolmogorov, An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision, IEEE Transactions on Pattern Analysis and Machine Intelligence 26 (9) (2004) 1124–1137. doi:10.1109/TPAMI.2004.60.

[17] H. Ishikawa, Exact optimization for markov random fields with convex priors, IEEE Transactions on Pattern Analysis and Machine Intelligence 25 (10) (2003) 1333–1336. doi:10.1109/TPAMI.2003.1233908.

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

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

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

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

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

[23] Z. Wang, A. Bovik, A universal image quality index, IEEE Signal Processing Letters 9 (3) (2002) 81–84. doi:10.1109/97.995823.