CEIG - Spanish Computer Graphics Conference (2014) , pp. 1–9
Pere-Pau Vázquez and Adolfo Muñoz (Editors)

Improving depth estimation using superpixels

Ana B. Cambra[1], Adolfo Muñoz[1], Ana C. Murillo[1], José J. Guerrero[1] and Diego Gutierrez[1]

[1]Instituto de Investigación en Ingeniería de Aragón I3A, Universidad de Zaragoza, Spain

Input Image **(a)** | Input depth **(b)** | Superpixels **(c)** | Initial superpixel depth **(d)** | Depth propagation **(e)**

Figure 1: *Given an **(a)** input image and **(b)** its corresponding depth or equivalent (it could come from a RGB-d depth map or be estimated by any standard 3D reconstruction algorithm, where darker color means closer distances), our work is focused on improving this input depth result. We combine the **(c)** superpixel segmentation with the input depth to obtain our **(d)** initial superpixel depth. We use a Markov Random Field to optimize the superpixel depth values assigned to the whole image. In **(e)** we can see how we achieve significant improvements with regard to the input depth.*

Abstract

This work is focused on assigning a depth label to each pixel in the image. We consider off-the-shelf algorithms that provide depth information from multiple views or depth information directly obtained from RGB-d sensors. Both of them are common scenarios of a well studied problem where many times we get incomplete depth information. Then, user interaction becomes necessary to finish, improve or correct the solution for certain applications where accurate and dense depth information for all pixels in the image is needed. This work presents our approach to improve the depth assigned to each pixel in an automated manner. Our proposed pipeline combines state-of-the art methods for image superpixel segmentation and energy minimization. Superpixel segmentation reduces complexity and provides more robustness to the labeling decisions. We study how to propagate the depth information to incomplete or inconsistent regions of the image using a Markov Random Field (MRF) energy minimization framework. We propose and evaluate an energy function and validate it together with the designed pipeline. We present a quantitative evaluation of our approach with different variations to show the improvements we can obtain. This is done using a publicly available stereo dataset that provides ground truth information. We show additional qualitatively results, with other tests cases and scenarios using different input depth information, where we also obtain significant improvements on the depth estimation compared to the initial one.

Categories and Subject Descriptors (according to ACM CCS): I.4.6 [Image Processing and Computer Vision]: Segmentation—Pixel classification

1. Introduction

One very challenging and exciting area in computer vision is 3D reconstruction from a set of images, since it presents plenty of industrial applications in diverse areas such as navigation, archaeology, augmented reality... As such, it has drawn the attention of researchers world wide, which have proposed a set of solutions, each of one finding their specific tradeoff between cost, accuracy, restrictions, user interaction and estimation time. It is a well studied problem, and there are several available tools that lead to a reasonable solution.

© The Eurographics Association 2014.

Specific applications (such as reilumination, augmented reality or image navigation) require an image as input (a view of the scene) and its per-pixel depth. State-of-the-art reconstruction algorithms usually do not provide such a dense depth information for a specific view: regions with no significant features or areas with unstructured high frequency details are very ill-conditioned for such algorithms and lead to incomplete or noisy depth maps that are unusable. Even directly using an RGB-depth sensor (such as Kinect) the resolution and range of the provided depth map can be very low.

In this work, we tackle the problem of, given an incomplete and potentially inaccurate depth estimation and the corresponding image for the same view, completing and improving the depth map. Our algorithm is based on reasonable heuristics related to both geometrical features and image properties, and provide plausible and dense depth maps that can be used in a wide range of applications.

We present an approach to improve the depth estimation of a certain scene by combining any kind of rough initial estimation with a pipeline for pixel-wise labeling optimization. This pipeline makes use of superpixel image segmentation and Markov-Random-Field solvers, both of them very powerful tools frequently used to obtain a robust and consistent labeling in an image. Figure 1 presents a summary of the main steps of this process. Given an input image and an input depth estimated for that view, the steps we perform are the following:

1. *Superpixel segmentation.* This step groups similar image pixels to avoid discontinuities in the results from following steps.
2. *Initial superpixel depth.* As a second step, we obtain a rough depth estimation (or equivalent) with any available method, which typically will not provide a dense depth map, and combined it and the superpixel segmentation to obtain a initial depth labeling.
3. *Depth propagation* through the graph of connected superpixels. We model how the superpixels in the image are related and connected with a Markov Random Field. We use this framework to propagate the depth information across the whole image and improve the initial solution.

Besides detailing these steps, in this paper we analyze and propose different modifications on this pipeline, and we evaluate the improvements achieved in depth estimation using a public dataset with depth ground truth information (consisting of stereo pair images and disparity maps). We also show how this pipeline could improve the depth information obtained from other sources, such as 3D reconstruction from multiple views [FP10] or 3D information directly obtained from RGB-d sensors.

2. Related Work

Markov Random Fields: Many problems in computer vision and scene understanding can be formulated in terms of finding the most probable value for a set of variables, which encode certain property of the scene. This labeling problem is often formulated by means of a Markov Random Field (MRF). In [SZS*08], the authors compare different algorithms to solve MRF optimization problems and show the results obtained applying them to several computer vision tasks such as stereo, image stitching, interactive segmentation, and de-noising image pixels. This kind of labeling has been frequently used to assign a label to each pixel in an image [MAJ11], but lately we find more and more excellent proposals which actually assign a label per pixel group or superpixel instead of modeling each pixel individually [XQ09] [TL10] [SBS12].

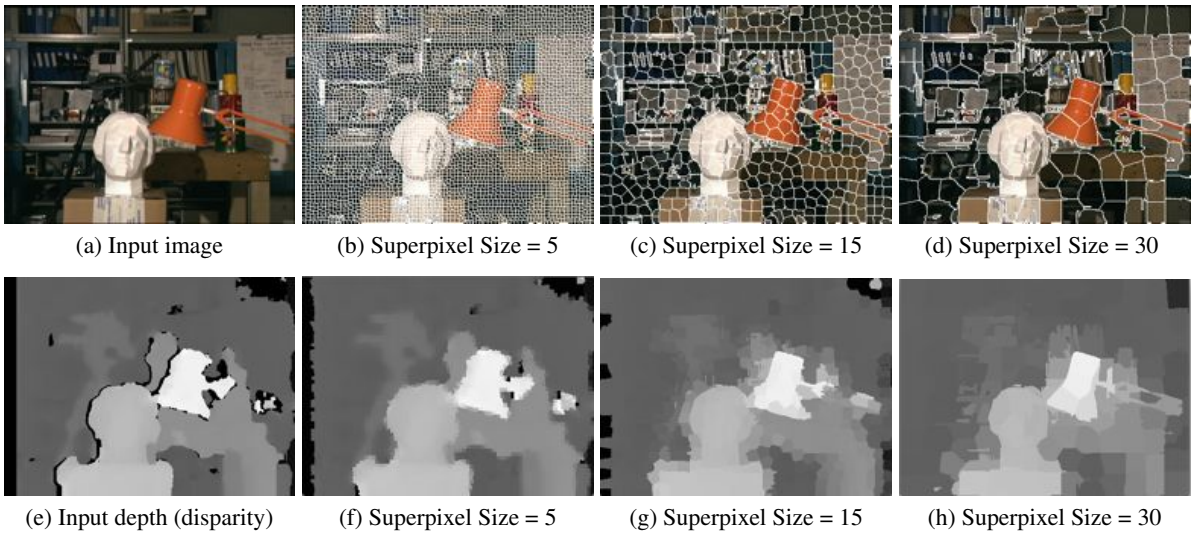
Superpixel segmentation: Superpixel segmentation is becoming increasingly popular as the initial pre-processing step in many computer vision applications, since it allows to make computations and decisions per superpixel instead of per pixel. This provides a more robust and efficient setting and has been shown to be very useful to combine image segmentation and object recognition [FVS09], intrinsic image decomposition [GMLMG12], to improve depth maps obtained from RGB-d cameras [VdBCVG13], depth estimation in a single image [LGK10], or 3D reconstruction results [MK09, CDSHD13]. There has been a lot of research on superpixel image segmentation since the term was established in [RM03].

They can be divided in two families: in the first one, the detection is based on graphs connecting image pixels and gradually adding cuts in this graph for example applying Normalizad Cuts [SM00], such as one of the early superpixel extraction methods presented by Fezenszwalb and Huttenlocher [FH04]; in the second group, the approaches gradually grow superpixels starting from an initial set of candidates, such as the SLIC superpixel detection method [ASS*10], or the more recent approach for SEEDS superpixel detection [VdBBR*12], which proposes a way to deform the boundaries from an initial superpixel partitioning. The different approaches have recently been compared [ASS*12] and although the SEEDS was presented to be faster than SLIC, we use SLIC because it provided a more homogeneous segmentation in our initial experiments. Furthermore, we are not focused on real time applications.

3D Reconstruction: In relation to our goal of improving the depth estimation assigned to each pixel in the image, we find plenty of state-of-the-art implementations of 3D reconstruction from multiple views [FP10, Wu13] or commercial software, e.g., Agisoft PhotoScan† and plenty of sensors are available in the market that provide RGB-depth information (such as Kinect). However, these approaches still need human interaction or additional post-processing to achieve a dense per pixel depth labeling. We find several ways of deal-

† http://www.agisoft.ru/

© The Eurographics Association 2014.



(a) Input image (b) Superpixel Size = 5 (c) Superpixel Size = 15 (d) Superpixel Size = 30

(e) Input depth (disparity) (f) Superpixel Size = 5 (g) Superpixel Size = 15 (h) Superpixel Size = 30

Figure 2: *SLIC superpixel size. (b) (c) (d) Different superpixel size values applied on the same input image (a). The superpixel size also affects the resulting disparity map (f) (g) (h), which will be the initial superpixel depth used in later with the MRF.*

ing with this in the literature: some hybrid human-in-the-loop approaches, where the information from the users is used to train an automatic system [KCGC11]; or approaches that try to fully automatically improve and propagate the depth information to every pixel in the image, such as the work in [VdBCVG13]. This last group is where we can currently classify our work.

3. Superpixel segmentation

Superpixel image segmentation is becoming increasingly popular as the key pre-processing step in plenty of computer vision tasks. This image segmentation provides a convenient form to compute local image features and reduces the complexity of many image processing tasks. It groups all the pixels in the image into different regions (covering all the image) with homogeneous properties, such as color content. This kind of segmentation assumes for instance that nearby pixels with similar color belong to the same object and in our particular problem, they have high probability to be at the same depth. For all these reasons, we found convenient to use superpixel segmentation. We assign a depth value to each superpixel, and in the next steps we propagate depth labels between superpixels.

In this work, we use the SLIC superpixel extraction algorithm [ASS*10], in particular the implementation provided in the VLFeat library [‡]. There are some parameters in this algorithm that will strongly influence the results of our next labeling and labeling propagation steps:

- **Superpixel size:** In Figure 2, we can see different extractions with different superpixel sizes. Using small superpixels leads to larger processing times, while choosing large superpixels hides the segmentation information in small and background objects.
- **Superpixel regularity:** In Figure 3, we can see that if we decrease the regularity restriction in the superpixels form, we obtain better superpixels because they fit better to the object boundaries.

1 0.1 0.01

Figure 3: *SLIC superpixels shape. Bottom row: SLIC regularity parameter. Top row: Superpixel depth according to the regularity parameter. The lower the regularity restriction, the better the segmentation fits object boundaries.*

How these parameters affect the final depth propagation are detailed in the Section 6.2.

4. Initial superpixel depth

To be able to initialize the next step in our pipeline (the global optimization of the image labeling using the MRF formulation), we need an initial superpixel depth, which we will construct combining the superpixels segmentation and a given input depth. As previously mentioned, we could obtain this input depth of an image from multiple sources (using

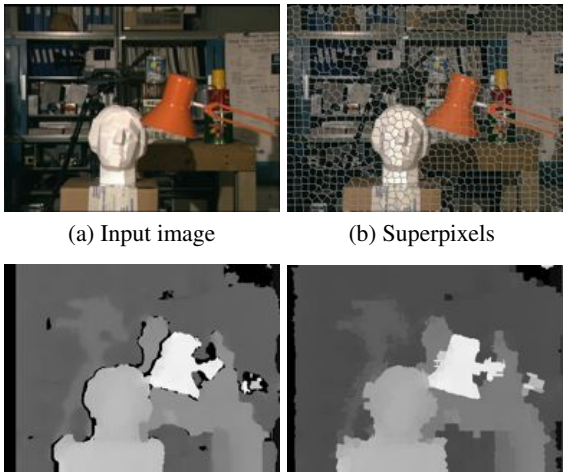
‡ VLFeat: An Open and Portable Library of Computer Vision Algorithms, http://www.vlfeat.org

© The Eurographics Association 2014.

multiple-view commercial software or state-of-the-art implementations, using depth and vision sensors or using stereo estimation), but in general, all of them frequently provide partially incomplete, sparse or incorrect depth estimation, i.e., there are pixels without an assigned depth value, what we will call *depth gaps* in the following.

Hence, in order to assign a depth value z to each superpixel S, we analyze the depth distribution among the pixels that belong to each superpixel and we choose the median depth value M_e as representative of that superpixel depth S_z. All depth values are normalized $\in [0,1]$. In cases where no pixel inside a superpixel S has a valid depth value, the superpixel gets assigned a 0 depth value ($S_z = 0$).

Using this simple step that merges the superpixel segmentation with the input depth we already manage to fill some depth gaps. In Figure 4 we can see an example where we improve the result in the estimated disparity map of an stereo pair if we combine it with the superpixel segmentation.



(a) Input image (b) Superpixels

(c) Input depth (disparity) (d) Initial superpixels depth

Figure 4: *(a) The original image is segmented in (b) superpixels. If we combine the superpixels segmentation with the input depth (c), disparity map, we obtain an improved disparity estimation (d).*

5. Depth propagation as a labeling problem

A Markov Random Field (MRF) provides a convenient way of modeling a labeling problem. The MRF defines an undirected graph G, where its nodes N represent a set of independent variables and its edges V represent the relationships between neighbor nodes. Given a set of labels L, a labeling problem consists in assigning to each node $p \in N$ a label $l \in L$. This problem can be formulated with an energy function E, which determines the total cost of a graph labeled. The energy equation 1 defines two costs: $C(l_p)$ denotes the cost to assign a particular label l to a node p and $C(l_p, l_q)$ denotes the cost related to two labels connected by an edge.

$$E = \sum_{p \in N} C(l_p) + \sum_{\{p,q\} \in V} C(l_p, l_q) \tag{1}$$

where $l_p \in L$ denotes the label l of the node p.

There are several techniques that deal with finding the optimal labeling, which minimizes this energy function. In our work, we use the graph cuts optimization [BVZ01] to resolve the energy minimization for Markov Random Fields. The code used in our experiments was provided by the authors [SZS*08].

The nodes in our MRF graph are the superpixels we have obtained. To build the connections (edges) in this graph, we need to determine the neighborhood condition between superpixels. We establish that two superpixels are neighbors when they share pixels between their borders. The labels assigned to each superpixel (node) consist on depth values. This approach favors that nearby superpixels have similar depth.

For defining the unary cost function $C(l_p)$ there are some specific aspects we want to take into account. We aim to favor that a superpixel preserves its initially assigned label z_p, except when this initial label is $z_p = 0$ (no depth information was found for that superpixel). Even so, this initial depth value can be incomplete (unlabeled pixels inside the superpixel) and noisy (inconsistent values of pixel depths). We analyze the distribution of pixel depth values within a superpixel, and we consider the accuracy a_p as the percentage of pixels within the superpixel p which have a valid depth value, and the variance σ^2 of its pixel depth values. This way, we measure how reliable are the superpixel original values. The expressions to calculate a_p and σ^2 are:

$$a_p = \frac{\sum_i^{n_p} (z_i > 0)}{n_p} \tag{2}$$

$$\sigma^2 = \frac{1}{2} \sum_{i=1}^{n_p} (z_i - \bar{z})^2 \tag{3}$$

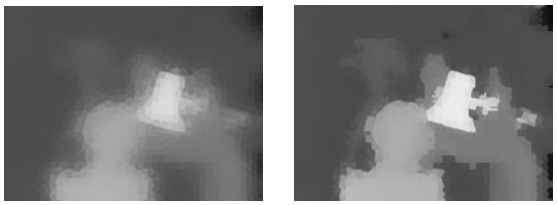
where z_i represents the depth value of pixel i and n_p the number of pixels of the superpixel p. This leads to the following unary const function:

$$C(l_p) = \begin{cases} 0 & : z_p = 0 \\ w_u \cdot a_p \cdot (1 - \sigma^2) \cdot (z_p - l_p)^2 & : z_p > 0 \end{cases} \tag{4}$$

where $w_u \in [0,1]$ is a control factor that leverages the effect of the unary cost function over the binary cost function. In Figure 5, we can see its effects in the depth propagation. When we increase the unary weight w_u, we reduce the global blur in the image, but increase the potential number of unlabeled or wrongly labeled superpixels.

With the unary cost function, we want to obtain higher cost when the label to be assigned is very different than the depth values that the superpixel originally had, except when depth value is 0. This value is modulated by the accuracy and noise of the pixel depths inside the superpixel.

© The Eurographics Association 2014.



(a) $w_u = 0.5$ (b) $w_u = 0.9$

Figure 5: *Increasing the weight w_u (unary vs. binary weight) we reduce the global blur in the image, but increase the potential number of unlabeled or wrongly labeled superpixels.*

For establishing the binary cost function $C(l_p, l_q)$, we consider that connected superpixels have similar depths. However, we assume that high color differences mark the boundaries between different objects that may lay at different depths. Therefore, we also include a measure about the actual similarity between two neighbor superpixels in the image (their appearance). Given two neighbor superpixels p and q, we compare their color histograms in the CIE-lab space color as follows:

$$d_{lab} = \frac{d(H_p^L, H_q^L) + d(H_p^a, H_q^a) + d(H_p^b, H_q^b)}{3} \quad (5)$$

where H_p^L represents the histogram in the luminance L channel of the superpixel p (with analogous definitions for the chrominance channels a and b and superpixel q). The color histograms are normalized between $[0,1]$ and the difference between two histograms $d(H_1, H_2)$ is defined as:

$$d(H_1, H_2) = \frac{\sum_i (H_1(i) - \overline{H_1}) \cdot (H_2(i) - \overline{H_2})}{\sqrt{\sum_i (H_1(i) - \overline{H_1})^2 \cdot \sum_i (H_2(i) - \overline{H_2})^2}} \quad (6)$$

We then define the binary cost function as follows:

$$C(l_p, l_q) = (1 - w_u)(1 - d_{lab})(l_p - l_q)^2 \quad (7)$$

where $(1 - w_u)$ is the weight of the binary cost function compared to the unary cost function (w_u has been defined in Equation 4).

With this binary cost equation, we want to encourage neighbor superpixels have similar labels. To avoid a global blur in the image, this cost depends on how similar the superpixels look on the image, i.e., the color similarity d_{lab} between the superpixels. This way, we manage to keep the object boundaries, because this similarity is likely to be low when superpixels belong to completely different parts or objects. We obtain high cost when two superpixels have different labels but they present a similar color distribution.

6. Experiments

This section presents experiments to validate the implemented pipeline, evaluate the proposed formulation for the energy function and measure the influence of the different terms and steps in the final solution. Section 6.1 presents a quantitative and exhaustive evaluation of the performance of our pipeline, comparing the results against a given ground truth. In section 6.2, we have analyzed how the different superpixel segmentation parameters affect to the solution obtained. Section 6.3 presents additional examples where the input depth has been obtained from a point cloud and a RGB-d camera respectively.

6.1. Quantitative evaluation of our approach

Our first tests are designed to evaluate the proposed cost functions and quantify the obtained improvements.

6.1.1. Dataset used

We use publicly available datasets [SS03, SS02], which are designed to evaluate stereo algorithms, where the ground truth represents the disparity between pixels from two images. Although, the disparity and the depth are not the same concept, they are closely related. In a stereo configuration (Figure 6), we only have a horizontal translation (without rotation) between the two cameras, and the disparity $disp$ can be calculated as the horizontal displacement between two corresponding pixels:

$$disp = x_L - x_R \quad (8)$$

$$z = \frac{fB}{disp} \quad (9)$$

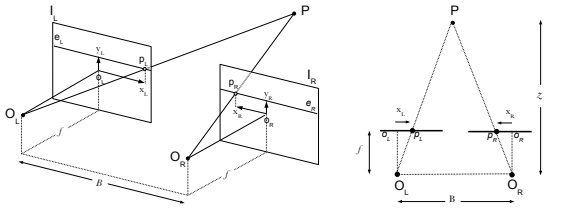


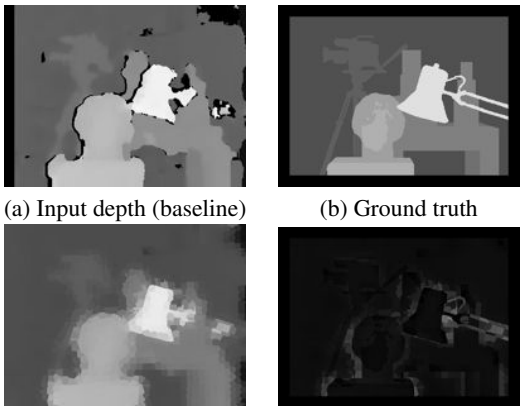
Figure 6: *In a stereo configuration the depth and disparity are inversely proportional.*

With this configuration, we know all parameters and can see that the disparity $disp$ and the depth z are inversely proportional. Hence, the points with same disparity belong to the same depth plane. The input depth, in this case the disparity map, which is going to be improved with our approach, is the result obtained with an implementation of the Hirschmuller algorithm [Hir08]. This algorithm computes stereo correspondence using the semi-global block matching algorithm.

6.1.2. Experimental set up.

To measure the improvement obtained in the depth estimation, we have evaluated how different parameters affect to

© The Eurographics Association 2014.



(a) Input depth (baseline) (b) Ground truth

(c) Depth propagation (d) Differences

Figure 7: *We calculate the (d) difference between the (b) ground truth values and (c) the solution provided by the (c) MRF from the (a) input depth, in this case a disparity map.*

the depth propagation. This performance, $\bar{\mu}_{\{G-I\}}$, is measured as how much we improve the initial depth, and it is calculated as the *mean of the differences (or mean error)* between each pixel in our resulting depth (after propagation) and the same pixel in the ground truth, as follows:

$$\bar{\mu}_{\{G-I\}} = \frac{\sum_p^N \left| l_p^G - l_p^I \right|}{\sum_p^N n_p} \tag{10}$$

where l_p^G denotes the labeling in the ground truth and l_p^I the our labeling proposed. Figure 7 shows the improvement achieved applying our depth propagation in a superpixel disparity map.

6.1.3. Results.

Figure 8 shows the improvements obtained using different cost functions. The *baseline* and *superpixels* represent the differences with the ground truth for the input disparity map and the initial superpixel depth respectively. The following bars represent variations on the parameters we use to build the cost function: *a* is the accuracy, σ^2 is the variance and *lab* means that we compare the color histogram between superpixels. These results in Figure 8 show that the depth propagation decreases the mean error for all the different cost functions we have tried, compared to *baseline* and *superpixels*, particularly noticeable as we increase the weight of the unary cost.

In the Figure 9, we show the numbers of iterations that were necessary to obtain the labeling with the minimum cost for the different cost functions. Less iterations are needed when we increase the weight of the unary cost. Doing so we also obtain better results as we can see in the Figure 5. We reduce the global blur in the image and keep the object boundaries.

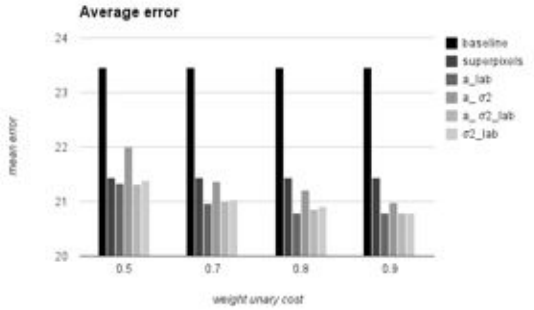


Figure 8: *Test-image tsukuba. The baseline and superpixels represent the differences with the ground truth for the input disparity map and the initial superpixel depth respectively. The other bars represent variations on the parameters we use to build the cost function: a is the accuracy, σ^2 is the variance and lab means that we compare the color histogram between superpixels. We always obtain better results if we run our depth propagation approach than with the input depth (disparity map), particularly noticeable when we increase the weight of the unary cost.*

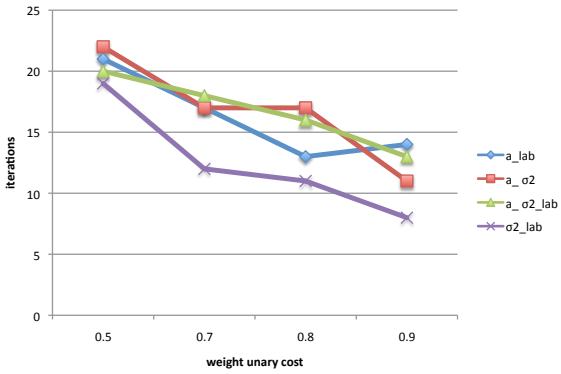
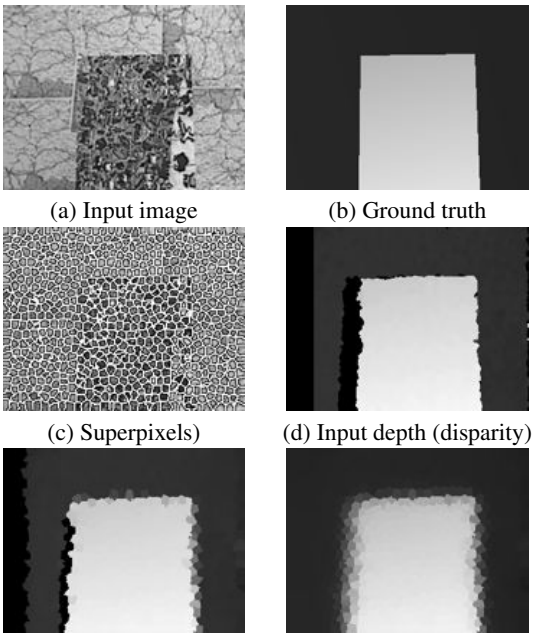


Figure 9: *Test-image tsukuba. We show the numbers of iterations that were necessary to obtain the labeling with the minimum cost for the different cost functions. Less iterations are needed when we increase the weight of the unary cost.*

The results of Figures 8 and 9 have been obtained using the test image *Tsukuba* from the evaluation dataset. Tests run with all the other dataset images are summarized in Table 1. We can see that the depth propagation always gets lower differences with regard to the ground truth than the input depth, therefore we always manage to improve that input depth, except for the test image *Map*. Figure 10 shows all steps of processing this test. In this case, the input depth is already a very good approximation, and we don't get to improve it with the depth propagation framework. This may be due to the fact that the background and object superpixels have very

© The Eurographics Association 2014.



(a) Input image (b) Ground truth

(c) Superpixels) (d) Input depth (disparity)

(c) Initial superpixel depth (d) Depth propagation

Figure 10: *Test image map. This is the only test image (a) where the depth propagation proposed in this work does not improve the initial input depth (d).*

Table 1: *Mean error (between different steps of our pipeline and the ground truth) for all dataset test images*

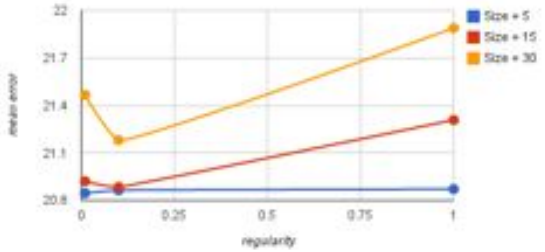
Input image	Input depth	Initial superpixel	Depth propagation
Tsukuba	23.4555	21.4349	20.7857
Venus	17.9249	13.3355	8.8710
Cones	31.9362	25.0465	8.9230
Teddy	32.0781	25.6495	9.6206
Sawtooth	16.0922	13.2377	9.51
Bull	12.7467	10.3099	7.4231
Poster	15.7758	11.4537	9.3676
Barn1	15.6588	12.3351	9.5196
Barn2	15.3817	12.8297	10.2926
Map	21.5471	23.2944	21.7738

similar textures, what prevents us from a good segmentation and propagation.

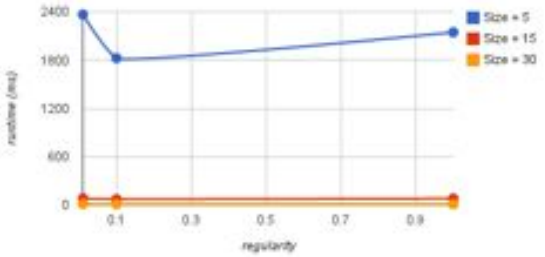
Our results prove that the MRF based propagation improves the obtained disparity and in most cases, it gets to eliminate all the artifacts. However, we can see that when a group of the black superpixels exists close to image boundary, the MRF does not get to eliminate all of them correctly, because three are not enough neighbors around the black superpixels. Figure 12 shows more result with some of the test images, showing the superpixel segmentation, the input disparity maps, the ground truth and the depth propagation results. These images show a clear improvement after running our approach with regard to the input disparity map.

6.2. Superpixel extraction parameters

As explained in previous section, the parameters (size and regularity) of the superpixel extraction algorithm affect to the initial depth labeling and hence, to the depth propagation. To measure their influence we have obtained superpixels segmentation, with different superpixel sizes and regularity, and we have compared their solutions with the ground truth.



(a) Mean Error



(b) Runtime time

Figure 11: *Mean error and runtime obtained with different superpixel extraction parameters. (a) With a large superpixel size, the difference mean error is higher (large sizes hide the segmentation information in small and background objects), while, with a small size, the numbers of superpixels increase and hence, (b) the runtime too. Regarding the regularity restriction, choosing a medium value we get the superpixels fit better to the object boundaries and we avoid to add noise pixels in the superpixel boundaries.*

In the Figure 11 we can see the difference mean error and runtime obtained. If we use a large superpixel size, the difference mean error is higher because a large size hides the segmentation information in small and background objects. However, if we choose a small size, the numbers of superpixels increase and hence, the runtime too. Regarding to the regularity restriction in the superpixels form, decreasing its

© The Eurographics Association 2014.

value, we get the superpixels fit better to the object boundaries but if the value is too small, we add noise pixels to the superpixel boundaries (Figure 3). In view of these results, to choose medium values of superpixel size and regularity is the best option.

6.3. Additional evaluation in different scenarios

With the following experiments we want to show the improvement obtained for depth maps which have poorly reconstructed regions. The depth maps of the first experiment have been obtained projecting a 3D point cloud into the image pixels. This point cloud was computed using a multiview stereo algorithm for 3D reconstruction of a scene from multiple views [FP10]. In Figure 13, we can see examples where we get to improve the initial depth map: the MRF fills the depth gaps and, in the first example, we correct the wrong superpixels in the bottom of the initial depth map.

The second experiment shows how we can improve the input depth obtained with a RGB-d camera, in particular a *Asus Xtion PRO LIVE*. These cameras usually provide depth maps with plenty of depth gaps. The images used to the tests belong to a publicly available dataset for activity recognition [§], other application that would benefit from improved depth estimation. Figures 14 shows some of the test images used. In these examples we can see that the RGB-d camera provides wrong or none information when objects are very close to the sensor or there are shadows in the scene. Our depth propagation approach improves the depth maps and fills all the gaps.

7. Conclusions and Future Work

There are plenty of algorithms that provide good estimations about the scene depth information from multiple views, and actually good depth information can be directly obtained from RGB-d sensors. However, most of these sources provide incomplete depth maps and in fields as 3D reconstruction, to get a perfect solution depends on these depth maps. Superpixel segmentation provides a convenient form to compute local image features and it reduces the complexity of image processing tasks. Combining superpixel segmentation with the depth maps we assign the same depth value to all of superpixel pixels. Although inside a superpixel, we get to propagate depth values, it would be better if we could share these values between different superpixels. Then, we can be consider it as a labeling problem. Many complex problems in computer vision require labeling each pixel as a preliminary step. In this work we have used superpixels instead of pixels. We have proposed using the depth propagation through a Markov random field (MRF) that models how superpixel graph. In a MRF we can decide the relation between a superpixels and a label and how its neighbor superpixels affect to it. With the results obtained in the depth propagation, we have improved the depth map in general cases, but there are values that it can not be correct. Then, the human interaction will be needed to improve the proposed solution. Our work can be useful as a previous step to user interactions. Usually, interactive algorithms require the user to provided tedious interactions to correct a scene. In future steps, we aim to combine our approach with other state-of-the-art techniques to learn from user interaction to improve the results and reduce the user interaction effort.

[§] https://i3a.unizar.es/es/content/wearable-computer-vision-systems-dataset

8. Acknowledgments

This work has been funded by project TAMA, Gobierno de Aragón.

References

[ASS*10] ACHANTA R., SHAJI A., SMITH K., LUCCHI A., FUA P., SÜSSTRUNK S.: Slic superpixels. *Ecole Polytechnique Fédéral de Lausssanne (EPFL), Tech. Rep 2* (2010), 3. 2, 3

[ASS*12] ACHANTA R., SHAJI A., SMITH K., LUCCHI A., FUA P., SUSSTRUNK S.: Slic superpixels compared to state-of-the-art superpixel methods. *Pattern Analysis and Machine Intelligence, IEEE Transactions on 34*, 11 (2012), 2274–2282. 2

[BVZ01] BOYKOV Y., VEKSLER O., ZABIH R.: Fast approximate energy minimization via graph cuts. *Pattern Analysis and Machine Intelligence, IEEE Transactions on 23*, 11 (2001), 1222–1239. 4

[CDSHD13] CHAURASIA G., DUCHENE S., SORKINE-HORNUNG O., DRETTAKIS G.: Depth synthesis and local warps for plausible image-based navigation. *ACM Transactions on Graphics (TOG) 32*, 3 (2013), 30. 2

[FH04] FELZENSZWALB P. F., HUTTENLOCHER D. P.: Efficient graph-based image segmentation. *International Journal of Computer Vision 59*, 2 (2004), 167–181. 2

[FP10] FURUKAWA Y., PONCE J.: Accurate, dense, and robust multiview stereopsis. *Pattern Analysis and Machine Intelligence, IEEE Transactions on 32*, 8 (2010), 1362–1376. 2, 8

[FVS09] FULKERSON B., VEDALDI A., SOATTO S.: Class segmentation and object localization with superpixel neighborhoods. In *Computer Vision, 2009 IEEE 12th International Conference on* (2009), IEEE, pp. 670–677. 2

[GMLMG12] GARCES E., MUNOZ A., LOPEZ-MORENO J., GUTIERREZ D.: Intrinsic images by clustering. *Computer Graphics Forum (Proc. EGSR 2012) 31*, 4 (2012). 2

[Hir08] HIRSCHMULLER H.: Stereo processing by semiglobal matching and mutual information. *Pattern Analysis and Machine Intelligence, IEEE Transactions on 30*, 2 (2008), 328–341. 5

[KCGC11] KOWDLE A., CHANG Y.-J., GALLAGHER A., CHEN T.: Active learning for piecewise planar 3d reconstruction. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on* (2011), IEEE, pp. 929–936. 3

[LGK10] LIU B., GOULD S., KOLLER D.: Single image depth estimation from predicted semantic labels. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on* (2010), IEEE, pp. 1253–1260. 2

© The Eurographics Association 2014.

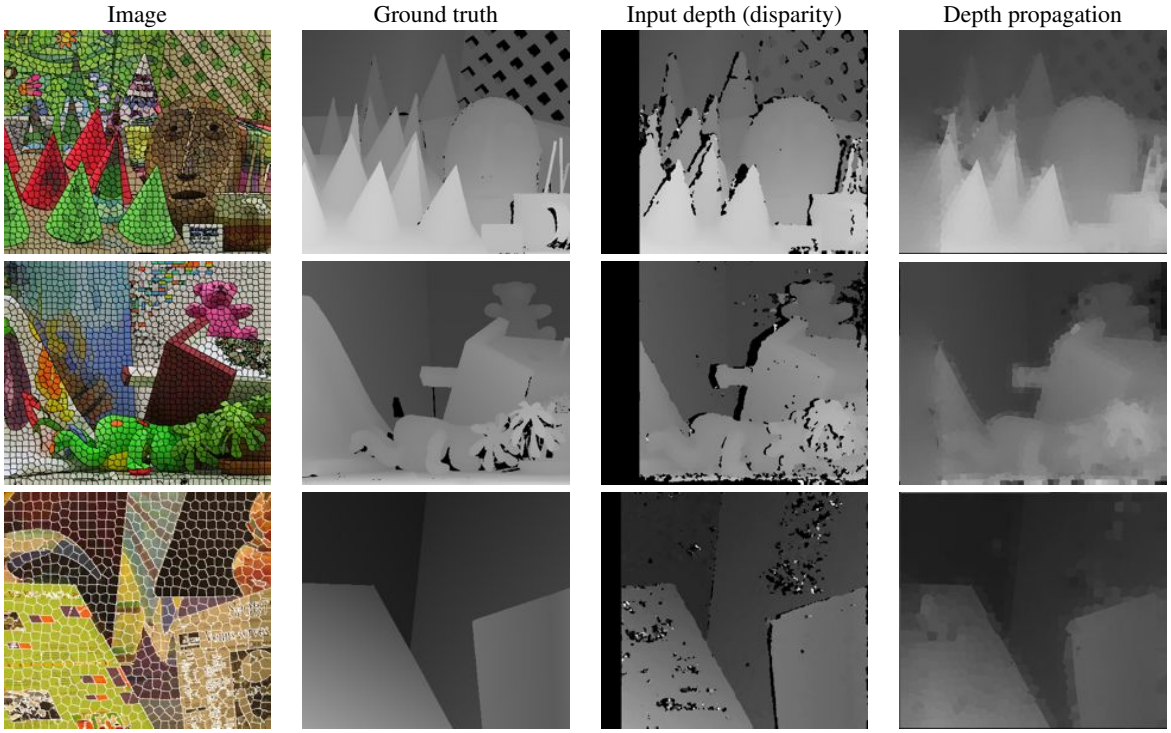


Figure 12: *Some test images: cones, teddy and venus respectively.*

[MAJ11] MISHRA A., ALAHARI K., JAWAHAR C.: An mrf model for binarization of natural scene text. In *Document Analysis and Recognition (ICDAR), 2011 International Conference on* (2011), IEEE, pp. 11–16. 2

[MK09] MICUSIK B., KOSECKA J.: Piecewise planar city 3d modeling from street view panoramic sequences. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on* (2009), IEEE, pp. 2906–2912. 2

[RM03] REN X., MALIK J.: Learning a classification model for segmentation. In *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on* (2003), IEEE, pp. 10–17. 2

[SBS12] SCHICK A., BAUML M., STIEFELHAGEN R.: Improving foreground segmentations with probabilistic superpixel markov random fields. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on* (2012), IEEE, pp. 27–31. 2

[SM00] SHI J., MALIK J.: Normalized cuts and image segmentation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on 22*, 8 (2000), 888–905. 2

[SS02] SCHARSTEIN D., SZELISKI R.: A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International journal of computer vision 47*, 1-3 (2002), 7–42. 5

[SS03] SCHARSTEIN D., SZELISKI R.: High-accuracy stereo depth maps using structured light. In *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on* (2003), vol. 1, IEEE, pp. I–195. 5

[SZS*08] SZELISKI R., ZABIH R., SCHARSTEIN D., VEKSLER O., KOLMOGOROV V., AGARWALA A., TAPPEN M., ROTHER C.: A comparative study of energy minimization methods for markov random fields with smoothness-based priors. *Pattern Analysis and Machine Intelligence, IEEE Transactions on 30*, 6 (2008), 1068–1080. 2, 4

[TL10] TIGHE J., LAZEBNIK S.: Superparsing: scalable nonparametric image parsing with superpixels. In *Computer Vision–ECCV 2010*. Springer, 2010, pp. 352–365. 2

[VdBBR*12] VAN DEN BERGH M., BOIX X., ROIG G., DE CAPITANI B., VAN GOOL L.: Seeds: Superpixels extracted via energy-driven sampling. In *Computer Vision–ECCV 2012*. Springer, 2012, pp. 13–26. 2

[VdBCVG13] VAN DEN BERGH M., CARTON D., VAN GOOL L. J.: Depth seeds: Recovering incomplete depth data using superpixels. In *WACV* (2013), pp. 363–368. 2, 3

[Wu13] WU C.: Towards linear-time incremental structure from motion. In *3DTV-Conference, 2013 International Conference on* (2013), IEEE, pp. 127–134. 2

[XQ09] XIAO J., QUAN L.: Multiple view semantic segmentation for street view images. In *Computer Vision, 2009 IEEE 12th International Conference on* (2009), IEEE, pp. 686–693. 2

© The Eurographics Association 2014.

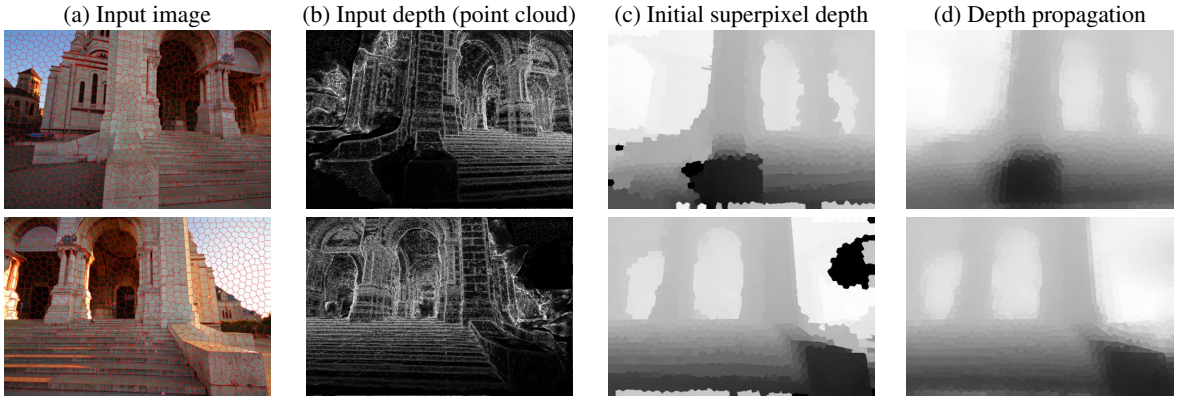


Figure 13: *Improving depth obtained from a multiview 3D reconstruction. The depth propagation (column (d)) fills the gaps and corrects the wrong superpixels. In these examples, there are a group of wrong superpixel labeling in the bottom of the initial superpixel depth (c). We can see how the depth propagation corrects these depth values.*

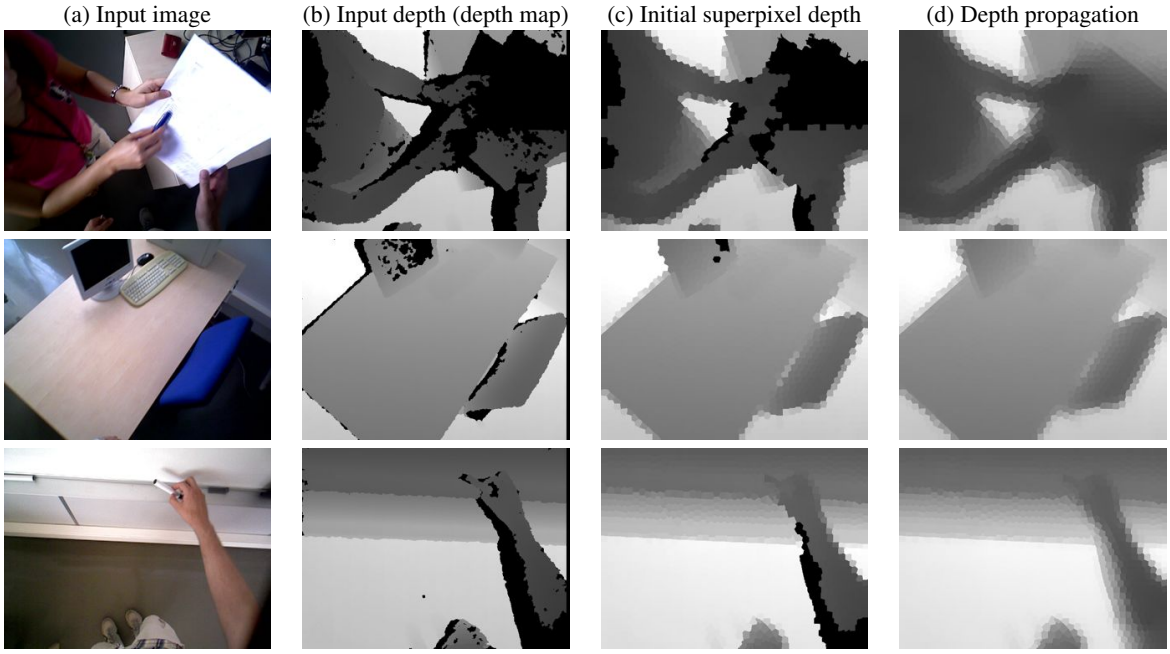


Figure 14: *Improving depth maps obtained with a RGB-d camera. In all these examples the depth propagation improves the input depth. The RGB-d camera provides wrong or none information when objects are very close to the sensor or there are shadows in the scene. Our depth propagation approach improves the depth maps and fills all the gaps.*

© The Eurographics Association 2014.