Dense Labeling with User Interaction: an Example for Depth-Of-Field Simulation

Ana B. Cambra
acambra@unizar.es
Adolfo Muñoz
adolfo@unizar.es
José J. Guerrero
jguerrer@unizar.es
Ana C. Murillo
acm@unizar.es

Instituto de Investigación en Ingeniería de Aragón
Universidad de Zaragoza
Zaragoza, Spain

Introduction. The main contribution of this work is a novel pipeline for interactive dense labeling, which provides a framework that can be applied in any application that involves dense labeling and user interaction. Our approach is focused on efficient dense labeling estimation and is particularly well suited for the use of continuous magnitudes. The dense labeling is formulated as a linear system of equations over superpixels and then solved as a linear least squares problem. Our experiments show that our approach is the fastest to obtain a solution compared to related approaches while keeping comparable quality in the results. Besides, we demonstrate how our pipeline is suitable for interactive applications developing an interactive application for depth-of-field simulated effects from a single image which requires a fast dense depth estimation.

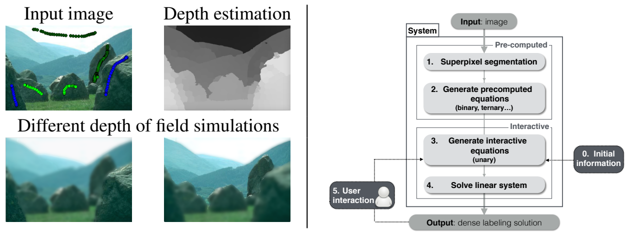
Input image
Depth estimation
Different depth of field simulations
Input: image
System
Pre-computed
1. Superpixel segmentation
2. Generate precomputed equations (binary, ternary...)
Interactive
3. Generate interactive equations (unary)
4. Solve linear system
0. Initial information
5. User interaction
Output: dense labeling solution

Figure 1: Left: Steps of the Interactive D-o-F application: the user provides a few strokes over the image and with each new edition, the application re-estimates the depth map estimation and applies a depth-of-field effect. Right: Steps of our dense-labeling pipeline.

Dense labeling algorithm. The first stage of our pipeline is to segment the target image into superpixels using SLIC [1]. We model the image dense labeling problem as a graph, where the nodes N are the superpixels and the edges E represent the established relationships between superpixels. Given a set of labels L, a dense-labeling problem consists in assigning a label, $l \in L$, to each superpixel in the image. Given this representation, we define a linear system of equations, where the unknowns are the labels. We consider two types of equations: *Unary equations* assign to the unknown label a numerical value based on the individual superpixel properties and/or on the user input; *Binary equations* establish relations between labels of two connected superpixels. The over-determined system obtained can be solved with a common least squares method to find an approximate solution minimizing the error.

Interactive depth of field simulation. Users only need to mark a few depth values in the image, from which our application estimates a dense depth map. Figure shows an example of the application when the user is drawing a few strokes, representing different distances to the camera. The stroke to mark nearby objects is painted in blue and the rest of depths are painted in different ranges of green (the darker, the further). Each pixel affected by a user stroke generates a unary equation that is added to the system (step 3). This, combined with the pre-computed binary equations (step 2), leads to a system that is interactively solved and yields a dense estimated depth map and the corresponding effect: we generate the simulated image by applying a variable Gaussian blur filter. The amount of blur applied increase proportionally with the distance of each pixel to the focal plane and the user can interactively select other focus points.

Experiments. We focus on the part of the pipeline that estimates the labeling (step 4), since it is the only common part in all the approaches compared. We compare our results against state-of-art dense labeling approaches using two very different inputs: a dense but unreliable input labeling obtained automatically from two stereo images and a sparse but more reliable input labeling obtained from a few user strokes. We compare our algorithm to the state-of-the-art algorithms described in a well known comparative of MRF-based dense labeling approaches [7]. We include the recently presented block coordinate descent algorithm (*BCD*) [4], which was developed with an emphasis on speed, and a Random Walk based approach, the implementation provided by [5], which was designed as an interactive segmentation tool.

Table 1: Execution time (*seconds*) and mean error (*err*) for dense labeling obtained for 3 test images with different inputs. (b) Many of the studied approaches are not designed to handle such a sparse initialization. They have been removed from de table because they did not provide a solution.

(a) AUTOMATIC DENSE INPUT

Method	**Tsukuba** (384x288) #Labels=16		**Venus** (434x383) #Labels=20		**Teddy** (450x375) #Labels=60	
	time	**err**	**time**	**err**	**time**	**err**
Pixel-based						
ICM [2]	0.520	0.12	0.460	0.10	1.900	0.13
Expansion [3]	2.220	0.02	6.940	0.02	19.90	0.05
Swap [3]	2.250	0.02	7.010	0.02	12.60	0.05
TRW-S [6]	8.840	0.02	115.0	0.02	158.0	0.05
BP-S [8]	1.370	0.02	8.690	0.03	21.20	0.05
BP-M [8]	13.30	0.02	–	–	193.0	0.05
BCD [4]	0.920	0.09	1.500	0.17	2.760	0.08
RW [5]	0.200*	0.12	0.400*	0.20	0.600*	0.16
Superpixel-based						
Expansion	3.090	0.06	6.320	0.10	6.210	0.08
Ours	**0.002**	0.06	**0.005**	0.10	**0.005**	0.09

(b) SPARSE USER INPUT

Method	**Tsukuba** (384x288) #Labels=5		**Venus** (434x383) #Labels=5		**Teddy** (450x375) #Labels=5	
	time	**err**	**time**	**err**	**time**	**err**
Pixel-based						
BP-M [8]	24.40	0.14	34.20	0.09	35.10	0.18
BCD [4]	–	–	–	–	–	–
RW [5]	0.500*	0.13	0.600*	0.20	0.700*	0.09
Superpixel-based						
Expansion	4.170	0.06	6.370	0.14	7.960	0.09
Ours	**0.002**	0.06	**0.005**	0.15	**0.005**	0.06

–: can't provide to available implementation.
*: execution time is measured in Matlab.

Conclusions. The proposed dense labeling technique has great flexibility to model this problem and has the advantage of providing an interactive solver. We have shown that we yield results which are accurate enough for many applications and often comparable to other slower state of the art methods. Besides, since we target an interactive technique the user can always refine and improve the input iteratively. We believe that our approach will inspire future research for interactive editing applications based on dense labeling.

[1] Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurelien Lucchi, Pascal Fua, and Sabine Susstrunk. Slic superpixels compared to state-of-the-art superpixel methods. *IEEE Trans. Pattern Anal. Mach. Intell.*, 34(11):2274–2282, 2012.

[2] Julian Besag. On the statistical analysis of dirty pictures. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 259–302, 1986.

[3] Yuri Boykov, Olga Veksler, and Ramin Zabih. Fast approximate energy minimization via graph cuts. *IEEE Trans. Pattern Anal. Mach. Intell.*, 23(11): 1222–1239, 2001.

[4] Qifeng Chen and Vladlen Koltun. Fast mrf optimization with application to depth reconstruction. In *CVPR*, pages 3914–3921, 2014.

[5] Leo Grady. Random walks for image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 28(11):1768–1783, 2006.

[6] Vladimir Kolmogorov. Convergent tree-reweighted message passing for energy minimization. *IEEE Trans. Pattern Anal. Mach. Intell.*, 28(10):1568–1583, 2006.

[7] Richard Szeliski, Ramin Zabih, Daniel Scharstein, Olga Veksler, Vladimir Kolmogorov, Aseem Agarwala, Marshall Tappen, and Carsten Rother. A comparative study of energy minimization methods for markov random fields with smoothness-based priors. *IEEE Trans. Pattern Anal. Mach. Intell.*, 30(6): 1068–1080, 2008.

[8] Marshall F Tappen and William T Freeman. Comparison of graph cuts with belief propagation for stereo, using identical mrf parameters. In *ICCV*, pages 900–906. IEEE, 2003.