A Novel 2D to 3D Video Conversion System Based on a Machine Learning Approach

José L. Herrera, Carlos R. del-Blanco and Narciso García

Abstract — *There has been recently a significant increase in the number of available 3D displays and players. Nevertheless, the amount of 3D content has not increased in the same magnitude, creating a gap between 3D offer and demand. To reduce this difference, many algorithms have appeared that perform 2D-to-3D image and video conversion. These algorithms usually require several images from the same scene to perform the conversion. In this paper, an automatic algorithm for estimating the 3D structure of a scene from a single color image is proposed. It is based on the key assumption that color images with similar structure will likely present similar depth structures. The conversion algorithm is split into an offline and an online module to be easily deployable into consumer devices, such as smartphones or TVs. The offline module pre-processes a color+depth image database to speed up the subsequent depth estimation. The online module infers a depth prior from a color query image using the previous database as training data. Then, it is refined through a segmentation-guided filtering. The conversion algorithm has been evaluated in three publicly available databases, and compared with several state-of-the-art algorithms to prove its efficiency*[1].

Index Terms — depth extraction, 2D-to-3D conversion, depth maps, machine learning, clustering.

I. Introduction

In the last decade, the availability of displays and players with 3D capability, such as TVs, cinemas, smartphones, video game consoles, DVD/Blu-Ray players, and projectors, has experienced a significant rise. Nevertheless, the amount of 3D content that can be played in those devices, such as images, movies, or TV broadcastings, is still scarce. To close the gap between the number of 3D players/displays and the quantity of available 3D content, different techniques have appeared to convert the current 2D content into 3D [1].

This 2D-to-3D conversion task is usually divided into two stages. The first one is the estimation of a depth map from color images, and the second one is the Depth-Image-Based Rendering (DIBR) of a new image to form a stereo-par, or a multi-view set of images. While for the rendering stage there are algorithms that generate good quality results, the estimation of depth maps from color images is still a challenging process. For this reason, this paper is focused on recovering the depth information from 2D color images, and more specifically and challenging from a single color image.

The techniques used to convert 2D images into 3D ones can be divided into two groups, semi-automatic and automatic methods, depending on whether a human operator is involved in the process or not.

In semi-automatic methods, a human operator assigns depth values to different parts of a scene, creating a sparse depth map. Then, this map is processed to build a dense depth map of the whole image. Alternatively, the operator may assign a convenient depth prior map to the scene, which is then refined to include independent objects in the scene that can not be modeled by the prior map. The human operation varies from small sketches that assign depth values to different scene regions up to an accurate delimitation of objects in the scene with their corresponding depth value assignments. Nowadays, these methods represent the most successful strategy for 2D-to-3D conversion, and they are used by the industry to convert many 2D films into 3D. Nevertheless, the fact that a human operator must interact during the conversion process makes these methods highly time consuming and costly. An example of a semi-automatic system was presented by Guttmann et al. [2] to recover a dense depth map from sparse depth values assigned by the operator via diffusion. Similarly, Angot et al. [3] proposed an approach where a cross-bilateral filtering is applied to an initial depth map that is selected by a human operator from a library of prior depth maps. Phan et al. [4] presented a simplified and more efficient conversion method by using scale-space random walks, and a graph cut strategy. Liao et al. [5] reduced the human burden by computing first the optical flow, and then applying a structure from motion stage. The role of the human operator was reserved for correcting errors, and manually assigning depths to undefined regions.

In automatic approaches, no human operation is required to estimate the depth from images or videos. This fact accelerates the conversion task and makes potentially feasible to perform the conversion in real time [6], [7]. Many automatic depth extraction algorithms have been proposed, which use different pictorial cues such as defocus [8], motion [9], shading [10], or saliency [11]. The main problem of these methods is that they can not be usually applied to all the scenes that composed a movie (or generic video content). For example, structure from motion algorithms require that the camera is in motion, since

[1] This work has been partially supported by the Ministerio de Economía y Competitividad of the Spanish Government under project TEC2013-48453 (MR-UHDTV).

José L. Herrera is with Grupo de Tratamiento de Imágenes, Universidad Politécnica de Madrid, Madrid, 28040, Spain (e-mail: jhc@gti.ssr.upm.es).

Carlos R. del-Blanco is with Grupo de Tratamiento de Imágenes, Universidad Politécnica de Madrid, Madrid, 28040, Spain (e-mail: cda@gti.ssr.upm.es).

Narciso García is with Grupo de Tratamiento de Imágenes, Universidad Politécnica de Madrid, Madrid, 28040, Spain (e-mail: narciso@gti.ssr.upm.es).

the depth is estimated from different non-simultaneous views of the same scene. The combination of different cues (perspective geometry, defocus, and visual saliency) have been also proposed in some works [12], but they still rely on different acquisition or scene assumptions that are only satisfied for specific situations. There are also more simple approaches that rely on heuristic assumptions to provide a real-time 2D-to-3D conversion. Although some of them have been already embedded in many 3D commercial devices (such as TVs, video game consoles, or DVD/Blu-Ray players), a satisfactory conversion is provided only in some simple and restricted scenarios.

Recently, a new family of automatic methods based on machine learning have emerged, which uses a repository of color + depth images (RGBD) as training data to infer the depth of a color image. This approach is based on the key hypothesis that color images with similar gradient/edge/texture structure will likely present a similar depth configuration. A key advantage is that these methods can estimate a dense depth map from only one color image without any specific scene constrain. The general framework is as follows. Given a query color image, its depth map is estimated from a color+depth training image database by first finding the color images in the database that are most structurally similar to the query image. Then, the depth images corresponding to the previous selection of color images are combined to obtain the final depth map estimation. One of the seminal works was presented by Saxena et al. [13] who used a parsing technique and a Markov Random Field to estimate both 3D locations and orientations. An improved depth map estimation was achieved by Li et al. [14] and Liu et al. [15] through the inclusion of semantic labels and more sophisticated models. Konrad et al. [16] adopted a similar approach that transferred depth data instead of labels. Karsch et al. [17] included smoothness and consistency priors to refine the depth map estimation. More recently, Konrad et al. [18], [19] developed a more computationally-efficient approach by discarding the image alignment stage of previous works. Additionally, the feature descriptor Histogram of Oriented Gradients (HOG) were used to characterize and find structurally-similar images. As a result, the processing time was greatly reduced with a minimal impact on the achieved depth map quality. Similarly, Herrera et al. [20] employed Local Binary Patterns (LBP) as image features, and an adaptive strategy to select a variable number of similar images for the depth fusion process. One common limitation of these methods is that they adopt a nearest neighbor approach to find the most similar images in the database, which implies costly and exhaustive searches. This strategy can be impractical for huge databases, which would be required to estimate the 3D structure of a wide range of different scenarios. Another limitation is related to the characterization capacity of an image descriptor. There is not one that achieves the best performance for all the potential scenes.

In this paper, an automatic machine learning algorithm to infer the 3D structure of unconstrained scenes from a single color image is proposed, which addresses the previous limitations. Given a query color image, the algorithm starts by finding the most structurally similar color images in a database of color + depth images. The structure of color images is described by a combination of HOG, LBP, GIST, and Speeded-Up Robust Features (SURF) descriptors, which makes the algorithm more robust and adaptive to different types of scenarios (first contribution). The selection of the most similar images in the database is accomplished by a hybrid clustering and classification process that has a twofold advantage: adaptation to different scenes and fast image selection. Thus, it is possible to deal with huge databases (second contribution). The number of selected images and the specific combination of the feature descriptors are learned in the training phase with the purpose of obtaining an adaptive combination of parameters that improves the results (third contribution). The obtained selection of the most similar color images are then aligned with the query image via a fast feature-based image registration process, which additionally discards low quality registered images that can mislead the depth estimation. The depth maps of these color images are also aligned, and then fused to generate a depth prior estimation. Finally, a segmentation-based filtering refines the depth estimation by transferring smoothness and abruptness priors from the query color image to the depth prior estimation (fourth contribution).

The paper is organized according to the following structure. In Sec. II, an overview of the proposed depth extraction algorithm is presented. Next, the different stages are described in detail in Secs. III and IV. In Sec. V, the obtained results are presented and discussed. Finally, in Section VI, the conclusions are drawn.

II. Algorithm Overview

The proposed automatic 3D structure estimation algorithm can be formulated as follows. Given a query color image Q, and a database DB composed by color images and their corresponding depth maps (RGBD), the goal is to estimate the depth map D_{est} of Q. The algorithm is divided into two main modules: an offline pre-processing of DB to speed up the posterior image searches, and an online processing to estimate the depth of each incoming Q.

The offline module has a twofold purpose. The first one is to improve the efficiency of the image search strategy by clustering DB according to the structural similarity of the involved color images. The second purpose is to learn the best parameters (weight combination and number of candidate images) for the posterior depth extraction stage.

The online module estimates the depth of Q using the clusterized DB and the parameters learned from the offline module. Fig. 1 shows the block diagram of the proposed algorithm, including the two main modules, which are in turn composed by other sub-modules that will be described in the detail in the following sections.

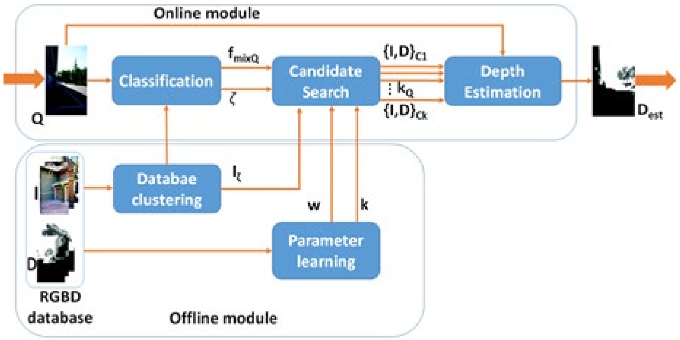


Fig. 1. Block diagram of the proposed 3D reconstruction method.

III. OFFLINE MODULE

The offline module performs two independent tasks. One is the division of *DB* into clusters that represent different classes of indoor and outdoor scenarios. This clustering is a key component to speed up the depth estimation of *Q* and deal with large databases. The other task is to learn some critical parameters of the system from *DB* to achieve the best performance in the posterior depth estimation process (online module).

A. Database Clustering

The clustering of *DB* is performed applying a k-means strategy over a set of feature-based representations of the color images. The feature-based representation combines four state-of-the-art image descriptors to capture the structural content: HOG, LBP, GIST, and SURF. The process is as follows. Every color image is divided into N_{row} x N_{col} blocks. Then, the four image descriptors are computed for each tile, which are then grouped and stacked per descriptor type as

$$\begin{aligned} f_{hog} &= \left[hog_{1,1}\ hog_{1,2} \cdots hog_{1,Ncol} \cdots hog_{Nrow,Ncol} \right] \\ f_{lbp} &= \left[lbp_{1,1}\ lbp_{1,2} \cdots lbp_{1,Ncol} \cdots lbp_{Nrow,Ncol} \right] \\ f_{gist} &= \left[gist_{1,1}\ gist_{1,2} \cdots gist_{1,Ncol} \cdots gist_{Nrow,Ncol} \right] \\ f_{surf} &= \left[surf_{1,1}\ surf_{1,2} \cdots surf_{1,Ncol} \cdots surf_{Nrow,Ncol} \right] \end{aligned} \tag{1}$$

where $hog_{n,m}$, $lbp_{n,m}$, $gist_{n,m}$, and $surf_{n,m}$, represent the HOG, LBP, GIST, and SURF descriptors, respectively, of the tile located in the n-th row and the m-th column. Subsequently, a super-descriptor representing the global image structure is obtained by concatenating the previous vectors as

$$f_{mix} = \left[f_{hog} \quad f_{lbp} \quad f_{gist} \quad f_{surf} \right]. \tag{2}$$

Finally, the previous vector is normalized as

$$f_{mix}^{norm} = \left(f_{mix} - \mu_{f_{mix}} \right) / \sigma_{f_{mix}} \tag{3}$$

using the mean $\mu_{f_{mix}}$ and the standard deviation $\sigma_{f_{mix}}$.

A weighted correlation function is used to compute distances among feature vectors in the k-means framework. The weighting operation allows to combine the different behavior of the descriptors that form part of f_{mix} to maximize its representation capability for every type of scenario (such as indoor and outdoor ones).

The number of clusters $N_{cluster}$ is a critical parameter in the algorithm for the computational efficiency and the quality results. A low value of $N_{cluster}$ tends to produce clusters whose images belong to very different scenarios. Therefore, their structural similarity is low, which is useless for a proper depth estimation. On the other hand, a large value of $N_{cluster}$ tends to create clusters whose images are more structurally similar, improving the quality of the depth map estimation. But the computational cost related to the image search significantly increases. As conclusion, an optimum tradeoff is necessary, as will be discussed in the result section. These clusters represent different classes of indoor and outdoor images in *DB* according to their structural similarity.

B. Parameter learning

The purpose of this module is to learn the following parameters to improve the quality of the conversion process: the weights *w* used for the weighted correlation function associated to the k-means clustering of *DB*, and the number of candidate images *k* that will be used in the depth estimation.

The learning process is based on the minimization of the depth estimation error of all the images in *DB*. The minimization procedure consists in an exhaustive search strategy performed over a dense grid of values of *w* and *k*. The depth estimation error computes the dissimilarity between the estimated depth map D_{est} of *Q* and its real depth map D_Q. The depth estimation process is the same as that performed by the online module, but using a Leave One Out (LOO) configuration. This configuration iteratively selects one image+depth pair from *DB* to be used as *Q* and D_Q, leaving the others pairs as training data for the estimation of D_{est}.

IV. ONLINE MODULE

The online module can divided into three stages. The first one is the classification of *Q* into one (or several) of the classes determined by the *DB* clusters (i.e. every cluster represents a class). Then, the search of the most similar color images to *Q* among the members of the resulting class (or classes) is performed. Finally, a combination of the depth maps corresponding to the previous selected color images is carried out to obtain D_{est}.

A. Classification

This stage selects the images in *DB* that are structurally more similar to *Q* via a classification technique, which determines the most probable classes in *DB* which *Q* belongs to. Notice that *Q* will share a high similarity with the images of these classes, since they were created using a structural similarity criteria. A block diagram of the classification stage can be seen in Fig. 2. First, a feature descriptor f_{mixQ} is computed from *Q* (see Sec. III.A), which is then delivered to a Nearest Neighbor classifier (NNC) to determine the $N\zeta$ most probable classes ζ in *DB*. For this purpose, the correlation function is used as similarity metric between f_{mixQ} and the image feature descriptors in *DB*. The value of $N\zeta$ is a tradeoff between the accuracy in the classification and the computational cost. A high value guarantees that the most similar images in *DB* are found, but at the expense of increasing the computational cost of the search performed by NNC. On the other hand, a low value speeds up the classification process, but images in *DB* that are similar to *Q* can be missed.

As result, a selection I_ζ of structurally similar color images to *Q* is obtained.

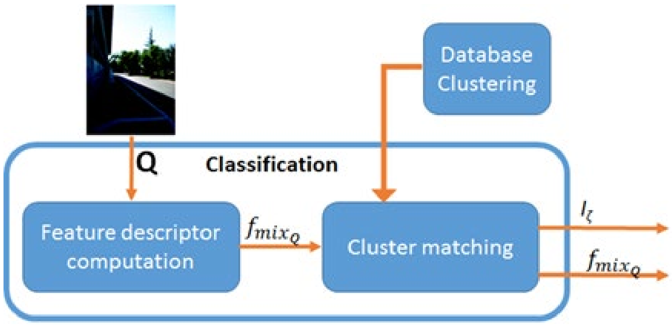


Fig. 2. Block diagram of the classification step.

B. Candidate search

This stage refines the previous selection of images I_ζ . Fig. 3 shows the block diagram composed by two modules. The first module, Parameter estimation, computes the most suitable parameters, w_Q and k_Q, for the given *Q*. These are computed via a weighted average of the parameters *w* and *k* (see Sec. III.B) related to each image in I_ζ , as shown Eq. 7

$$w_Q = \sum_i S(i)\,w(i) \Big/ \sum_i S(i)$$
$$k_Q = \sum_i S(i)\,k(i) \Big/ \sum_i S(i) \tag{7}$$

where

$$S(i) = corr\left(f_{mix_Q}, f_{mix_{i_m}}\right) \tag{8}$$

is the correlation value between f_{mix_Q} and $f_{mix_{i_m}}$ (the descriptor of the i-th image in I_ζ).

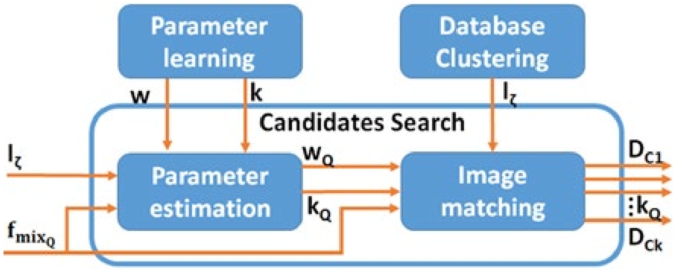


Fig. 3. Block diagram of the Candidate Search stage.

The second module, Image matching, performs an exhaustive search over the images in I_ζ to select the k_Q most structurally similar color images to *Q*. The search is carried out by a weighted correlation function *WS* given by

$$WS = \sum_{descr} w_Q S_{descr}$$
$$S_{descr} = corr\left(descr_Q, descr_{i_m}\right) \tag{9}$$

where *descr* can be f_{hog}, f_{lbp} f_{gist} or f_{surf}. Figs. 4 and 5 show examples of the images obtained by this search.

Finally, the corresponding depth images in *DB* of the previous k_Q color images are selected to perform the depth estimation of *Q*.

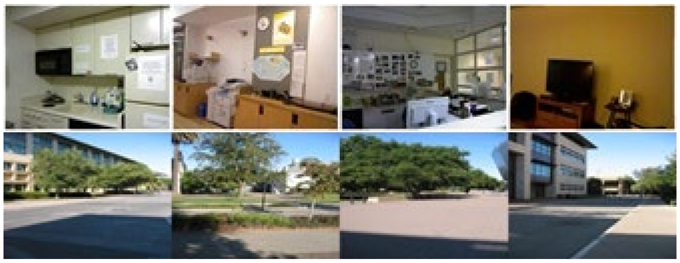


Fig. 4. From left to right: query image *Q* and the three most similar images to *Q* sorted by similarity for two examples of NYU dataset.

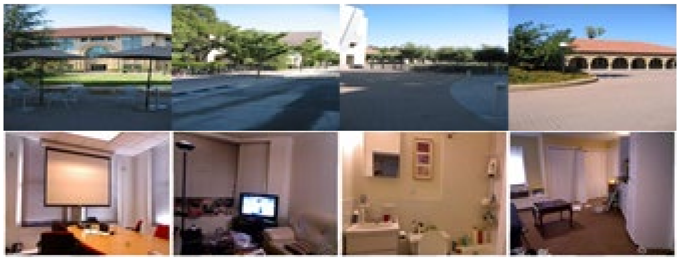


Fig. 5. From left to right: query image *Q* and the three most similar images to *Q* sorted by similarity for two examples of Make3D dataset.

C. Depth estimation

The depth estimation stage computes the depth map of *Q* using the selection of depth images obtained in the previous stage. Fig. 6 shows the block diagram of this stage. First, an image registration process is performed with the k_Q selected color images in *DB*. The registration is performed using the algorithm presented by Mahesh et al. [21]. This is a feature-based method that uses SIFT for detecting and matching key points, and RANSAC for the robust perspective image transformation.

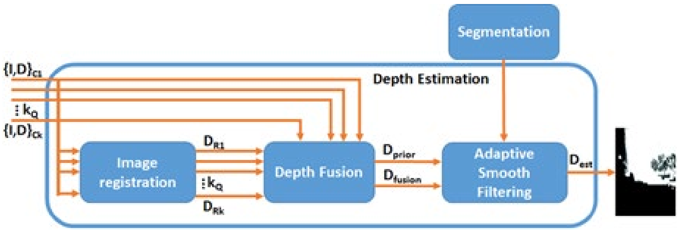


Fig. 6. Block diagram of the depth estimation stage.

Second, the resulting geometric transformation is applied to the corresponding depth images, which are then averaged to obtain the depth map D_{reg}.

In the previous registration process, there can be some problems. For example, the obtained transformation is considered unfeasible, or even though being feasible, the warped image does not cover entirely the image frame of *Q*. To solve this problem, a coarse depth estimation, $D_{fusion,}$ is computed by fusing all the considered depth images without any alignment. The fusion process is carried out by computing a weighted average of the candidate depth maps

$$D_{fusion} = \frac{\sum_{i=1}^{k_Q} WS(i) D_c(i)}{\sum_{i=1}^{k_Q} WS(i)} \quad (10)$$

where D_c is the *i-th* depth map to be fused, and *WS(i)* is the weighted similarity score computed in (9).

Then, a dense depth prior D_{prior} is obtained by filling in the missing information in D_{reg} with the values from D_{fusion}.

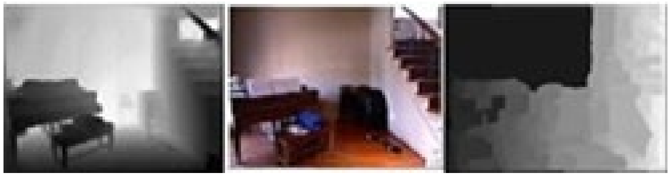


Fig. 7. From left to right: Actual depth, query color image *Q*, and segmentation of *Q*. A high correlation between the region transitions of color and depth can be observed.

Finally, D_{prior} is refined using an adaptive smooth filtering technique to reduce noise and artifacts, and enhance edge definition. This filtering is guided by a hierarchical segmentation applied to *Q*. I.e., the structure of *Q* is used to refine the depth estimation, taking advantage of the high correlation between the transitions of the color and depth maps of a scene. An example can be seen in Fig. 7, which shows a depth image, the corresponding color image, and the resulting segmentation of the color image. Observe the high correlation between the edges of the depth and color images. However, the color segmentation includes edges that do not have correspondence in depth, due to the different reflectance patterns that an object surface can have. This problem is addressed by forcing an over-segmentation, i.e. more image partitions than those that a human could determine. Under this situation, an object located at a certain depth is split into two or more regions in the color segmentation. However, the smoothing will be the same over those segmented regions since all of them have similar depth values. Notice that an under-segmentation is explicitly avoided since the filtering would average regions belonging to different objects that would have different depth values. For the purpose of obtaining an over-segmentation, the algorithm presented by Arbelaez et al. [22] has been used, which produces a hierarchical image segmentation that allows to control the degree of over-segmentation. This algorithm computes first the image contours by combining multiple local cues via a spectral clustering framework. And then, it performs the image segmentation by converting the computed contours into a hierarchical region tree. Next, an adaptive average filtering is applied over the estimated depth map, using a uniform kernel with an adaptive spatial size that depends on the region sizes of the over-segmented image. Mathematically, the kernel of the adaptive filtering can be defined as

$$s(x,y) = \begin{cases} 1, \ if\ (x,y) \in R_s \\ 0, \ if\ (x,y) \notin R_s \end{cases} \quad (11)$$

where R_s is a 2D over-segmented region containing the spatial coordinates of the depth pixel to be filtered.

The effect of the above adaptive smooth filtering technique is more notable when the image registration process is not feasible, since the fusion of misaligned depth images can produce a significant quality loss in the estimated depth image D_{est}. Thus, the algorithm can still produce reasonable quality depth maps, even though the image registration process is not feasible.

V. RESULTS

The proposed algorithm has been evaluated in three different databases: Make3D [13], NYU [23], and Stereo RGBD 1 [17] (a subset of Stereo RGBD from Karsch et al. containing indoor images). Make3D is composed by 534 outdoor images with their corresponding depths maps, acquired by a laser range finder. This database is divided into a test subset of 134 images and a train subset of 400 images. The resolution of the color and depth images are 1704 x 2272 and 55 x 305 pixels, respectively. NYU is formed by 1449 color images and their associated depth maps, captured by the Kinect sensor in indoor scenarios. The resolution is 640 x 480 pixels for both color and depth images. Stereo RGBD 1 is composed by 8431 indoor images and their associated depth maps, arranged in short video sequences. The resolution is 430 x 579 pixels for both color and depth images. A fourth database has been created and evaluated by joining the images from Make3D and NYU with the purpose to test the behavior of the conversion algorithm in hybrid databases containing indoor and outdoor situations. This database has been called Make3D-NYU.

For a straightforward comparison with the results presented by the previous works in the literature, color and depth images from the above databases have been resized to 320 x 240 pixels.

As was introduced in Sec III.A, the right selection number of clusters $N_{cluster}$ is a tradeoff between depth map accuracy and computational cost. Fig. 8 shows the normalized search time involved in the NN classification as a function of $N_{cluster}$. The normalization is performed respect to the search time considering only one cluster, which is virtually the same condition that not considering any cluster. As can be observed, the search time decreases fast, and remains stable in values between 5% and 20% (depending on the size of the database). Those percentages are obtained for values between 30 and 100 clusters. Consequently, the clustering strategy allows to deal with large databases, which would be unfeasible in a pure exhaustive search framework. Notice also that the search time for the Make3D dataset starts to rise slowly as the number of clusters is increased. The reason is that the hierarchical search with one cluster is computationally equivalent to the hierarchical search with a number of clusters equal to the total number of images in the database. This phenomenon in deep occurs for any database.

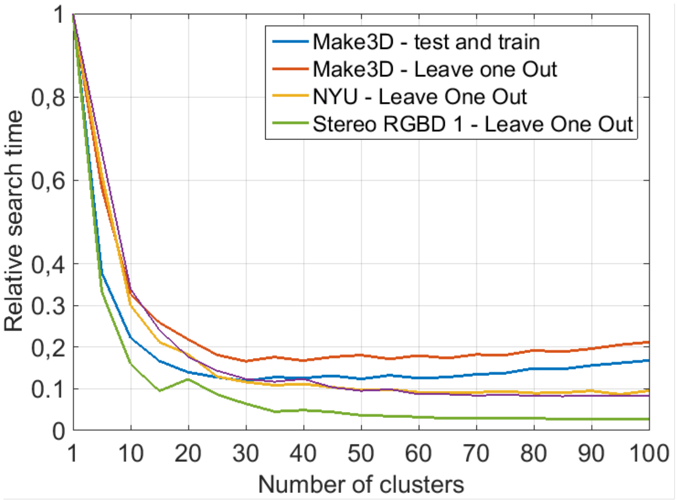


Fig. 8. Normalized search time as a function of the number of clusters.

To evaluate the presented 2D-to-3D conversion algorithm, it has been trained using a LOO methodology. The quality of the generated depth map D_{est} has been measured using the correlation function

$$C = \frac{\sum_i \left(D_{est}[i]-\mu_{D_{est}}\right)\left(D_Q[i]-\mu_{D_Q}\right)}{N_p \sigma_{D_{est}} \sigma_{D_Q}}, \qquad (12)$$

where N_p is the number of pixels in D_{est}, D_Q is the ground truth depth image, $\mu_{D_{est}}$ and μ_{D_Q} are the mean values of D_{est} and D_Q, respectively, and $\sigma_{D_{est}}$ and σ_{D_Q} the corresponding standard deviations. C takes values from -1 to +1, where values close to +1 indicate similar depth maps, and values close to -1 suggest they are complementary. The correlation function was first used in this context by Konrad et al. [19] to solve the problems that presented other metrics, such as the relative error (rel_{err}), the logarithmic error (log_{10} err), and the root mean square error (RMSE), which can be defined as

$$\begin{aligned} rel_{err} &= \frac{\left|D_Q(i)-D_{est}(i)\right|}{D(i)} \\ \log_{10} err &= \left|\log_{10}\left(D_Q(i)\right)\log_{10}\left(D_{est}(i)\right)\right| \\ RMSE &= \sqrt{\sum_i \left(D_Q(i)-D_{est}(i)\right)^2 / N} \end{aligned} \qquad (13)$$

These metrics are not invariant to lineal transformation of the depth maps, which implies that only good results would be obtained if depth estimations are in the same range of values as the actual depth maps. However, most of the 2D-to-3D conversion methods only can estimate the scene depth up to a scale factor. Other problem is that rel_{err} and log_{10} err metrics cannot be computed if D_Q has some zero-value pixels. Table I illustrates the problems of these metrics using the Make3D dataset (considering a test subset of 134 images and a train subset of 400 images). This table shows the results of the presented proposal and other state-of-the-art algorithms in the literature. The results are the average over the 134 test images. Note that some results are not available, since some metrics are not used by all works (indicated by the letters `NR': Not Reported). As can be observed, the results are not very coherent, since one algorithm can seem to be better than another depending on the used metrics. Therefore, it is difficult to claim which algorithm performs better.

TABLE I
RESULTS ON MAKE3D DATABASE

Algorithm	Rel Err	Log10 Err	RMSE
Baseline	0.698	0.334	NR
Learn Depth: MRF [24]	0.530	0.198	16.7
Pointwise MRF [13]	0.458	0.149	NR
Superpixel MRF [13]	0.370	0.187	NR
Surface Layout [25]	1.423	0.320	NR
Cascade Models [26]	NR	NR	15.4
Ground Plane [27]	NR	NR	22
Semantic Labe3ls [15]	0.375	0.148	NR
Feedback Cascades [28]	NR	NR	15.2
Theta-MRF [29]	NR	NR	15.0
Depth Transfer [17]	**0.361**	**0.148**	15.1
Depth Fusion [19]	0.432	0.187	18.3
Adaptive LBP [20]	0.628	0.184	16.8
Proposed approach	0.505	0.180	**14.7**

Evaluation of state-of-the-art algorithms using the Relative Error (Rel. Err.), the Logarithmic Error (Log10 Err.) and the Root Mean Square Error (RMSE) metrics in the Make3D database.

The correlation measure does not depend on absolute values, and therefore it is not affected by the aforementioned problems. For this reason, it is used to fairly compare the quality of the different algorithms, provided that they have either reported their results using the correlation, or at least their code is available to compute them. More specifically, the proposed algorithm has been compared with the HOG-Based Depth Learning approach of Konrad et al. [19], the Depth Transfer approach of Karsch et al. [17], and the Adaptive LBP-based approach [20].

Table II shows the correlation results using the four proposed databases. Results for Make3D are obtained using a fixed subset of 400 images for training and other of 134 images for testing. A LOO configuration has been adopted for NYU, Make-NYU and Stereo RGBD 1 datasets. The results are averaged over all the images in each dataset. As can be seen, a decrease in the quality is produced for indoor images in all methods. Indoor images are more challenging due to their higher variability (mainly because of the huge variation of different objects than can be present). The proposed approach not only reduces the impact of this decrease, but also outperforms the other approaches in both NYU (outdoor images) and Make3D-NYU (combination of outdoor and indoor images). Regarding Make3D (outdoor images), the presented approach achieves similar results to Depth Transfer algorithm [17]. Overall the proposed conversion algorithm can obtain better results in real situations, where indoor and outdoor scenes are mixed. The results for and Stereo RGBD 1 are significantly better because the dataset is composed by short video sequences. In this case, the classification and candidate search modules find images that are really close to Q.

TABLE II
QUALITY RESULTS ON DIFFERENT DATABASES

Algorithm	Make3D	NYU	Make-NYU	Stereo RGBD 1
Depth Transfer [17]	**0.69**	0.59	0.60	NR
Depth Fusion [19]	0.610	0.61	0.60	0.90
Adaptive LBP [20]	0.66	0.63	0.62	0.90
Proposed approach	0.67	**0.63**	**0.63**	**0.97**

Evaluation of state-of-the-art algorithms using the Correlation Coefficient (C) in different databases.

Table III shows the average computational time in seconds for the 2D-to-3D conversion for different state-of-the-art algorithms and for all the considered databases. As can be seen, the conversion time for the proposed algorithm is significantly lower than for the others. Unlike other approaches, the presented approach performs some processing tasks in an offline module, whose time is also reflected in the table (notice that this processing is not involved in the conversion process).

TABLE III

COMPUTATIONAL TIMES

Algorithm	Make3D	NYU	Make-NYU	Stereo RGBD 1
Depth Transfer [17]	**92.7**	98.7	101.4	108,3
Depth Fusion [19]	0.83	3.22	4.11	18.16
Adaptive LBP [20]	0.98	3.81	4.93	19.84
Proposed approach	0.17	**0.48**	**0.52**	**1.7**
Offline module	6.9	**7.2**	**7.3**	**8.3**

Evaluation of the computational time in seconds for different state-of-the-art algorithms and for different databases.

Figs. 9 and 10 show some examples of depth map estimations for the Make3D and NYU, respectively. Due to the nature of these databases, the image registration is not feasible (there are not images close enough to *Q*). Therefore, D_{prior} has only a moderate quality, but the adaptive smooth filtering achieves to enhance the quality of the final depth estimation.

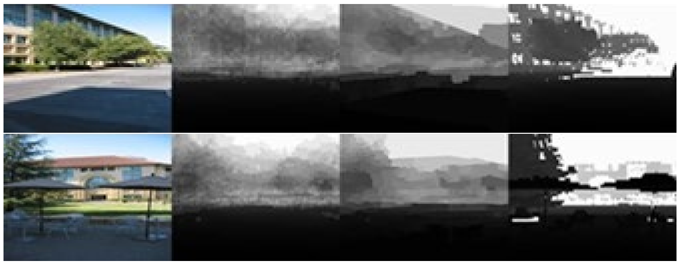


Fig. 9. From left to right: query image *Q*, depth prior D_{prior}, final depth estimation D_{est}, and depth ground truth D_Q for two examples in Make3D.

Fig. 11 shows an example of the obtained depth map for the Stereo RGBD 1. In this dataset, the image registration is feasible, and consequently the estimated depth map offers a higher quality. In this case, the impact of the adaptive smooth filtering is not very noticeable.

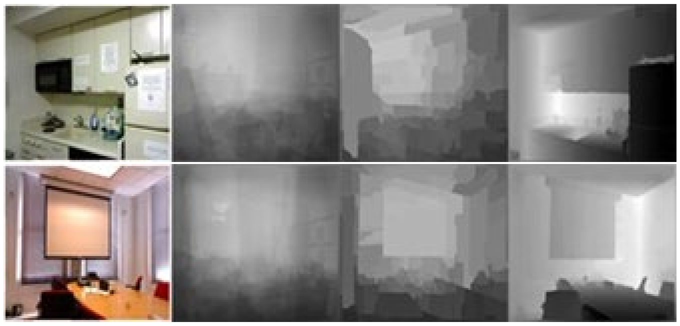


Fig. 10. From left to right: query image *Q*, depth prior D_{prior}, final depth estimation D_{est}, and depth ground truth D_Q for two examples in NYU.

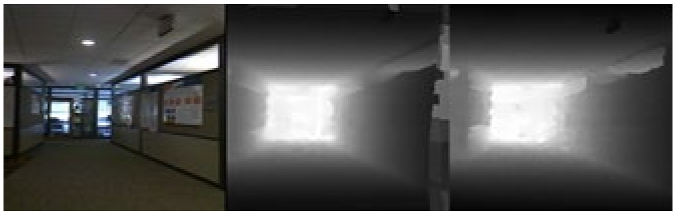


Fig. 11. From left to right: query image *Q*, final depth estimation D_{est}, and depth ground truth D_Q for one example in Stereo RGBD 1. D_{prior} is not shown because of his high similarity to D_{est}.

VI. CONCLUSIONS

There has been proposed a new automatic 2D-to3D conversion algorithm for indoor and outdoor scenarios that is based on a machine learning framework. It can effectively estimate a dense depth map from a single color image. The presented approach learns the optimum weights for combining several feature descriptor, as well as the optimum number of candidate database images to be used in the depth map estimation. The algorithm also clusters the database according to their structural similarity to be able to use realistic and huge database images. The algorithm is divided into two different modules. The offline module is run beforehand and involves the main computational cost, alleviating the burden of the online module and making feasible the implementation of the proposed approach in consumer electronics devices such as smartphones or TVs

The approach achieves similar or higher results to the best algorithms in the state of the art, outperforming them for the most challenging cases, such as the indoor scenes, and for the combinations of indoor and outdoor scenes. Additionally, in cases where very similar images are available, as in video sequences, the presented approach experiments a very significant increment in the quality, due to the inclusion of an image registration step.

REFERENCES

[1] C. C. Cheng, C. T. Li, and L. G. Chen, "A novel 2D-to-3D conversion system using edge information," *IEEE Trans. Consumer Electron*., vol. 56, no. 3, pp. 1739-1745, Aug. 2010.

[2] M. Guttmann, L. Wolf, and D. Cohen-Or, "Semi-automatic stereo extraction from video footage," *in Proc. IEEE International Conference on Computer Vision*, Kyoto, Japan, pp. 136-142, Sept. 2009.

[3] L. J. Angot, W. J. Huang, and K.-C. Liu; "A 2D to 3D video and image conversion technique based on a bilateral filter". In *Proc. SPIE 7526, Three-Dimensional Image Processing (3DIP) and Applications, 75260D, San José, USA,* vol. 7526, pp. 75260D, Feb. 2010.

[4] R. Phan, R. Rzeszutek, and D. Androutsos, "Semi-automatic 2D to 3D image conversion using scale-space Random Walks and a graph cuts based depth prior," in Proc. *IEEE International Conference on Image Processing*, Brussels, Belgium, pp. 865-868, Sep. 2011.

[5] M. Liao, J. Gao, R. Yang, and M. Gong, "Video Stereolization: Combining Motion Analysis with User Interaction," *IEEE Trans. Vis. Comput. Graphics*, vol. 18, no. 7, pp. 1079-1088, Jul. 2012.

[6] Z. Zhang, S. Yin, L. Liu, and S. Wei, "A real-time time-consistent 2D-to-3D video conversion system using color histogram," *IEEE Trans. on Consumer Electron*., vol. 61, no. 4, pp. 524-530, Nov. 2015.

[7] S. F. Tsai, C. C. Cheng, C. T. Li, and L. G. Chen, "A real-time 1080p 2D-to-3D video conversion system," *IEEE Trans. on Consumer Electron*., vol. 57, no. 2, pp. 915-922, May 2011.

[8] M. Subbarao and G. Surya, “Depth from defocus: A spatial domain approach,” *Int. J. Comput. Vision*, vol. 13, no. 3, pp. 271–294, 1994.

[9] R. Szeliski and P. Torr, “Geometrically constrained structure from motion: Points on planes,” *Lecture Notes in Computer Science*, vol. 1506, pp. 171–186, Nov. 1998.

[10] R. Zhang, P.-S. Tsai, J. E. Cryer, and M. Shah, "Shape-from-shading: a survey," *IEEE Trans. Pattern Anal. Mach. Intell*., vol. 21, no. 8, pp. 690-706, Aug. 1999.

[11] C. Huang, Q. Liu, and S. Yu, “Regions of interest extraction from color image based on visual saliency,” *The Journal of Supercomputing*, vol. 58, no. 1, pp. 20–33, Oct. 2011.

[12] P. Ji, L. Wang, D.-X. Li, and M. Zhang, “An automatic 2D to 3D conversion algorithm using multi-depth cues,” in Proc, *International Conference on Audio, Language and Image Processing*, Shangai, China, pp. 546–550, 2012.

[13] A. Saxena, M. Sun, and A. Ng, “Make3D: Learning 3D scene structure from a single still image”, IEEE *Trans. on Pattern Anal. and Mach. Intell*., vol. 31, no. 5, pp. 824-840, May 2009.

[14] C. Liu, J. Yuen, and A. Torralba, “Nonparametric scene parsing: Label transfer via dense scene alignment”, in Proc *IEEE Conference on Computer Vision and Pattern Recognitio,* Miami, USA, pp. 1972-1979, Jun. 2009.

[15] B. Liu, S. Gould, D. Koller, “Single image depth estimation from predicted semantic labels”, *IEEE Conf. on Comput. Vis. and Pattern Recognit.,* San Francisco, pp. 1253-1260, Jun. 2010.

[16] J. Konrad, G. Brown, M. Wang, P. Ishwar, C. Wu, D. Mukherjee, “Automatic 2D-to-3D image conversion using 3D examples from the internet”, *Proc. SPIE,* vol. 8288, pp. 82880F, Mar. 2012.

[17] K. Karsch, C. Liu, and S. Kang, “Depth extraction from video using non-parametric sampling”, *in Proc European Conference. in Computer Vision*, Firenze, Italy, pp. 775-788. Oct. 2012

[18] J. Konrad, M. Wang, and P. Ishwar, “2D-to-3D image conversion by learning depth from examples”, *in Proc IEEE. Conference on Computer Vision And Pattern Recognition, Workshops*, Providence, USA, pp. 16-22, Jun. 2012

[19] J. Konrad, M. Wang, P. Ishwar, C. Wu, and D. Mukherjee, “Learning-based, automatic 2D-to-3D image and video conversion”, *IEEE Trans. on Image Process*., vol. 22, no. 9, pp. 3485-3496, Sep. 2013.

[20] J. Herrera, C. del Blanco, and N. García, “Learning 3D structure from 2D images using LBP features”, *IEEE Int. Conf. on Image Processing*, Paris, pp. 2022-2025, Oct. 2014.

[21] Mahesh and M. V. Subramanyam, "Automatic feature based image registration using SIFT algorithm," *in Proc International Conference on Computing, Communication. and Networking Technologies,* Coimbatore, India, pp. 1-5, Jul. 2012.

[22] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik, “Contour detection and hierarchical image segmentation”, *IEEE Trans. on Pattern Anal. and Mach. Intell*., vol. 33, no. 5, pp. 898-916, May 2011.

[23] N. Silberman and R. Fergus, “Indoor scene segmentation using a structured light sensor”, *in Proc IEEE Int. Conference on Computer Vision Workshops*, Barcelona, Spain, pp. 601-608 Nov. 2011.

[24] A. Saxena, H. C. Sung, and Y. N. Andrew, “Learning depth from single monocular images”, *Adv. Neural Inform. Process. Syst*., vol. 18, pp 1-8, Dec. 2005.

[25] D. Hoiem, A. Efros, and M. Hebert, “Recovering surface layout from an image”, *Int. J. Comput. Vision*, vol. 75, no. 1, pp. 151-172, Oct. 2007.

[26] G. Heitz, S. Gould, A. Saxena, D. Koller, “Cascaded classification models: Combining models for holistic scene understanding”, *Adv. Neural Inform. Process. Syst.,* vol. 21, pp. 641-648, Dec. 2008.

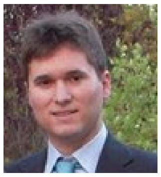
[27] A. Cherian, V. Morellas, and N. Papanikolopoulos, “Accurate 3D ground plane estimation from a single image”, *in Proc. IEEE International Conference on Robotics and Automation,* Kobe, Japan, pp. 2243-2249, May 2009.

[28] C. Li, A. Kowdle, A. Saxena, and T. Chen, “Toward holistic scene understanding: Feedback enabled cascaded classification models”, *IEEE Trans. on Pattern Anal. and Mach. Intell*., vol. 34, no. 7, pp. 1394-1408, Jul. 2012.

[29] C. Li, A. Saxena, and T. Chen, “Ɵ-MRF: Capturing spatial and semantic structure in the parameters for scene understanding”, *Adv. Neural Inform. Process. Syst.* vol. 24, pp. 549-557, Dec. 2011.

BIOGRAPHIES

José L. Herrera received the Telecommunication Engineering degree (integrated BSc-MSc accredited by ABET) from the Universidad Politécnica de Valencia (UPV), Spain in 2009, and the Technologies and Systems of Communications Master degree from the Universidad Politécnica de Madrid (UPM), Spain in 2013. He is currently a PhD student at the same University. Since 2011 he has been a member of the Image Processing Group of the UPM. His research interests are in the area of 3D image and video.



Carlos R. del-Blanco received the Telecommunication Engineering and Ph.D. degrees in telecommunication from the Universidad Politécnica de Madrid (UPM), Madrid, Spain, in 2005 and 2011, respectively. Since 2005, he has been a member of the Image Processing Group, UPM. Since 2011, he has also been a member of the faculty of the ETS Ingenieros de Telecomunicación as an Assistant Professor of ´ signal theory and communications at the Department of Signals, Systems, and Communications. His professional interests include signal and image processing, computer vision, pattern recognition, machine learning, and stochastic dynamic models.

Narciso García received Telecommunication Engineering degree and the Ph.D. degree in Telecommunication, both from the Universidad Politécnica de Madrid (UPM), in 1976 (Spanish National Graduation Award) and 1983 (Doctoral Graduation Award), respectively. Since 1977 he is a member of the faculty of the UPM, where he is currently Professor of Signal Theory and Communications. He leads the Image Processing Group of the UPM. He was Coordinator of the Spanish Evaluation Agency from 1990 to 1992 and evaluator, reviewer, and auditor of European programs since 1990. His professional and research interests are in the areas of digital image and video compression and of computer vision.