UNIVERSIDAD POLITÉCNICA DE MADRID

ESCUELA TÉCNICA SUPERIOR

DE INGENIEROS DE TELECOMUNICACIÓN

2D to 3D image and video conversion using
machine learning

Ph. D. Thesis
Tesis Doctoral

José Luis Herrera Conejero

Ingeniero de Telecomunicación

2020

DEPARTAMENTO DE SEÑALES, SISTEMAS Y
RADIOCOMUNICACIONES
ESCUELA TÉCNICA SUPERIOR
DE INGENIEROS DE TELECOMUNICACIÓN

**2D to 3D image and video conversion using
machine learning**

Tesis Doctoral

Autor:

José Luis Herrera Conejero

Ingeniero de Telecomunicación
Universidad Politécnica de Madrid

Directores:

Narciso García Santos

Doctor Ingeniero de Telecomunicación
Universidad Politécnica de Madrid

Carlos Roberto del Blanco Adán

Doctor Ingeniero de Telecomunicación
Universidad Politécnica de Madrid

2020

TESIS DOCTORAL

2D to 3D image and video conversion using machine learning

Autor: José Luis Herrera Conejero

Directores: Narciso García Santos

Carlos Roberto del Blanco Adán

Tribunal nombrado por el Mgfco. y Excmo. Sr. Rector de la Universidad Politécnica de Madrid, el día de de 2020.

Presidente: D.

Vocal: D.

Vocal: D.

Vocal: D.

Secretario: D.

Realizado el acto de defensa y lectura de la Tesis el día de de 2020 en

Calificación:

EL PRESIDENTE

LOS VOCALES

EL SECRETARIO

A Beatriz
y a ti, apreciado lector.

Agradecimientos

Llegados a este punto del camino, me gustaría pararme un momento, echar la vista atrás y dedicar unas breves líneas a agradecer a todos los que, de una manera u otra, han contribuido a que estemos hoy aquí. El camino ha sido largo y, sin duda, no son pocos los que han sido condición necesaria para que esto sea posible. En primer lugar, quiero empezar por mis directores de Tesis.

Muchas gracias, Narciso y Carlos R., por vuestra guía y la mucha paciencia durante todo este tiempo y por el sprint final. Gracias a Narciso porque a pesar de lo apretado de su agenda y lo cotizado y disputado de su tiempo, cada minuto suyo ha resultado de un provecho inefable. Gracias a Carlos R. por estar al pie del cañón día a día, por aguantar conmigo un viernes hasta *medianoche* – ε para submitir un paper cuando, a pesar de lo apasionante que pudiera sonar el plan, había seguro opciones más tentadoras.

También quiero agradecer a todo el resto del personal del GTI con el que he tenido el gusto y el privilegio de coincidir en estos años. Ha sido una gran ayuda y he aprendido mucho de todos. Gracias también por todos esos partidos de fútbol y por hacer crecer mi nivel futbolístico aparente de infinitesimal a paupérrimo.

No quiero olvidarme de los meses de estancia en Boston. Gracias Janusz por acogerme en tu grupo y por tu guía y tiempo esos meses. También agradecer a mis compañeros de despacho en Boston Univeristy, Jonathan y Amanda el trato recibido haciendo la estancia mucho más agradable. Thank you so much.

Gracias también a mis amigos. Gracias, Álex, Yago, Iñaki, Belén y Ana por tanto tiempo compartido, tanto jinx, tanto strike y tanto club. Y gracias a todos los que habéis preguntado y dejado de preguntar por la tesis, pues ambas situaciones han sido de agradecer.

No puedo despedirme sin agradecer antes a Gauss y todo su legado en forma de leyes, teoremas, campanas, filtros, ruido, métodos, club, 16 gigas de RAM y seguro que multitud de cosas más, que me han acompañado desde hace tantos años. También agradecer al artículo 116 de la Constitución Española y al Real Decreto 463/2020, que me permitieron reunir el tiempo suficiente para realizar el sprint final de esta carrera que ya toca su fin.

Gracias también a ti, estimado lector, si no te has encontrado en ninguna de las categorías anteriores. Seguramente si estás leyendo esto es porque tenía cosas que agradecerte, y de no ser el caso, gracias igualmente por dedicarme parte de tu valioso tiempo. Espero que si estás buscas algo en concreto, lo encuentre pronto, sea en este documento o en cualquier otro.

Por último, pero no por ello menos importante quiero dar las gracias a mi familia. Sobre todo a mis padres y hermanos que son quienes me han acompañado desde el principio y no sería quien soy sin ellos.

Y dejando lo mejor para el final, quiero dar las gracias a Beatriz, que me precedió en este camino que ahora llega a su fin y que me acompaña en uno que no lo tiene. Gracias por todo.

Resumen

A pesar de que el concepto de imagen y vídeo tridimensional (3D) existe desde hace muchos años, recientemente se ha producido un incremento significativo en el número de reproductores y monitores 3D disponibles. Sin embargo, la cantidad de contenido 3D no ha sufrido un aumento de la misma magnitud, provocando una brecha entre la oferta y la demanda de contenido 3D. Para reducir esta diferencia, han aparecido muchos algoritmos que realizan la conversión de imagen y vídeo de 2D a 3D. Mientras muchas de estas técnicas, normalmente requieren de varias imágenes del mismo escenario para llevar a cabo la conversión, la familia más reciente de estas técnicas, que se corresponde con los algoritmos basados en aprendizaje automático, no están restringidos por esta limitación, calculando la imagen 3D con una única vista de la escena. Los métodos basados en aprendizaje automático requieren el uso de bases de datos de imágenes 2D y 3D para aprender cómo llevar a cabo esta conversión. Debido a que el número de bases de datos disponibles ha aumentado recientemente de manera significativa, estos algoritmos se han vuelto muy populares. No obstante, la calidad conseguida por las técnicas actuales de conversión 2D-3D está lejos de ser plenamente satisfactoria y necesita ser mejorada para poder ser usada para producir contenido 3D de buena calidad.

Esta tesis propone dos sistemas para la conversión 2D a 3D, uno para imágenes y otro para vídeo, correspondientes a la familia de métodos basados en aprendizaje automático. Con relación al sistema de conversión de imágenes, se propone un nuevo enfoque que hace el algoritmo más robusto y con mayor capacidad de adaptación a diferentes tipos de escenarios usando una combinación de descriptores de características. Al mismo tiempo, la propuesta de dividir la base de datos por grupos hace que la solución sea más rápida y eficiente para lidiar con grandes bases de datos. El sistema aprende además como adaptar automáticamente los valores de los diferentes parámetros involucrados en la conversión, dando como resultado una solución plenamente automática.

Respecto a la conversión de vídeo, la propuesta para imágenes basada en aprendizaje automático se extiende para abarcar secuencias de vídeo. El algoritmo se divide en tres partes principales. En la primera, se realiza una estimación de la profundidad del fondo utilizando el sistema previamente presentado. Posteriormente, se segmenta el primer plano analizando el flujo óptico para procesar los diferentes objetos presentes en la escena de forma individual. Finalmente, la estimación de profundidad del fondo se combina con la información de los objetos del primer plano y se filtran para obtener una estimación final de la profundidad.

Los algoritmos desarrollados han sido probados con diferentes bases de datos de imágenes y secuencias de vídeo 3D que están disponibles de forma pública.

Abstract

Despite the concept of three-dimensional (3D) image and video was introduced many years ago, there has been recently a significant increase in the number of available 3D displays and players in the last years. 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. While many of this techniques usually required several images from the same scene to perform the conversion, the most recent family of these techniques, which corresponds to machine learning-based algorithms, is not restricted by this limitation computing the 3D image with a single view of the scene. Machine learning-based methods require the use of databases of 2D and 3D images to learn how to perform this conversion. Since the number of available datasets has recently increased significantly, these algorithms have become very popular. However, the quality achieved by the current 2D to 3D conversion techniques is far to be fully satisfying and they need to be improved in order be used for producing good quality 3D content.

This thesis proposes two systems for 2D-to-3D conversion, one for image and another one for video, belonging to the machine learning family of methods. With respect to the image conversion system, a new approach is proposed that makes the algorithm more robust and adaptive to different types of scenarios by using a combination of feature descriptors. At the same time, the proposed clustering of the dataset makes the solution faster and more efficient to deal with large datasets. The system also learns how to automatically adapt the value of the different parameters involved in the conversion, resulting in a fully automatic solution.

With respect to video conversion, the learning based approach for images is extended to video sequences. The algorithm is divided in three main parts. In the first one, a depth estimation of the background is computed using the previous presented approach. Then, the foreground is segmented analyzing the optical flow to manage the different objects individually. Finally, the depth estimation of the background is combined with the foreground information and filtered to obtain the final depth estimation.

The developed algorithms have been tested with different publicly available datasets of 3D images and video sequences.

Contents

1	Introduction	1
1.1	Motivation	1
1.2	Goals	1
1.3	Contributions	2
1.4	Document structure	3
2	State of the art	5
3	System overview	11
3.1	Introduction	11
3.2	Proposed 2D-to-3D image conversion system	11
3.3	Video conversion	13
3.4	Conclusion.	14
4	2D to 3D image conversion system	15
4.1	Image representation and hierarchical search	15
4.1.1	Compact image representation	15
4.1.2	Database clustering	16
4.1.3	Candidate Search	17
4.2	Depth estimation	19
4.2.1	Depth fusion	20
4.2.2	Segmentation-based edge enhancement	21
4.3	Parameter learning	23
4.4	Conclusion.	24
5	2D to 3D video conversion	25
5.1	Introduction	25
5.2	System	25
5.3	Background depth estimation	26
5.4	Optical flow and foreground segmentation	26
5.5	Foreground-background combination	27
5.6	Filtering	27
5.7	Conclusion	28
6	Results	29
6.1	Quality metrics	29
6.2	Image representation and hierarchical search	31

6.3	Depth estimation	34
6.4	Parameter learning	39
6.5	Global results for 2D to 3D image conversion	40
6.6	2D to 3D video conversion	40
6.7	Conclusions	43
7	Conclusions and future work	45
7.1	Image representation and hierarchical search	45
7.2	Depth estimation	45
7.3	Parameter learning	46
7.4	2D to 3D video conversion	46
7.5	Future work	46

List of Figures

1.1	Overall workflow of the 2D-to-3D image and video conversion system.	2
2.1	Overall workflow of a semi-automatic 2D-to-3D image conversion system.	6
2.2	Overall workflow of the structure from motion algorithm[1].	7
2.3	Overall workflow of a machine learning-based 2D-to-3D conversion system.[2] . . .	8
2.4	Overall workflow of a CNN-based 2D-to-3D conversion system [3].	9
3.1	Overall algorithm for both Konrad [4] and Karsch [5] 2D-to-3D conversion approaches [4].	12
3.2	General working diagram of machine learning-based 2D-to-3D image conversion.	13
3.3	Block diagram of the machine learning-based 2D-to-3D image conversion proposed approach [6].	13
3.4	Block diagram of the proposed 2D-to-3D video conversion system [7].	14
4.1	The feature descriptors are computed in each image tile, and then concatenated to form an image descriptor [8].	16
4.2	Block diagram of the coarse search module [6].	18
4.3	Block diagram of the fine search module[8].	19
4.4	Block diagram of the depth estimation stage. [8]	20
4.5	Depth map candidates D_C are registered to better fit the edges of the query image Q , and then, combined to build the depth prior D_{prior}	20
4.7	From left to right: query color image, depth prior and depth estimation. A significant correspondence between many of the edges in the color image and the edges of depth estimation can be observed [8].	22
4.6	From left to right: actual depth, query color image Q , and segmentation of Q . A significant correspondence between many of the edges in the color image and in their associated depth images can be observed [6].	22
4.8	Diagram of the parameter learning process [8].	23
5.1	Block diagram of the 2D-to-3D video conversion system [7].	25
5.2	Two examples of frames sequences and their optical flow results	26
6.1	Variations in the quality level for every database and for each feature descriptor. Quality values are normalized to the quality level corresponding to the proposed image descriptor f_{mix}	31

6.2	Search time as a function of the number of clusters $N_{cluster}$ in which the database is divided. Time values are normalized to the search time corresponding to $N_{cluster} = 1$. This is virtually equivalent to not dividing the database into clusters [6].	32
6.3	Variations in the achieved quality level as a function of the number of clusters $N_{cluster}$ in which the database is divided. Quality values are normalized to the quality level corresponding to one cluster.	34
6.4	Effect of the registration task in the quality of the final depth estimation D_{est}	35
6.5	Variations in the quality level achieved in every database when using a median operator and a weighted average operator for depth fusion. Quality values are normalized to the quality level corresponding to median operator.	36
6.6	Variations in the quality level achieved in every database for different filters applied to D_{prior} for edge enhancement.	36
6.7	From left to right: Query image Q , depth prior D_{prior} , final depth estimation D_{est} , and ground truth for the NYU database [8].	37
6.8	From left to right: Query image Q , depth prior D_{prior} , final depth estimation D_{est} , and ground truth for the Make3D database [8].	38
6.9	Difference in quality achieved using a uniform mix descriptor and an adaptive one.	39
6.10	Difference in quality achieved using a fix number of candidate images and an adaptive one.	40
6.11	Variation of the quality (C) as a function of the number k of images used in the conversion [9]	41
6.12	From left to right: Query Frame, Optical Flow binary mask, Background Depth Estimation, Final Depth Map Estimation and Depth Ground Truth for images of StereoRGBD Indoor1 database [7].	43

List of Tables

6.1	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 [6].	30
6.2	Evaluation of the computational time in seconds for different state-of-the-art algorithms and for different databases [8].	33
6.3	Evaluation of the state-of-the-art algorithms using the RMSE, PSNR and normalized cross-covariance (C) metrics for the Make3D (outdoor images) with 400 images for train and 134 images for test. The results are the average over the 134 test images[8].	41
6.4	Evaluation of the state-of-the-art algorithms using the RMSE, PSNR and normalized cross-covariance (C) metrics for the NYU (indoor images) in LOOCV configuration. The results are averaged over the 1449 images of the dataset. [8].	41
6.5	Evaluation of the state-of-the-art algorithms using the RMSE, PSNR and normalized cross-covariance (C) metrics for Make3D and NYU (outdoor and indoor images) in LOOCV configuration. The results are averaged over the 1983 images of both datasets.[8].	41
6.6	Evaluation of state-of-the-art algorithms using the RMSE, PSNR and normalized cross-covariance(C) metrics in the static camera sequences of the StereoRGBD Indoor1 database in Leave One Out configuration. The results are the average over the 6967 test images [7].	42

Chapter 1

Introduction

1.1 Motivation

Despite the concept of three-dimensional (3D) image and video was introduced many years ago, there has been in the last decade when 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. 3D video has been established as a key feature in the visual experience for a significant group of users.

Nevertheless, the mass production of 3D content that can be played in those devices, such as images, movies, or TV broadcastings, is still a pending issue producing an important gap between the capability of playing/displaying 3D content and the volume of this type of content. To close this gap between offer and demand of 3D content, different techniques to convert the current 2D content into 3D [1] have been developed. The most recent family of these techniques corresponds to machine learning-based algorithms that is still far to be fully satisfying and they need to be improved in order to be used for producing good quality 3D content.

1.2 Goals

This main goal of this thesis is to develop a new machine learning-based 2D-to-3D conversion algorithm for estimating the depth of 2D images and videos, that improves the existing machine learning-based algorithms in the state of the art. In order to achieve this goal, some improvements to the different modules of the classical machine learning-based 2D to 3D conversion algorithm will be developed.

As can be seen in Fig. 1.1. the 2D-to-3D conversion task for both images and video consists in two main steps [10]. In the first one, given a query image or video, it is generated an estimated depth map using image or video analysis techniques. In the second one, the Depth Image Based Rendering (DIBR) technique is used to build, from a query image or video and its corresponding estimated depth map, an stereo (or multiview) 3D image or video. This document, is focused on the first step, that results in a more challenging task than DIBR, since there are many algorithms that performs this second step with a satisfactory quality.

In particular, there are three points in which the effort of this work are focused:

1. Improve the quality of the final depth estimations of the 3D images and videos.

Figure 1.1: Overall workflow of the 2D-to-3D image and video conversion system.

2. Improve the robustness of the system for a higher variety of scenarios.
3. Reduce the computational cost of the algorithm and make it able to deal with larger databases

1.3 Contributions

The 2D-to-3D image and video conversion algorithm exposed in this thesis presents the following main contributions:

1. Database clustering. Dividing the dataset into different clusters helps the algorithm to better find those images in the database that are more similar to query one. This contribution was first published in [11], and the effect of the clustering in the quality of the conversion as well as in the speed of the algorithm may be seen in Sec. 6.2. This contribution is focused on goal 3 in Sec. 1.2.
2. Adaptive feature descriptor combination. Using more than one feature descriptor to represent the images alleviates the limitations that each feature descriptor has by itself. By assigning different weights to different descriptors adaptively, depending on the specific situation, makes the system more flexible. This contribution was first introduced in [6], and the effects on the quality achieved by the system are shown in Sec. 6.2. This contribution tackles goals 1 and 2 in Sec. 1.2.
3. Edge enhancement improvement. An adaptive size filtering stage to refine the depth estimations was first introduced in [12]. A new segmentation-based version of the filtering task, which outperforms the previous results was published in [8]. The effects of this contribution in the quality of the presented approach is reflected in Sec. 6.3. This contribution is related to goal 1 in Sec. 1.2.
4. Adaptive number of candidates and weights. The use of adaptive number of candidate images and their associated weights leads the system to a better performance, as it can be observed in [2]. Sec. 6.4 shows the effects that this contribution has over the performance of the system. This contribution is focused on goal 1 in Sec. 1.2.
5. Parameter learning. The optimal parameters involved in the conversion task for different situations are learned in the training stage and this knowledge is used in the test stage to increase the performance. It was first introduced in [8]. Sec. 6.4 exposes how this

contribution affects the results of the proposed algorithm. This contribution faces goals 1 and 2 in Sec. 1.2.

6. Mix learning + optical flow. A framework that combines this learning-based strategy with other classical video analysis techniques, such as optical flow, extends the image conversion system to videos, as was first presented in [7], and then extended in [6]. The results of this contribution are presented in Sec. 6.6. This contribution is related to goals 1 and 2 in Sec. 1.2.

On summary, the clustering stage of the algorithm allows the system to scale for large datasets preventing the computational cost to increase excessively and makes it more flexible to work in different situations (i.e. indoor, outdoor, urban, nature...), whereas the others exposed contributions are oriented to increase the performance of the system presented in the current document.

1.4 Document structure

The current document is structured as follows. Chapter 2 introduces the state of the art in 2D-3D image and video conversion. Chapter 3 presents the general process of the 2D-3D image and video conversion system. Chapter 4 describes the different subsystems the image conversion algorithm as follows: Section 6.2 focuses on the image compact representation and the clustering performed over the dataset. Section 6.3 explains how depth estimations are built. Section 6.4 exposes the learning process followed to estimate some key parameters of the 2D-to-3D conversion. In Chapter 5 it is presented the proposed 2D-to-3D image and video conversion approach. Chapter 6 shows the partial and global result of the presented approaches. Finally, Chapter 7 exposes the conclusions of the presented thesis and future work lines to continue improving the machine learning-based 2D-to-3D conversion systems.

Chapter 2

State of the art

There are two different strategies for 2D-to-3D image conversion. Semi-automatic methods require a human operator intervention in the conversion process, while automatic methods do not need such human interaction.

Semi-automatic methods, as can be seen in Fig. 2.1, need a significant human interaction for the conversion process to segment the different objects in the scene, and then giving them an appropriate depth. 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 to 3D [4]. Nevertheless, the fact that a human operator must interact during the conversion process makes these methods highly time consuming and costly. For example, a semi-automatic system was proposed by Guttmann et al. [13] to recover a dense depth map from sparse depth values assigned by the operator via diffusion. Angot [14] proposed an approach where cross-bilateral filtering is applied to an initial depth map that is selected by the human operator from a library of initial depth maps. A similar approach was presented by [15], where the cross-bilateral filtering is replaced by an edge-based filtering. Also, based on [13], Phan et al. [16] presented a simplified and more efficient conversion algorithm by using scale-space random walks and a graph cut strategy. Liao et al.[17] reduced the human burden by computing first the optical flow, and then applying a structure from motion stage, reserving the role of the operator to correct errors, and manually assigning depth to undefined areas in the automatically generated depth maps.

On the contrary, automatic methods do not need any human intervention to perform the 2D-to-3D conversion. This fact accelerates the conversion task and makes potentially feasible to perform the conversion in real time [18]. There are many classic approaches that have faced this problem attending to different cues, such as shape from shading [19] and visual saliency [20]. A combination of different cues (perspective geometry, defocus, and visual saliency) have been also used in [21]. All these methods rely on different assumptions, and therefore they only achieve satisfactory depth maps for specific situations. Alternatively, some approaches try to recover the depth of the scene from different non-simultaneous views of the same scene using structure from motion algorithms [22]. Alternatively, a depth-from defocus algorithm can be used to achieve the same goal by modifying the focal length of the camera [23].

One of the most relevant techniques for the 2D-to-3D conversion task is Structure from Motion, which has been briefly introduced before. A wide range of works in the literature use the structure from motion framework, which estimates the 3D structure of a scene from a set of

Figure 2.1: Overall workflow of a semi-automatic 2D-to-3D image conversion system.

photographs that either have some scene overlapped, or have been taken by a camera that was in motion, i.e. a video sequence [24]. The structure from motion techniques (see Fig. 2.2) use the correspondence information between sparse feature points computed from the set of images to iteratively estimate the camera pose in the 3D space, the internal parameters of the camera, and the the 3D space position of the matched points. All these methods yield a sparse 3D point cloud that represents the scene structure, making few assumptions about the input data, and therefore widely applicable to different scenarios. Snavely et al. [25] developed a system based on the structure from motion technique, called Photo Tourism, which allowed the virtual navigation in a urban environment reconstructed from a large amounts of registered photographs (taken from a network of cameras). Other works have exploited the availability of unstructured collections of Internet images [26, 27] to perform the reconstruction under the structure from motion paradigm, trying to improve the performance and accuracy, and at the same time strengthening the scalability behavior of the algorithm at dealing with huge numbers of input photographs. Alternatively, some methods perform a line-feature correspondences across images [28], which can improve the reconstruction quality for urban entities characterized for presenting these features, for instance buildings and streets.

However, structure from motion techniques usually generate quite sparse 3D clouds that do not contain any solid geometry. Multi-view Stereo is an evolution of the structure from motion framework that delivers a dense structure containing information from all the pixels of the input images. They usually use a structure from motion technique as a first step, and then try to densify the obtained sparse 3D point cloud. For this purpose, many multi-view stereo methods use the plane-sweeping methodology introduced by Collins [29]. This technique tests a family of plane hypotheses by estimating how many intersecting rays, coming from pixel projections of several images, lies over a candidate plane. The set of planes with more accumulated intersection points are used to represent the dense 3D structure of the scene. Bauer et al. [30] proposed a method that used plane sweeping to specifically densify sparse point clouds from buildings. Another approach for 3D dense reconstruction is to perform pairwise dense matching between two registered images. The resulting depth maps are then fused to improve the quality of the final estimated depth map. Hirschmüller [31] proposed a robust dense matching method based on a cost function that maximizes the mutual information. This cost function had the

Figure 2.2: Overall workflow of the structure from motion algorithm[1].

advantage of being robust to radiometric differences of input images. Multi-view Stereo has been also satisfactorily applied for large scale problems, such as the work of Frahm et al. [32] that processed almost 3 million images to compute a dense depth map in a single day with a single PC.

Structure from Motion and Multi-view Stereo methods can reconstruct 3D scenes without any consideration about the singularities of the image contents. There are other works that try to improve the quality of the computed 3D models by exploiting the specific characteristics of certain urban entities. Buildings are a meaningful examples that exhibits common visual cues, such as parallelism and orthogonality, which can be used for the reconstruction task. Several works have used these visual cues to extract vanishing points and line features from several images [33, 34] to recover both the camera parameters and 3D structure. Specially interesting is the work of Hoiem et al. [35] that only needed one image to obtain a basic reconstruction (a simplified pop-up 3d model). This work used the estimated vanishing points to perform first an image segmentation, and then depth assignments. In the same line, Barinova et al. [36] applied a Conditional Random Field model for the reconstruction of urban scenes from one image.

In the last years, several machine learning-based algorithms have appeared as an alternative to the classical 2D-to-3D conversion methods (see Fig. 2.3). The main hypothesis behind this family of methods is that images with a high photometric similarity will likely have a similar 3D structure. Therefore, using a color+depth image database, the depth map of a query color image can be inferred by first finding the color images in the database most similar to the query image, and finally by combining the depth images associated to the previous selection of color images. In this sense, Saxena et al. [37, 38] developed a learning approach for estimating the depth of a scene from a single image using a parsing technique and Markov Random Fields, which estimate both 3D locations and orientations. In [39] and [40], the inclusion of semantic labels and more

Figure 2.3: Overall workflow of a machine learning-based 2D-to-3D conversion system.[2]

elaborated models were considered to achieve better depth maps. Konrad et al. [41] presented a similar strategy, but transferring depth data instead of labels. Following this methodology, Karsch et al. [5] incorporated a depth optimization processing to assure the smoothness and consistency of a candidate depth map. Also, they extended the approach to work with videos. More recently, Konrad et al. [42, 4] developed a computationally-efficient approach by avoiding the SIFT-flow based image alignment. Instead, the Histogram of Oriented Gradients (HOG) feature descriptor were used to directly find photometrically-similar images. Moreover, they applied a Cross-Bilateral Filtering (CBF) to enhance the final depth map estimation. This way, they achieved to reduce the processing time in several orders of magnitude due to the discarding of the SIFT-flow alignment, which had a minimal impact on the quality.

Even more recently, inspired by the rapid development of deep learning technology for image classification, object detection and semantic segmentation, many approaches have been proposed to estimate single image depth. Different deep learning architectures, such as Convolutional Neural Network (CNN, see Fig. 2.4) based methods[43, 44, 45, 46], recurrent neural network (RNN)[47, 48], and generative adversarial network (GAN)[49, 50] have been presented to automatically convert single 2D images to 3D. These methods are based on the the extraction of abstract features of the query images and the prediction of a dense pixel-wise depth map, and generally achieve better performances compared with more traditional methods.

This document presents two approaches for the 2D-to-3D conversion task (one for image conversion and another one for video). which addresses the previous limitations. Related to the image conversion, 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[51], LBP[52], GIST[53], and SURF[54] descriptors, which makes the algorithm more robust and adaptive to different types of scenarios

Figure 2.4: Overall workflow of a CNN-based 2D-to-3D conversion system [3].

than using only one feature descriptor. 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 larger databases. 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. 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.

While all the previously mentioned methods try to solve the 2D-to-3D conversion for images, the video conversion task has not been so widely discussed by the machine learning based algorithms due to some other strategies such as Structure from Motion (SfM) achieve very good results in the case of video sequences. In the second approach presented in this document, the learning based approach for images is extended to video in those sequences with a static camera that are not so well solved by techniques such as SfM. The algorithm is divided in three main parts. In the first one, a depth estimation of the background is computed using the previous presented approach. In the second part, the foreground is segmented analyzing the optical flow to manage the different objects individually. In the last step of the algorithm, the depth estimation of the background is combined with the foreground information to obtain a depth prior of each frame of the video sequence, and then, this depth prior is filtered using a Cross Bilateral Filtering to enhance the edges of the final depth estimation.

Chapter 3

System overview

3.1 Introduction

In the literature, the two most relevant works for 2D-to-3D image and video conversion belonging to the machine learning-based methods are the publications of Konrad [4] and Karsch [5]. Both systems present a similar workflow, which is represented in Fig. 3.1. The main difference between them lies in the Depth Fusion block. While Karsch approach implement a SIFT flow-based algorithm, Konrad argues that this only provides marginal improvements and chooses to fuse the depth maps applying the median.

Taking this into account, we present two automatic approaches for 2D-to-3D image conversion using a machine learning framework. The overall working of this family of algorithms is shown in Figure 3.2.

Given a query color image Q , and a RGBD database DB , composed by color images and their corresponding depth maps, the goal is to estimate the depth map D_{est} of Q . Based on the key hypothesis that images with a high similarity in color will likely have a high depth similarity, the algorithm first searches in DB for those color images that are the most similar to Q . Then the depth maps associated to those color images are fused to obtain the depth map estimation D_{est} . The correct performance of the algorithm is subject to the presence of structurally similar images to the query frames Q in DB .

Based on the presented scheme, we propose a 2D-to-3D image conversion system, (see Sec. 3.2), and another method for 2D-to-3D video conversion, (see Sec. 3.3)

3.2 Proposed 2D-to-3D image conversion system

The first system proposed in this thesis is based on a machine learning scheme. To build the presented approach, several contributions have been developed over the general system represented in Figure 3.2. A more detailed diagram of the proposed system is showed in Figure 3.3.

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 .

Figure 3.1: Overall algorithm for both Konrad [4] and Karsch [5] 2D-to-3D conversion approaches [4].

The offline module is composed by two sub-modules. The first one is the Database Clustering sub-module that clusters structurally similar color images together to speed up the hierarchical search of color images in DB similar to Q (instead of checking the similarity with every image in the database, it is estimated the cluster to which the query image belongs, and then the search is reduced to images inside that cluster). For the clustering process the images are compactly represented by feature descriptors that encode their structure. Section 4.1 explains in detail the feature based image representation and how these descriptors and the database clustering affects the results of the 2D-to-3D image conversion. The second sub-module, Parameter learning, performs a learning procedure to estimate the best parameters w and k for the depth estimation process adapted to the structure of every image Q . Chapter 6 describe those parameters needed for the 2D-to-3D image conversion, and the effect of the adaptive computation in comparison to the use of fixed values.

The online module estimates the depth of Q using the clustered DB and the parameters learned from the offline module. It is composed by 3 sub-modules: Classification, Candidate Search, and Depth Estimation. The Classification sub-module estimates the cluster of images in the database to which the query image Q belongs to according to its structure. For this purpose, structural similarity between a cluster and a query image is measured via a distance function (for instance, Euclidean distance or correlation) between the feature-vector based representations of the cluster representatives and the query image (see Section 4.1 for details about image feature representation). This sub-modules allows to speed up the search of images in the database that are structurally similar to the query images, and to deal with large databases

Figure 3.2: General working diagram of machine learning-based 2D-to-3D image conversion.

Figure 3.3: Block diagram of the machine learning-based 2D-to-3D image conversion proposed approach [6].

that will be require to obtain a satisfactory depth image estimation. The second sub-module, Candidate Search, computes the k -most structurally similar color images to the query image in the previously selected database cluster. Then, the last submodule, Depth Estimation, explains how the corresponding depth maps of the selected candidate color images are combined and filtered to obtain the final depth estimation D_{est} . Chapter 5 describes in detail this sub-module.

3.3 Video conversion

In this tesis, it is also presented a system for converting both images and video sequences from 2D to 3D by extending the scheme showed in Figure 3.2. Estimating depth from video sequences instead of just from single images allows the use of movement to infer the depth scene. The movement in a video sequence may be caused by the movement of one or several objects in the scene, or by a moving camera. In the case of a moving camera, techniques such as structure from motion provide satisfactory results obtaining the 3D structure of the scene. Due to this

reason, the developed system is focused on video sequences where the camera is static but there are one or more moving objects in the scene. The overall block diagram of the presented approach can be seen in Figure 3.4

Figure 3.4: Block diagram of the proposed 2D-to-3D video conversion system [7].

In this case, given a query Video sequence V_Q , the background depth of the whole sequence and the foreground of each frame are computed in parallel. The background depth is estimated using the machine learning-based approach presented in Sec 3.2. And the foreground depth estimation is based on an optical flow-based technique. Then, both foreground and background depth estimations are combined and filtered to obtain a realistic depth estimation of the video sequence D_{est} . Chapter 5 describes in detail the 2D-3D video conversion process.

3.4 Conclusion.

In this chapter, the proposed systems for image and video depth estimation have been presented. Both methods are not independent since the image 2D-to-3D conversion algorithm is used as a sub-system in the video conversion process. In the next chapter, the image conversion system is presented in detail.

Chapter 4

2D to 3D image conversion system

4.1 Image representation and hierarchical search

In this section, it is explained the feature descriptor used to compactly represent the images. This image feature descriptor is built as a combination of several classical and widely used region feature descriptors. It is discussed how the use of this image descriptor affects the results of the proposed 2D-to-3D conversion, and how much it can improve the quality performance in comparison to the individual use of the involved feature descriptors.

It is also explained how these descriptors are used to cluster the *RGBD* database by means of k-means algorithm, and how they are also used to perform a hierarchical search of candidate images along the dataset. It is exposed how the clustering and hierarchical search affects the performance of the algorithm in terms of speed and quality compared to a non-cluster strategy that uses an exhaustive search of candidate images. A discussion about the influence of the number of clusters in which the dataset is divided into regarding the performance of the 2D-to-3D conversion system is also included.

4.1.1 Compact image representation

Features are quantifiable values that are used to describe images or regions of images from a high level perspective. They provide information about color, texture, shapes, or edges contained in images. One key property of descriptors is repeatability, this means that the same physical interest points should be detected under different viewing conditions, such as illumination changes, noise, and slight translation, rotations, and scales. Another interesting property of feature descriptors is that similar points in two or more images should have similar descriptions. The length of the descriptors is a tradeoff between discrimination capability and computational cost. Large descriptors that have a high discrimination capability have also a large dimensionality, leading to a higher computational cost. On the contrary, low dimensionality descriptors have associated a lower computational cost, but also a reduced discrimination capability. [55]

In the presented system, the images are compactly represented by a feature descriptor. This feature-based representation is based on a combination of four state-of-the-art region descriptors: Histogram of Oriented Gradients (HOG) [51], Local Binary Patterns (LBP) [52],

Figure 4.1: The feature descriptors are computed in each image tile, and then concatenated to form an image descriptor [8].

the GIST descriptor [53], and the Speeded-Up Robust Features (SURF) [54]. The procedure to obtain a feature descriptor that describes the structure of a color image is as follows. Every color image is divided into 4 x 4 tiles (as can be shown in Fig 4.1). Then, the HOG, LBP, GIST and SURF image descriptors are independently calculated for each tile, and stacked together by concatenating all the tile descriptors per type as

$$\begin{aligned}
 f_{hog} &= [hog_{1,1} \ hog_{1,2} \ ...hog_{1,N_{col}} \ ...hog_{1,N_{col}}] \\
 f_{lbp} &= [lbp_{1,1} \ lbp_{1,2} \ ...lbp_{1,N_{col}} \ ...lbp_{N_{row},N_{col}}] \\
 f_{gist} &= [gist_{1,1} \ gist_{1,2} \ ...gist_{1,N_{col}} \ ...gist_{N_{row},N_{col}}] \\
 f_{surf} &= [surf_{1,1} \ surf_{1,2} \ ...surf_{1,N_{col}} \ ...surf_{N_{row},N_{col}}]
 \end{aligned} \tag{4.1}$$

where $N_{row} = 4$ and $N_{col} = 4$ are the number of rows and columns, respectively, and the subindices of every region feature descriptor (such as $hog_{\{m,n\}}$) represents the tile coordinates. Finally, an image descriptor is obtained by concatenating the previous resulting vector as

$$f_{mix} = [f_{hog} \ f_{lbp} \ f_{gist} \ f_{surf}]. \tag{4.2}$$

4.1.2 Database clustering

The partition of the color images that compose the database DB into clusters is done using the k-means algorithm. For this purpose, the clustering is performed using the image feature-based representation described in Section 4.1.1., which captures the structural content of images. K-

means is an unsupervised learning algorithms that solves the well known clustering problem. The algorithm partitions a given dataset into a determined number (k) of clusters previously fixed. The main idea is to define one centroid for each of the k clusters. Since different locations for the centroids may lead to different clustering results, the better choice is to initially locate them as far away as possible from each other. The second step of this process is to associate each point (i.e. an image feature descriptor) belonging to the dataset to the nearest centroid. When every point of the dataset has been associated to a cluster, k new centroids are recalculated as the barycenters of the points associated to each cluster (which were calculated in the previous step). These two-steps are iterated until there is no change in the cluster centroids, or interactively until a predefined number of iterations. In other words, centroids do not move any more, and consequently every dataset point will be associated to the same centroid than the previous iteration [56]. Regarding the metric to measure distances between dataset points and centroids, the correlation function is used.

Additionally, a set of weights w are considered for each one of the four descriptors that formed the image descriptor with the purpose of assigning different relevance to each one depending on how well they capture the structure of the scene. The main reason for this fact is that depending on the type of image, one descriptor can characterize better one image than the others, and therefore it should have more relevance in the feature-based representation.

Since every feature descriptor (HOG, LBP, GIST and SURF) has different lengths, the image descriptor f_{mix} is normalized to avoid that longer descriptors have more importance in the clustering process. The normalization process is performed using the mean $\mu_{f_{mix}}$ and typical standard deviation $\sigma_{f_{mix}}$ of the set of f_{mix} obtained from the color images in DB as

$$f_{mix}^{norm} = \frac{f_{mix} - \mu_{f_{mix}}}{\sigma_{f_{mix}}}. \quad (4.3)$$

Once that the clustering is performed, the cluster centroids are stored as feature vector representatives of the set of color images that belong to every cluster. These will be used in the on-line processing module to speed up the k-Nearest Neighbor strategy used to find the most similar color image in DB to a query image.

The number of clusters $N_{cluster}$ is a critical parameter in the algorithm for two reasons. First, it determines the number of different types of considered scenarios, which has implications on the estimated depth map quality and the computational cost. A low value of $N_{cluster}$ can lead to clusters that have images from very different scenarios (i.e. their structural similarity is low), and therefore useless for a proper depth estimation. On the other hand, a large number of $N_{cluster}$ potentially generates clusters with images that are more representative of more specific scenarios in DB , but the computational cost of the posterior on-line processing module significantly increases, since it is based on a k-Nearest Neighbor search among all cluster centroids. An analysis of the influence of $N_{cluster}$ in the algorithm is performed in Sec. 6.2.

4.1.3 Candidate Search

In this section, it is explained the Candidate Search module introduced in Chapter 3. The clustering of the database DB allows the use of a hierarchical search to find the candidate images involved in the process of the depth estimation.

The first part of the hierarchical search is the coarse search, which aims to find the clusters of images whose representatives are closer to the query image Q . The diagram for this part of

Figure 4.2: Block diagram of the coarse search module [6].

the algorithm can be seen in Fig 4.2.

For this purpose, the image descriptor of Q (f_{mix_Q}) is computed (see Section 4.1.1). and then compared with every vector centroid f_{mix_c} resulted from the clustering database process. The similarity metric used to compare Q to images in DB is the correlation between the corresponding image descriptors. The following equation shows the metric computation between the image descriptor of the query image Q and a specific image in one cluster.

$$S_{clu} = \text{corr}(f_{mix_Q}, f_{mix_c}) \quad (4.4)$$

As a result, the m nearest clusters C_{match} are selected. This parameter m is a tradeoff between computational cost and accuracy in the search. While high values of m guarantee that the closest descriptors in the database are found, it also increases the computational cost. Low values of m speed-up the search, but does not ensure that the closest image descriptors in the database are found.

The second part is the fine search inside the previously selected clusters. The block diagram of the fine search part of the algorithm is shown in Fig 4.3. The first step is the estimation of the parameters w^Q and n_c^Q more suitable for our query image Q . For this purpose, the query descriptor f_{mix_Q} is compared to the descriptors $f_{mix_{im}}$ of the images in C_{match} using the correlation as the similarity metric.

$$S_{im}(i) = \text{corr}(f_{mix_Q}, f_{mix_{im}}) \quad (4.5)$$

Then, given the images in C_{match} , the similarity scores S_{im} , and the parameters w^{opt} and n_c^{opt} learned in the offline module (see Section 4.3), values w^Q and n_c^Q are computed as the weighted average of the parameters w^{opt} and n_c^{opt} for each of the images in C_{match} , where the weights are the similarity values $S_{im}(i)$. Mathematically, it is given by

Figure 4.3: Block diagram of the fine search module[8].

$$\begin{aligned}
 w^Q &= \frac{\sum_i (S(i)W(i))}{\sum_i S(i)} \\
 n_c^Q &= \frac{\sum_i (S(i)N_c(i))}{\sum_i S(i)}
 \end{aligned} \tag{4.6}$$

where w^Q are the weights to apply to the feature descriptor F_Q , n_c^Q is the number of similar images to Q that are going to be selected, $S_{im}(i)$ is the similarity score between f_Q and the descriptor of the i -th image in C_{match} , W and N_c are the set of weights w and number of candidate images n_c to select, respectively, learned in the offline process that corresponds to the images in C_{match} .

Then, an exhaustive search is performed over the images in C_{match} to find those n_c^Q images that are more structurally similar to Q . This search is applied using the correlation as the similarity metric, and weighting the descriptors with w^Q .

$$\begin{aligned}
 S_{descr} &= \text{corr}(\text{descr}_Q, \text{descr}_{im}) \\
 WS &= \sum_{descr} w_{descr} S_{descr}
 \end{aligned} \tag{4.7}$$

where $descr$ is f_{hog} , f_{lbp} , f_{gist} , or f_{surf} correspondingly, and S_{descr} is the similarity value between the classic feature descriptor of the query image f_{mix} and the descriptors of the images in C_{match} , and WS is the weighted similarity using the set of weights w^Q . As a result of this task, a set of images in DB that are the most structurally similar to Q , and their associated depth maps are obtained.

4.2 Depth estimation

In this section, it is explained how the corresponding depth maps of the selected candidate color images are combined and filtered to obtain the final depth estimation D_{est} . First, in the depth fusion process, an image registration process is applied to the candidate depth maps to get a better correspondence with the edges in the query image Q . Then, those registered depth maps are combined using a weighted average approach to build a depth prior D_{prior} . Lastly, an edge enhancement process is performed where a segmentation-guided filtering is applied to

Figure 4.4: Block diagram of the depth estimation stage. [8]

 Figure 4.5: Depth map candidates D_C are registered to better fit the edges of the query image Q , and then, combined to build the depth prior D_{prior} .

enhance D_{prior} , and obtain finally the depth map estimation D_{est} .

The depth estimation stage computes the depth map of Q using the depth maps corresponding to the selection of candidate images obtained in the previous stage (see Section 4.1.3). Fig. 4.4 shows the block diagram of this stage. The Depth Fusion and Edge Enhancement modules are described in Sec. 4.2.1 and 4.2.2 respectively.

In Sec 6.3, it is discussed how the registration task affects the quality of the final depth estimation D_{est} , as well as which the effect of the weighted average fusion is in the 2D-to-3D conversion compared to the use of a median operator, which is usually used in the state of the art. It is also discussed the effects of using this filtering compared to other filters such as cross-bilateral filtering.

4.2.1 Depth fusion

The first module of this process, the depth fusion module, is performed in two steps. First, an image registration process is performed with the k selected color images in DB . The registration is performed using the algorithm presented by Mahesh et al.[57]. This is a feature-based method that uses SIFT for detecting and matching key points, and RANSAC for the robust perspective image transformation. The registration is only performed if there can be found in the database DB images that are similar enough to the query image Q . If this condition is not satisfied, the registration task does not provide a consistent result, and then result is discarded, and just the depth candidate is used instead of the registered depth. Fig 4.5 shows the working scheme of this module.

The resulting geometric transformation is applied to the corresponding depth images D_c ,

which are then averaged to obtain the registered depth map D_{reg} .

The depth maps corresponding to the k candidate images with the highest similarity score are combined to build a depth prior estimation D_{prior} . While other approaches [5, 4] uses a median operator to combine candidate depth maps, in the presented system, this combination is performed by computing the weighted average of the selected depth maps using as weights, the weighted similarity scores previously obtained in Eq 4.7. This combination is defined as follows:

$$D_{prior} = \frac{\sum_{i=1}^k (WS(i)D_{reg}(i))}{\sum_{i=1}^k WS(i)} \quad (4.8)$$

where $D_{reg}(i)$ is the result of applying the registering process to the $i - th$ depth map selected to be fused, and $WS(i)$ is the weighted similarity score computed in Eq 4.7. As a result of this process, a prior depth estimation is obtained.

4.2.2 Segmentation-based edge enhancement

As a result of the depth fusion stage, a globally consistent depth estimate is obtained. However, this preliminary depth estimate contains local inconsistencies around the edges due to the smoothing generated by the weighted average filtering of the depth maps of the k most similar images. In order to enhance edges and align them with respect to the original edges of the query image Q , a post-processing filter is applied to obtain the final depth estimation D_{est} .

A widely used proposal for this task is the use of the cross-bilateral filtering [4], which is a variant of bilateral filtering (an edge preserving smoothing filtering) where a Gaussian smoothing function is controlled by an external intensity image [58]. Nevertheless, in the proposed system, the obtained prior depth estimation D_{prior} is refined using a new smooth filtering technique that reduces noise and artifacts, and thus improves the quality of the final depth map.

The smooth filtering is guided by a hierarchical segmentation applied to every query color image. The main idea behind this filtering is to use the image structure of the query image to refine the depth estimation, since there exists a significant correspondence between many of the edges in the color image and in their associated depth images. Fig.4.6 shows an example of a depth image, the corresponding color image, and the resulting segmentation of the color image for a better understanding. Observe the high correlation between the depth transitions and the resulting borders from the segmented color image. It can be also noted that the color segmentation includes borders that do not correspond to depth transitions, because of the different reflectance patterns that can have an object that have the same depth. The proposed smooth filtering algorithm handles this situation seamlessly.

Figure 4.7: From left to right: query color image, depth prior and depth estimation. A significant correspondence between many of the edges in the color image and the edges of depth estimation can be observed [8].

Figure 4.6: From left to right: actual depth, query color image Q , and segmentation of Q . A significant correspondence between many of the edges in the color image and in their associated depth images can be observed [6].

Let's assume that the color image is segmented. Theoretically, the depth value of the regions corresponding to each region of the segmentation/partition would be coarsely the same. Then, an adaptive average filtering is applied over the depth estimation, using a uniform kernel with a custom spatial size defined according to the shape of every color segmented region. This means that every depth pixel is smoothed using the average of values of an adaptive region of depth pixels. The shape of this region corresponds to one of the segmented regions in the query image, according to the belonging of the spatial coordinates of the depth pixel to one of the segmented region. In other words, for every segmented region in the color image, an average filter is applied to only the pixels that belong to the region that is being filtered. Those active pixels for each region are selected using a mask defined as:

$$s(x, y) = \begin{cases} 1 & \text{if } (x, y) \in R_s \\ 0 & \text{if } (x, y) \notin R_s \end{cases}, \quad (4.9)$$

where R_s is the 2D segmented region that contains the spatial coordinates of the depth pixel to be filtered.

Practically, an accurate segmentation of a color image is quite challenging. However, the proposed smooth strategy can perfectly work with over-segmented images (with more partitions that a human user could determine). Consider an object that is located at the same depth but it is split into two or more regions in the color segmented process. Then, the smoothing effect of those adaptive filters derived from those segmented regions will be very similar, since all of them will average similar depth values. On the contrary, an under-segmentation would have potentially very negative effects, since the adaptive filters would average very different depth

Figure 4.8: Diagram of the parameter learning process [8].

values belonging to different objects. For the purpose of obtaining an over-segmentation, the algorithm presented in [59] 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 using a spectral clustering framework. And then, it performs the image segmentation by converting the computed contours into a hierarchical region tree.

4.3 Parameter learning

The purpose of this section is to show the learning process followed to estimate the value of some key parameters of the 2D-to-3D conversion. The first ones are the weights w used for the weighted correlation function associated to the candidates search. Those weights are needed to give a different importance to each of the feature descriptors that compose the super-descriptor as was explained in Section 4.1. The other parameter that is learned in the presented algorithm is the number n_c of candidate images that will be used in the depth fusion stage, as detailed in Section 4.2.1. In this chapter, it is also discussed how the learning-based estimation of these parameters affect the performance of the conversion system compared to using fix values for this parameters, as is usually done in the state of the art.

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 n_c . The depth estimation error computes the dissimilarity between the estimated depth map D_{est} of Q and its real depth map DQ . 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 DQ , respectively, leaving the others pairs as training data for the estimation of D_{est} . The whole process is performed in a non-clustering strategy.

The first step in the learning stage is the computation of the super-descriptor f_{mix} as it was explained in Section 4.1.1 and 4.1.3. Fig. 4.8 shows an outline of the procedure. It starts by computing the feature descriptor f_{mix} for all the N_{Im} color images in DB . With these descriptors and the depths in DB , we carry a Leave One Out test for estimating the best values for parameters W and K , resulting in one hand, one feature descriptor $f_{mix_{test}}$ and one depth map D^{test} , and in the other hand, the descriptors of the rest of images $F_{mix_{train}}$ and D^{train} .

To find the optimum values for the weights of the descriptors w_{opt} , we compute the similarity between f_{mix}^{test} and each descriptor from the set of $N_{Im} - 1$ descriptors, F_{mix}^{train} . This similarity is calculated as a weighted correlation between them, as can be seen below for a particular descriptor f_{mix}^{train} in F_{mix}^{train} , modifying the weights w to perform this comparison from 0 to 1 with a step of 0.25 for each descriptor.

$$S_f = \sum_{i=1}^4 w(i) \text{corr}(f_{mix}^{test}, f_{mix}^{train}) \quad (4.10)$$

For each combination of weights, all values of similarities are sorted in descend mode creating the vector of similarities S_f . The depth maps in D^{train} are also sorted according to the same sequence as the descriptors (D^{sorted}) and fused creating $N_{Im} - 1$ depth map fusions D^{fusion} where the $i - th$ D^{fusion} , is the combination, in a weighted average, of the i first depth maps in D^{sorted} . The weights for this weighted average computation are the values of the vector of similarities of the feature descriptors S_f as it is shown below:

$$D^{fusion}(n) = \frac{\sum_{i=1}^n (S_f(i) D^{sorted}(i))}{\sum_{i=1}^n S_f(i)} \quad (4.11)$$

where $D^{fusion}(n)$ is the $n - th$ built depth map, $S_f(i)$ is the $i - th$ similarity value, computed as the correlation, and $D^{sorted}(i)$ is the $i - th$ most similar depth map to D^{test} . Once all D^{fusion} have been built, the correlation between them, and D^{test} is computed.

$$S_{dm}(n) = \text{corr}(D^{test}, D^{fusion}(n)) \quad (4.12)$$

This process is repeated for every combination of $w(i)$, and k and those producing the highest score S_{dm} are selected as the optimum parameters w_{opt} and k_{opt} for the test image represented by f_{mix}^{test} .

4.4 Conclusion.

In this chapter, the 2D-to-3D image conversion system has been presented in detail. It has been explained how the images are compactly represented using a mix of different common used feature descriptors, how the dataset of images is hierarchically clustered, and how the algorithm learns the optimum weights for combining several feature descriptors, as well as the optimum number of candidate database images to be used in the depth map estimation. In the next chapter, an extension of this system to be used for the video depth estimation task will be presented.

Chapter 5

2D to 3D video conversion

5.1 Introduction

In this chapter, it is presented a system for automatically convert a 2D video sequence to 3D using a machine learning framework. First, an optical flow-based segmentation of the image is performed to detect the foreground and the background of the scene. A machine learning-based algorithm is applied to generate a depth estimation of the background of the scene. Then, the background depth estimation is combined with the foreground to provide a depth prior. This prior is finally filtered to provide the final depth estimation D_{est} of the video sequence. Fig 5.1 shows the overall block diagram of this system.

5.2 System

Given a query sequence V_Q and a database DB composed by pairs of color images I and their associated depth maps D , the aim of the presented approach is the extraction of depth maps D_{est} of all the frames in V_Q . The correct performance of the algorithm is subject to the presence of structurally similar images to the query frames Q in DB .

The proposed 2D-to-3D video conversion algorithm is performed in parallel in two branches that are finally merged. In the first one, every frame in V_Q is processed by the system presented in Chapter 4 to build a depth estimation of the scene background D_{bg} . In the second branch, an optical flow analysis of the scene is used to detect and extract the moving objects in the foreground of the scene. Finally, the background and the foreground are combined to compute

Figure 5.1: Block diagram of the 2D-to-3D video conversion system [7].

Figure 5.2: Two examples of frames sequences and their optical flow results

a depth prior of the scene D_{prior} . Then, a Cross Bilateral Filtering is applied to enhance the edges and obtain the final depth estimation D_{est} .

The main differences with the approach presented in Chapter 4 ([2, 11, 12]) are the extension of the method from only images to video sequences using optical flow to detect moving objects in the scene, the independent computation of background and foreground, instead of obtaining the global depth in one step, and the computation of just one background for the whole video sequence, instead of doing it for every frame as others methods do, thus improving the computational efficiency of the algorithm.

5.3 Background depth estimation

As explained in the previous section, for every frame in V_Q , the 2D-to3D image conversion process presented in Chapter 4 is performed to obtain a depth estimation of the scene background D_{bg} for each frame in the original Video query V_Q

5.4 Optical flow and foreground segmentation

To extract the foreground from a query video sequence V_Q , we have computed the optical flow OF of the sequence as described in [60].

Then, a threshold Th is applied in such a way that only pixels with an absolute value of optical flow greater than this value are considered to be moving pixels, and a binary mask M_{fg}

is created with the pixels classified as in movement.

$$M_{fg}(x, y) = \begin{cases} 0 & \text{if } OF(x, y) < Th \\ 1 & \text{if } OF(x, y) \geq Th \end{cases} \quad (5.1)$$

With the values of OF normalized between 0 and 1, Th have been experimentally set to 0.25.

5.5 Foreground-background combination

Now, a depth prior D_{prior} is built with the information of the foreground objects M_{fg} and the previously obtained depth estimation of the background D_{bg} , as follows.

$$D_{prior}(x, y) = \begin{cases} D_{bg}(x, y) & \text{if } (x, y) \in R_s \\ \alpha M_{fg}(x, y) & \text{if } (x, y) \notin R_s \end{cases} \quad (5.2)$$

where R is the region that groups all the pixels selected as foreground and α is a parameter that allows to choose the depth of the foreground objects.

5.6 Filtering

After the combination of the foreground and the background, a globally consistent depth estimation is obtained. However, the result is too smooth and presents local inconsistencies around the edges due to the fusion of the k most similar images. In order to maintain the global result while enhancing the edges, and aligning them respect to the original edges of the query frames Q , a Cross Bilateral Filtering is applied. The reason of using this technique instead of the filter presented in Section 4.2.2 is the computational cost, which is lower for the Cross Bilateral Filter.

Cross Bilateral Filtering is a variant of bilateral filtering where the Gaussian function is controlled by an external intensity image. In this case, the query frame Q is used to control the smoothing. Moreover, Cross Bilateral Filtering reduces the noise in homogeneous areas, and align and enhance the edges of the estimated depth prior regarding to Q . Formally, it can be expressed as:

$$W(x) = \sum_y h_d(x - y) h_Q(Y(x) - Y(y)) \quad (5.3)$$

$$D_{est} = \frac{1}{W(x)} \sum_y D_c(y) h_d(x - y) h_Q(Y(x) - Y(y)),$$

where D_{est} is the final estimated depth map of a query frame Q , $h_d(x)$ and $h_Q(x)$ are Gaussian functions, and $Y(x)$ is the intensity value of pixel x in image Y . The Gaussian function over the position, $h_d(x)$ is calculated over the depth map image, while the Gaussian function over the intensity $h_Q(x)$ is computed over the query image Q . The depth map is smoothed as a result of this filtering, but preserving and enhancing the edges of the query frame.

5.7 Conclusion

In this chapter, the 2D-to-3D video conversion system has been presented. As it has been detailed, the image conversion system presented in the previous chapter is used to estimate the depth of the scene background, while the foreground is extracted using optical flow. Then, the information of both background depth estimation and the foreground segmentation is combined to get the final video depth estimation. In the next chapter, the results of the two systems presented in the previous and the current chapters are presented.

Chapter 6

Results

This chapter exposes the results obtained with the 2D-to-3D image and video algorithms, previously presented in Chapters 4 and 5. In Section 6.2 it is explained how the image compact representation used in the system improves the efficiency of the 2D-to-3D conversion system, along with the effect that the hierarchical search implementation has over the performance of the algorithm, as well as the computational time. Sec. 6.3 shows how it is combined the candidate depth images to create a depth prior and how it is converted into the final depth estimation, which helps the system to improve the result of the 2D-to-3D conversion task. In Sec. 6.4, the effect of using an approach that automatically learns the set of parameters to be used instead of using predefined values for these parameters is discussed. Finally, Sec. 6.6 compares the performance of the presented 2D-to-3D image and video conversion system with the previously existing methods in the state-of-the-art.

6.1 Quality metrics

In this section, the different metrics used to evaluate the quality of the 2D-to-3D conversion systems are presented. In the literature, the metrics frequently used are the following:

$$Rel = \frac{|D_Q(i) - D_e(i)|}{D_Q(i)} \quad (6.1)$$

$$Log_{10}error = |\log_{10}(D_Q(i)) - \log_{10}(D_e(i))| \quad (6.2)$$

$$RMSE = \sqrt{\sum_i (D_Q(i) - D_e(i))^2 / N} \quad (6.3)$$

$$PSNR = 20 \log_{10} \frac{\max(D_Q)}{RMSE} \quad (6.4)$$

where D_Q is the ground truth depth map, D_e is the estimated depth map, i refers to each pixel of the image, N is the amount of pixels in image, and $\max()$ is a function that return the maximum value.

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

Algorithm	Rel Err	Log10 Err	RMSE
Learn Depth: MRF [37]	0.530	0.198	16.7
Pointwise MRF [38]	0.458	0.149	NR
Supersixel MRF [38]	0.370	0.187	NR
Surface Layout [61]	1.423	0.320	NR
Cascade Models[62]	NR	NR	15.4
Ground Plane [63]	NR	NR	22
Semantic Labels [40]	0.375	0.148	NR
Feedback Cascades [39]	NR	NR	15.2
Theta-MRF [64]	NR	NR	15.0
Depth Transfer [5]	0.361	0.148	15.1
Depth Fusion [4]	0.432	0.187	18.3
Adaptive LBP[2]	0.628	0.184	16.8
Proposed approach [6]	0.505	0.180	14.7

Table 6.1: 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 [6].

the scene depth up to a scale factor. Other problem is that rel err and log10 err metrics cannot be computed if D_Q has some zero-value pixels. In this case, the depth range has been set to $[0; 1]$. Table 6.1 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.

To solve the problems that presented the previous metrics, Konrad et al. [4] introduced as metric for measure the quality of the conversions the normalized cross-covariance (C) between the estimated depth and the ground truth depth defined as follows:

$$C = \frac{\sum_x (D_e[x] - \mu_{D_e})(D_Q[x] - \mu_{D_Q})}{N\sigma_{D_e}\sigma_{D_Q}} \quad (6.5)$$

where N is the number of pixels in D_e and D_Q (ground-truth depth of the query image Q), μ_{D_e} and μ_{D_Q} are the empirical means of D_e and D_Q , respectively, while σ_{D_e} and σ_{D_Q} are the corresponding empirical standard deviations. The normalized cross-covariance C takes values from -1 to +1 (values close to +1 indicate that the depth maps are very similar and values close to -1 suggest they are complementary). This metric is not affected by the referred problems since it is invariant to lineal transformation

Another metric that has been used is the Structural Similarity (SSIM) originally presented in [65]. This metric has been widely used as a quality metric for images since it presents a good correlation with the subjective metrics. Nevertheless, this relation between SSIM and subjective quality for images is not assured for depth maps

6.2 Image representation and hierarchical search

In this section, it is discussed the effects of the modules of the 2D-to-3D conversion system presented in Section 4.1 in the performance of the whole algorithm. This modules have been evaluated in three different databases: The Make3D database [38], the NYU dataset [66], and the RGBD database [5]. The Make3D database is composed by 534 outdoor images with their corresponding depths maps, which have been 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. On the other hand, the NYU database is formed by 1449 color images and their associated depth maps, captured by the Kinect sensor in indoor scenarios. The resolution of both color and depth images is 640 x 480 pixels. RGBD dataset is composed by a train subset of 8430 images and a test subset of 2772 images. The resolution of both color and depth images for RGBD database is 579 x 430 pixels.

Figure 6.1: Variations in the quality level for every database and for each feature descriptor. Quality values are normalized to the quality level corresponding to the proposed image descriptor f_{mix} .

Fig. 6.1 shows the performance of the proposed image descriptor as a combination of multiple region descriptors in the computation of the 2D-to-3D conversion, and a comparison with other image descriptors that are based purely on only one region descriptor.. Every line shows the quality achieved in the depth map estimation for different used datasets. The values have been normalized by the quality value obtained by the proposed image descriptor f_{mix} . The metric used to measure the quality of the computed depth estimation is the normalized cross-covariance between this and the ground truth, as defined in Eq. 6.5

Figure 6.2: Search time as a function of the number of clusters $N_{cluster}$ in which the database is divided. Time values are normalized to the search time corresponding to $N_{cluster} = 1$. This is virtually equivalent to not dividing the database into clusters [6].

Notice that the different descriptors have different behaviors for different databases. Therefore, not the same individual feature descriptor (HOG, LBP, GIST, and SURF), is always the most suitable for every situation. Moreover as can be seen in the figure, with all of them, the obtained results are below 1, and since the values are normalized to the values obtained with the super descriptor f_{mix} the use of this descriptor provides better results than any of the other classical tested feature descriptors.

On the other hand, as discussed in Section 4.3, the number of clusters $N_{cluster}$ is an important parameter that implies a tradeoff between accuracy and computational cost. Fig 6.2, shows the variation of the search time as a function of the parameter $N_{cluster}$. Each line represents a determined database configuration: Make3D composed by 400 images, NYU composed by 1449 images, and the database resulting from the union of the previous ones (1983 images). Every line shows the total search time (coarse and fine search) as a function of the number of clusters $N_{cluster}$ in which the database is divided into. The values have been normalized by the search time corresponding to one cluster. This is virtually the same condition as not considering any cluster strategy (the search is accomplished by exhaustively searching all images). As can be observed, the search time decreases fast, and remains stable between the 10% and 20% of the search time corresponding to one cluster. Those time reduction percentages are obtained for values of $N_{cluster}$ between 25 and 100 clusters. Consequently, the clustering strategy allows to use large databases that would be unfeasible with an exhaustive search approach.

The figure also shows that the time search for the Make3D dataset starts to rise slowly when the number of clusters is higher and higher. This can be explained by the fact that a hierarchical search performed over only one cluster (equivalent to an exhaustive search inside the cluster that contains all the images) is equivalent to a hierarchical search performed over a number of clusters equal to the total number of images in the database (the search is accomplished for all

Algorithm	Make3D	NYU	Make-NYU
Depth Transfer [5]	92.67	98.65	101.39
Depth Fusion [4]	0.83	3.22	4.11
Adaptive LBP [2]	0.98	3.81	4.93
Proposed approach (online module)[8]	0.65	0.93	1.05
Proposed approach (offline module) [8]	6.9	7.2	7.3

Table 6.2: Evaluation of the computational time in seconds for different state-of-the-art algorithms and for different databases [8].

the clusters, whose number is the same as the total number of images). This increment of time for large numbers of clusters is not appreciated in the figure corresponding to the NYU dataset and the combined dataset, since they contain a larger number of images, and therefore the increment of time is produced for higher numbers of clusters than those represented in the figure. The figure also suggests that the benefit provided by the clustering and the hierarchical search would be also more significant for larger databases (in comparison with the exhaustive search strategy). The times shown in Fig. 6.2 have been obtained running a *Matlab*® implementation of the algorithms (state-of-the art and ours) in a *Intel*®.Core™ i7-4510U CPU @2.00GHz 2.6GHz.

Table 6.2 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). The times have been obtained running a *Matlab*® implementation of the algorithms (state-of-the art and ours) in a *Intel*®.Core™ i7-4510U CPU @2.00GHz 2.6GHz.

Fig 6.3 shows how the number of clusters $N_{cluster}$ in which the database is divided affects the quality of the results of the estimated depth maps. Each line represents a determined database configuration and the quality achieved in the depth map estimation as a function of the number of clusters $N_{cluster}$, in which the database is divided. The values have been normalized by the quality value corresponding to one cluster. The metric used to measure the quality of the depth estimation is again the normalized cross-covariance between the estimated depth map and the ground truth, as defined in Eq: 6.5.

Figure 6.3: Variations in the achieved quality level as a function of the number of clusters $N_{cluster}$ in which the database is divided. Quality values are normalized to the quality level corresponding to one cluster.

As can be observed, the results achieve quality values above the 95 of the value corresponding to non-clustering strategy for values of $N_{cluster}$ around 30. As seen in Fig. 6.2, for this value of $N_{cluster}$, the search time was between the 10% and 20% of the search time for a non-clustering strategy. That means, that we are able to save between the 80% and 90% of the search time of the algorithm while only decreasing the quality of the 2D-to-3D conversion less than a 5%.

Notice that for some specific configurations, the achieved quality values are lightly above the 100 of the value corresponding to non-cluster strategy. The reason is that the clustering strategy works as a filter, avoiding images in the depth fusion stage that have not a very similar depth structure to Q .

The results presented in the current section are consequence of the contributions 1 and 2 referred in Sec. 1.3

6.3 Depth estimation

This sections shows the impact that the algorithm used to create first a depth prior and then the final depth map estimation (presented Section 4.2) have in the performance of the 2D-to-3D conversion system.

The implemented algorithms have been tested in three different databases: The Make3D database [38], the NYU dataset [66], and the RGBD database [5] (See Sec. 6.2).

All color and depth images from the three databases have been resized to 320 x 240 pixels for a straightforward comparison with the previous works. The metric used to measure the

Figure 6.4: Effect of the registration task in the quality of the final depth estimation D_{est} .

quality of the depth estimation is the normalized cross-covariance between the estimated depth map and the ground truth, as defined in 6.5.

A relevant part of the depth fusion module of the proposed system is the registration step. In Fig. 6.4, it is shown how this process affects to the quality of the final depth estimation. Results are only shown for the RGBD database since it is the only one of the three used databases that has images similar enough to the query one Q . The values have been normalized by the quality value obtained for a non registration configuration. It can be appreciated an improvement when this registration is performed. The reason of this increment in the quality of the conversion is the registration task aligns the edges of the depth candidates with the edges of the query image, since color and depth images have a similar structure.

Fig. 6.5 shows how the use of the weighted average operator for the depth fusion (see Section 4.2.1) affects the performance of the 2D-to-3D conversion system. The values have been normalized by the quality value obtained using the median operator, which is the most commonly used in the state of the art [5, 4].

As can be seen, the use of the weighted-based technique for the depth fusion improves the final quality of the depth estimations for every database. These improvement are between 1 and 10 depending of the dataset configuration. The reason of this improvement is that with the proposed weighted average-based fusion, a higher importance is given to those depth maps that corresponds to the images with a higher similarity to the query image Q , avoiding that images that are not really similar to Q may damage the final result.

An important part in the presented 2D-to-3D conversion system is the edge enhancement module. Fig. 6.6 shows how different filter techniques provide different quality values for the final dept estimation D_{est} . It can be seen the comparison of not implementing any kind of edge enhancement to the depth prior D_{prior} to the use of a Cross-Bilateral filter [4] and to the implementation of the segmentation-based filter presented in Section 4.2.2. The Cross-Bilateral filter is computed as follows:

Figure 6.5: Variations in the quality level achieved in every database when using a median operator and a weighted average operator for depth fusion. Quality values are normalized to the quality level corresponding to median operator.

Figure 6.6: Variations in the quality level achieved in every database for different filters applied to D_{prior} for edge enhancement.

Figure 6.7: From left to right: Query image Q , depth prior D_{prior} , final depth estimation D_{est} , and ground truth for the NYU database [8].

$$\begin{aligned}
 W(x) &= \sum_y h_d(x - y) h_Q(Y(x) - Y(y)) \\
 D_e &= \frac{1}{W(x)} \sum_y D_c(y) h_d(x - y) h_Q(Y(x) - Y(y)),
 \end{aligned} \tag{6.6}$$

where D_e is the final estimated depth map, $h_d(x)$ and $h_Q(x)$ are Gaussian functions, and $Y(x)$ is the intensity value of pixel x in image Y . The Gaussian function over the position $h_d(x)$ is calculated over the depth map image, while the Gaussian function over the intensity $h_Q(x)$ is computed over the query image Q . As a result of this process, the depth map is generally smoothed, but preserving the edges of the query image.

As can be seen, the use of the proposed segmentation-based filter improves the result of other used techniques in the state of the art for each of the tested database configurations. Some examples of the the depth map estimations for NYU and Make3D datasets are shown in Fig.6.7 and 6.8. As can be observed, the depth prior represents a global estimation that follows the general behavior of the ground truth depth map. Then, the segmentation-based filtering enhances the edges of main objects in the scene, obtaining a final depth estimation that is consistent with the query image, and highly correlated with its depth.

The results presented in the current section are consequence of the contribution 3 referred in Sec. 1.3

Figure 6.8: From left to right: Query image Q , depth prior D_{prior} , final depth estimation D_{est} , and ground truth for the Make3D database [8].

Figure 6.9: Difference in quality achieved using a uniform mix descriptor and an adaptive one.

6.4 Parameter learning

This section presents the difference between using fixed values for parameters in the conversion system and using values for those parameters that have been previously learned by the system according to Section 4.3. The parameter learning strategy has been tested using the three databases presented in Section 6.2:

All color and depth images from both databases have been resized to 320 x 240 pixels for a straightforward comparison with the previous works. Additionally, both databases have been joined to create a third database that is used to test the behavior of our algorithm in hybrid databases containing indoor and outdoor situations. The metric used to measure the quality of the depth estimation is the normalized cross-covariance between the estimated depth map and the ground truth, as defined in 6.5.

Fig. 6.9 shows the difference in the quality level achieved using a uniform mixed descriptor, and an adaptive mixed descriptor where the different weights are estimated from previously learned data. It can be seen that, for every tested database and configuration, the adaptive strategy provides around a 5% increase in the quality of the final depth map estimation D_{est} .

Fig. 6.10 shows the difference in the quality level achieved using a fixed number of candidate images in the depth fusion step and using an adaptive number of candidate images, which is estimated from the learning data in the offline training stage. The value used in the fixed configuration is the one that provides the best quality result in average for each database configuration. It can be seen that, for every tested database and configuration, the adaptive strategy provides a gain around a 10% in the quality of the final depth map estimation D_{est} .

The reason of having better results for the adaptive version of the algorithm is that there is a combination of the number of candidate images and weights that provides the best result for every database configuration or for every image in a particular database configuration. The offline learning stage helps the algorithm to learn which are good parameters depending on each image, and this information is used in the online stage to estimate a set of values for w_{opt} and k_{opt} .

Fig. 6.11 shows how the quality achieved depends on the number k of candidate images

Figure 6.10: Difference in quality achieved using a fix number of candidate images and an adaptive one.

used in the depth fusion. The maximum value of C is achieved for $k = 30$. Nevertheless, the exact value of k is not critical since very similar values of C are obtained for a wide range of k values (in this case for $k = 20 - 50$, the value of C stays close to the maximum). The method seems robust to variations of this parameter. The figure has been created by running the Leave-One-Out Cross-Validation (LOOCV) test of the NYU dataset for each image for different values of k , and then averaging the results of the cross-covariance across all tests.

The results presented in the current section are consequence of the contributions 4 and 5 referred in Sec. 1.3

6.5 Global results for 2D to 3D image conversion

Table 6.3, 6.4, and 6.5 show, respectively, the quantitative results (averaged over all the test images) for the Make3D [38] database (in a train and test configuration of 400 training images and 134 testing images), the NYU database (in a Leave-One-Out configuration), and the database combination of both Make3D and NYU databases. As can be seen, the quality results are quite similar for all the approaches. This can be interpreted as our proposal is more computationally efficient and scalable than the others offering similar quality results. Even more, our algorithm is slightly better for some cases. Another interesting fact is that the segmentation-based post-filtering seems to be useless observing the C , PSNR, and SSIM metrics. However, it appears to improve the result from a perceptual point of view, as it was observed in Figs. 6.7, and 6.8.

6.6 2D to 3D video conversion

Finally, this section compares the results obtained with the 2D-to-3D video conversion approach presented in Ch. 5 with previous systems in the state of the art. In this case, the results show the performance of the full method, instead of the partial results presented in the previous sections for each of the different modules of the system.

Figure 6.11: Variation of the quality (C) as a function of the number k of images used in the conversion [9]

Algorithm	C	PSNR	SSIM
Depth Transfer [5]	0.69	14.56	0.84
HOG Learning Based [4]	0.61	13.44	0.79
Adaptive LBP-based [2]	0.66	14.06	0.77
Presented approach (before segmentation-based filtering) [8]	0.66	14.10	0.75
Presented approach [8]	0.67	14.15	0.81

Table 6.3: Evaluation of the state-of-the-art algorithms using the RMSE, PSNR and normalized cross-covariance (C) metrics for the Make3D (outdoor images) with 400 images for train and 134 images for test. The results are the average over the 134 test images[8].

Algorithm	C	PSNR	SSIM
Depth Transfer [5]	0.59	13.57	0.80
HOG Learning Based [4]	0.61	12.90	0.79
Adaptive LBP-based [2]	0.63	13.74	0.81
Presented approach (before segmentation-based filtering) [8]	0.63	13.90	0.78
Presented approach [8]	0.63	13.86	0.80

Table 6.4: Evaluation of the state-of-the-art algorithms using the RMSE, PSNR and normalized cross-covariance (C) metrics for the NYU (indoor images) in LOOCV configuration. The results are averaged over the 1449 images of the dataset. [8].

Algorithm	C	PSNR	SSIM
Depth Transfer [5]	0.60	12.48	0.76
HOG Learning Based [4]	0.60	13.58	0.79
Adaptive LBP-based [2]	0.62	13.81	0.81
Presented approach (before segmentation-based filtering) [8]	0.63	14.05	0.75
Presented approach [8]	0.63	14.00	0.78

Table 6.5: Evaluation of the state-of-the-art algorithms using the RMSE, PSNR and normalized cross-covariance (C) metrics for Make3D and NYU (outdoor and indoor images) in LOOCV configuration. The results are averaged over the 1983 images of both datasets.[8].

Algorithm - metric	RMSE	PSNR	C
HOG Learning Based [4][2012]	0.26	11.9	0.43
Adaptive LBP-based [2][2014]	0.24	12.6	0.55
Hierarchical Search [11][2014]	0.25	12.6	0.51
Background (Ours)	0.24	12.5	0.57
Background + Foreground (Ours)	0.23	13.2	0.60

Table 6.6: Evaluation of state-of-the-art algorithms using the RMSE, PSNR and normalized cross-covariance(C) metrics in the static camera sequences of the StereoRGBD Indoor1 database in Leave One Out configuration. The results are the average over the 6967 test images [7].

The proposed approach has been tested using a subset of the StereoRGBD Indoor1 database [5]. This subset consists in all the video sequences of the whole database where the camera is static and it is composed by 6967 color images + their associated depth maps.

The resolution of the color images and depth maps is 579×430 . Nevertheless, like Konrad [4], color and depth images have been resized to a 320×240 resolution for computational efficiency and for a straightforward comparison with with other approaches in the state of the art.

As seen in Section 6.2, The quality of the final estimation is sensible to the value of the number of clusters in which the databases are divided n_c , and the number k of depth maps fused to build the background depth estimation D_{bg} (see Section 4.2.1). We have experimentally chosen a value of $n_c = 80$ and $k = 15$ since these values result in both cases in a good trade-off between good performance and time efficiency.

To evaluate the performance of the algorithm quantitatively, tests in a leave-one-out cross-validation configuration have been performed, removing from the train database all the images belonging to the same sequence that the query image belongs. As the quality metric, we used the Root Mean Square Error (RMSE see Eq. 6.3), the (PSNR see Eq. 6.4), and the normalized cross-covariance C (see Eq. 6.5), all of them computed between the depth ground truth and the depth estimation provided by this method.

The results of our approach have been compared with the HOG-Based Depth Learning approach of Konrad et al [4], and with our previous works: adaptive LBP-based approach [2], and the hierarchical search approach [11]. Table 6.6 shows the quantitative results for StereoRGBD Indoor1 [5] database. For the proposed approach, we show the results of the generated depth estimation for just the background estimation, (without temporal information) and after adding the information provided by the optical flow analysis to better observe the effect of introducing the temporal information in the algorithm.

As can be seen, we outperform the other compared algorithms before the introduction of the temporal information in our estimation for the three used metrics (RMSE, PSNR and C). These results become even better for each metric after combining the information for both background and foreground.

In Fig. 6.12, some examples of query frames, the optical flow binary mask, background depth map, final depth map estimation and depth ground truth, generated by this method, are shown.

The results presented in the current section are consequence of the contribution 6 referred in Sec. 1.3

Figure 6.12: From left to right: Query Frame, Optical Flow binary mask, Background Depth Estimation, Final Depth Map Estimation and Depth Ground Truth for images of StereoRGBD Indoor1 database [7].

6.7 Conclusions

In this chapter, the results of the 2D-to-3D image and video conversion systems presented in Ch. 4 and 5 have been presented. It has been detailed the performance of the system for different image representation, how the quality and the processing time is affected for the clustering strategy, and how the different parameters of the system affect the final achieved quality. In the next chapter, the conclusions of the current thesis and the possible next steps to perform are exposed.

Chapter 7

Conclusions and future work

This chapter presents the contributions to the state of the art of 2D-to-3D image and video conversion task performed in Chapters 4 to 5.

7.1 Image representation and hierarchical search

In Section 4.1, we have described the following contributions regarding the state of the art: the feature descriptor used to represent the images, the database clustering, and the hierarchical search. These contributions were published in [11][6].

- Depending on the used *RGBD* database, the performance of typical feature descriptor is different. This situation makes unsuitable to use an specific feature descriptor for any situation. The use of the proposed image descriptor improves the performance of the 2D-to-3D image conversion system compared to the use of only one typical feature descriptor for every tested dataset.
- The clustering of the *RGBD* database along with a hierarchical search, instead of an exhaustive search, decreases very significantly the search time of the algorithm (between 80% and 90% of time saving), while the decrease of the quality of the depth map estimations is very slight (less than a 5%).

7.2 Depth estimation

In Section 4.2, we have explained the contributions to the state of the art concerning to the fusion of the depth map of the candidate images and the edge enhancement of the depth prior. These contributions were published in

- The depth map registration improves the final depth map estimation D_{est} when there exists in the database *DB* images that are close enough to the query image *Q*. When the algorithm cannot find images that are similar enough, this step is skipped to avoid to deteriorate the conversion. This contribution was published in [12].
- The use of a weighted average operator for combining the depth map of the candidate images leads to more accurate depth estimations D_{est} than the median operator for all the tested database configurations. The reason is that more importance are given in the depth

fusion to those images with a higher similarity to the query image Q . This contribution was published in [2].

- The implementation of a segmentation-based filtering provides depth map estimations with a higher quality values than using cross-bilateral filter. Furthermore, it provides an edge enhancement and alignment with the query image Q . This contribution was published in [8].

7.3 Parameter learning

In Section 4.3, it has been presented how the algorithm 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, and how the estimation of these parameters from the learning stage affects the quality of the depth estimation. These contributions were published in [8].

- For both cases, an improvement has been achieved when using the adaptive solution for every tested dataset. The improvement is around a 5% in the case of the number of candidate images in the depth fusion step, and around a 10% in the case of the weights of the mixed descriptor.
- The reason of this improvement is that there is not optimum values of these parameters that work properly for every case. The most suitable values of these parameter depend on the query image type, and the offline learning stage helps to estimate good parameters for the query image Q .

7.4 2D to 3D video conversion

Finally, Chapter 5 presents a novel algorithm for automatically estimate the 3D structure of a query video sequence. The approach uses a machine learning framework to globally estimate the depth of the background of the scene, while an optical flow analysis is used to extract the foreground and manage the moving objects locally. The method uses K-means to divide the database into different clusters to easily find the most similar images in the dataset, and also to extend the use of the algorithm to large databases avoiding the problem of excessively long times that present this kind of algorithms for the search stage. The query image is matched with all the cluster centroids, and the most similar images in this cluster are combined to build the depth of the background. In this background, we insert the objects in the foreground, segmented through optical flow, by highlighting the depth of the region corresponding to the foreground. The algorithm outperforms other state-of-the-art approaches while keeps the computational cost in a low level by using a hierarchical search and also by computing only one background depth estimation for each video sequence instead of one per image as others do. These contributions were published in [7, 6].

7.5 Future work

The presented work has improved the state of the art of the 2D-to-3D conversion in terms of quality, computational time, and scalability, as has been exposed in Ch. 6. Nevertheless, there

are some ways to continue improving the presented approaches which remains as future work since the stayed out of the scope of this work. Future lines of work are:

- Subjective evaluation. The metrics used in this document are all of them objective metrics. Since the objective of this algorithms should be centered in how human users feel these 3D images and videos rather than how much numerically accurate are the depth estimation regarding to the real values, subjective metrics should be introduced to measure the subjective quality results.
- The recent development of deep learning techniques, along with the increase in the available data and computation capabilities, suggest that they may be introduced to improve the current systems, taking advantage of their capabilities to optimize the whole system with end-to-end training approaches.

Nomenclature

C	Normalized cross-covariance
CBF	Cross-Bilateral Filtering
CNN	Convolutional Neural Network
CPU	Central Processing Unit
DIBR	Depth Image Based Rendering
GAN	Generative Adversarial Networks
GIST	GIST image feature descriptor from "Modeling the Shape of the Scene: A Holistic Representation of the Spatial Envelope"
HOG	Histogram of Oriented Gradients
LBP	Local Binary Pattern
LOOCV	Leave-One-Out Cross-Validation
NYU	New York University dataset
PSNR	Peak Signal to Noise Ratio
RANSAC	Random Sample Consensus
RGBD	Red - Green - Blue - Depth
RMSE	Root Mean Square Error
RNN	Recurrent Neural Network
SfM	Structure from Motion
SIFT	Scale Invariant Feature Transform
SSIM	Structural Similarity
SURF	Speeded Up Robust Features

Bibliography

- [1] J. Iglhaut, C. Cabo, S. Puliti, L. Piermattei, J. O'Connell, and J. Rosette. Structure from motion photogrammetry in forestry: a review. *Current Forestry Reports*, 5(3):155–168, Sep 2019.
- [2] J.L. Herrera, C.R. del Blanco, and N. Garcia. Learning 3D structure from 2D images using LBP features. In *IEEE International Conference on Image Processing*, pages 2022–2025, October 2014.
- [3] R. Garg, V. Kumar B.G., G. Carneiro, and I. Reid. Unsupervised cnn for single view depth estimation: Geometry to the rescue. In *Computer Vision – ECCV 2016*, pages 740–756, 2016.
- [4] 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.*, 22(9):3485–3496, Sept 2013.
- [5] K. Karsch, C. Liu, and S. Kang. Depth extraction from video using non-parametric sampling. In *Computer Vision - ECCV 2012*, volume 7576 of *Lecture Notes in Computer Science*, pages 775–788, 2012.
- [6] J. L. Herrera, C. R. del Blanco, and N. Garcia. A novel 2d to 3d video conversion system based on a machine learning approach. *IEEE Transactions on Consumer Electronics*, 62(4):429–436, 2016.
- [7] J. L. Herrera, C. R. del Blanco, and N. Garcia. Improved 2d-to-3d video conversion by fusing optical flow analysis and scene depth learning. In *2016 3DTV-Conference: The True Vision - Capture, Transmission and Display of 3D Video (3DTV-CON)*, pages 1–4, 2016.
- [8] J. L. Herrera, C. R. del Blanco, and N. Garcia. Automatic depth extraction from 2d images using a cluster-based learning framework. *IEEE Transactions on Image Processing*, 27(7):3288–3299, 2018.
- [9] J.L. Herrera, J. Konrad, C.R. del Blanco, and N. Garcia. Learning-based depth estimation from 2D images using GIST and saliency. In *IEEE International Conference on Image Processing*, September 2015.
- [10] S. Patil and P. Charles. Review on 2d-to-3d image and video conversion methods. In *2015 International Conference on Computing Communication Control and Automation*, pages 728–732, 2015.

-
- [11] J.L. Herrera, C.R. del Blanco, and N. Garcia. Fast 2D to 3D conversion using a clustering-based hierarchical search in a machine learning framework. In *IEEE 3DTV-Conference*, pages 1–4, July 2014.
 - [12] J. L. Herrera, C. R. del Blanco, and N. Garcia. Edge-based depth gradient refinement for 2d to 3d learned prior conversion. In *2015 3DTV-Conference: The True Vision - Capture, Transmission and Display of 3D Video (3DTV-CON)*, July 2015.
 - [13] M. Guttman, L. Wolf, and D. Cohen-Or. Semi-automatic stereo extraction from video footage. In *IEEE 12th International Conference on Computer Vision, 2009*, pages 136–142, Sept 2009.
 - [14] L.J. Angot, W. Huang, and K. Liu. A 2D to 3D video and image conversion technique based on a bilateral filter. In *Proc. SPIE*, volume 7526, pages 75260D–75260D–10, 2010.
 - [15] C. Cheng, C. Li, and L.e Chen. A novel 2D-to-3D conversion system using edge information. *Consumer Electronics, IEEE Transactions on*, 56(3):1739–1745, Aug 2010.
 - [16] R. Phan, R. Rzeszutek, and D. Androustos. Semi-automatic 2D to 3D image conversion using scale-space random walks and a graph cuts based depth prior. In *Image Processing (ICIP), 2011 18th IEEE International Conference on*, pages 865–868, Sept 2011.
 - [17] M. Liao, J. Gao, R. Yang, and M. Gong. Video stereolization: Combining motion analysis with user interaction. *Visualization and Computer Graphics, IEEE Transactions on*, 18(7):1079–1088, July 2012.
 - [18] Z. Zhang, S. Yin, L. Liu, and S. Wei. A real-time time-consistent 2d-to-3d video conversion system using color histogram. *IEEE Transactions on Consumer Electronics*, 61(4):524–530, 2015.
 - [19] R. Zhang, P.-S. Tsai, J.E. Cryer, and M. Shah. Shape-from-shading: a survey. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 21(8):690–706, Aug 1999.
 - [20] C. Huang, Q. Liu, and S. Yu. Regions of interest extraction from color image based on visual saliency. *The Journal of Supercomputing*, 58(1):20–33, 2011.
 - [21] P. Ji, L. Wang, D. Li, and M. Zhang. An automatic 2D to 3D conversion algorithm using multi-depth cues. In *International Conference on Audio, Language and Image Processing (ICALIP), 2012*, pages 546–550, July 2012.
 - [22] R. Szeliski and P.H.S. Torr. Geometrically constrained structure from motion: Points on planes. In Reinhard Koch and Luc Van Gool, editors, *3D Structure from Multiple Images of Large-Scale Environments*, volume 1506 of *Lecture Notes in Computer Science*, pages 171–186. Springer Berlin Heidelberg, 1998.
 - [23] M. Subbarao and G. Surya. Depth from defocus: A spatial domain approach. *International Journal of Computer Vision*, 13(3):271–294, 1994.
 - [24] L V. Gool and A. Zisserman. Automatic 3D model building from video sequences. *European Transactions on Telecommunications*, 8(4):369–378, 1997.
 - [25] K. N. Snavely. *Scene Reconstruction and Visualization from Internet Photo Collections*. PhD thesis, Seattle, WA, USA, 2009.

- [26] R.I Raguram, C. Wu, J.M. Frahm, and S. Lazebnik. Modeling and recognition of landmark image collections using iconic scene graphs. *International Journal of Computer Vision*, 95(3):213–239, 2011.
- [27] D. Crandall, A. Owens, N. Snavely, and D. Huttenlocher. Discrete-continuous optimization for large-scale structure from motion. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 3001–3008, June 2011.
- [28] G. Schindler, P. Krishnamurthy, and F. Dellaert. Line-based structure from motion for urban environments. In *Proceedings of the Third International Symposium on 3D Data Processing, Visualization, and Transmission (3DPVT'06)*, 3DPVT '06, pages 846–853, Washington, DC, USA, 2006. IEEE Computer Society.
- [29] R.T. Collins. A space-sweep approach to true multi-image matching. In *Computer Vision and Pattern Recognition, 1996. Proceedings CVPR '96, 1996 IEEE Computer Society Conference on*, pages 358–363, Jun 1996.
- [30] J. Bauer, C. Zach, K. Karner, and H. Bischof. Efficient sparse 3D reconstruction by space sweeping. In *3D Data Processing, Visualization, and Transmission, Third International Symposium on*, pages 527–534, June 2006.
- [31] H. Hirschmuller. Stereo processing by semiglobal matching and mutual information. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 30(2):328–341, Feb 2008.
- [32] J.M. Frahm, P. Fite-Georgel, D. Gallup, T. Johnson, R. Raguram, C. Wu, Y. Jen, E. Dunn, B. Clipp, S. Lazebnik, et al. Building rome on a cloudless day. In *ECCV*, pages 368–381. Springer, 2010.
- [33] C. Rother and S. Carlsson. Linear multi view reconstruction and camera recovery using a reference plane. *International Journal of Computer Vision*, 49(2-3):117–141, 2002.
- [34] J. Kosecka and W.i Zhang. Video compass. In *Proceedings of the 7th European Conference on Computer Vision-Part IV, ECCV '02*, pages 476–490, London, UK, UK, 2002. Springer-Verlag.
- [35] D. Hoiem, Alexei A. E., and M. Hebert. Automatic photo pop-up. *ACM Trans. Graph.*, 24(3):577–584, July 2005.
- [36] O. Barinova, V. Konushin, A. Yakubenko, K. Lee, H. Lim, and A. Konushin. Fast automatic single-view 3-d reconstruction of urban scenes. In David Forsyth, Philip Torr, and Andrew Zisserman, editors, *European Conference on Computer Vision ECCV 2008*, volume 5303 of *Lecture Notes in Computer Science*, pages 100–113. Springer Berlin Heidelberg, 2008.
- [37] A. Saxena, H. Chung Sung, and Y. Ng Andrew. Learning depth from single monocular images. In *NIPS 18*. MIT Press, 2005.
- [38] A. Saxena, M. Sun, and A.Y. Ng. Make3d: Learning 3D scene structure from a single still image. *IEEE Trans. on Pattern Anal. and Mach. Intell.*, 31(5):824–840, May 2009.
- [39] 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.*, 34(7):1394–1408, July 2012.

-
- [40] B. Liu, S. Gould, and D. Koller. Single image depth estimation from predicted semantic labels. In *IEEE Conf. on Comput. Vis. and Pattern Recognit., CVPR 2010.*, pages 1253–1260, June 2010.
 - [41] J. Konrad, G. Brown, M. Wang, P. Ishwar, C. Wu, and D. Mukherjee. Automatic 2D-to-3D image conversion using 3D examples from the internet. *Proc. SPIE*, 8288:82880F–82880F–12, 2012.
 - [42] J. Konrad, M. Wang, and P. Ishwar. 2D-to-3D image conversion by learning depth from examples. In *IEEE Comput. Soc. Conf. on Comput. Vis. and Pattern Recognit. Workshops (CVPRW), 2012*, pages 16–22, June 2012.
 - [43] I. Laina, C. Rupprecht, V. Belagiannis, F. Tombari, and N. Navab. Deeper depth prediction with fully convolutional residual networks. In *2016 Fourth International Conference on 3D Vision (3DV)*, pages 239–248, 2016.
 - [44] J.H. Lee, M. Heo, K.R. Kim, and C.S. Kim. Single-image depth estimation based on fourier domain analysis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.
 - [45] J. Lee and C. Kim. Monocular depth estimation using relative depth maps. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.
 - [46] H. Liu, D. Lei, Q. Zhu, H. Sui, H. Zhang, and Z. Wang. Single-image depth estimation by refined segmentation and consistency reconstruction. *Signal Processing: Image Communication*, 90:116048, 2021.
 - [47] A. CS Kumar, S. Bhandarkar, and M. Prasad. Depthnet: A recurrent neural network architecture for monocular depth prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2018.
 - [48] R. Wang S.M. Pizer and J.M. Frahm. Recurrent neural network for (un-)supervised learning of monocular video visual odometry and depth. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.
 - [49] F. Aleotti, F. Tosi, M. Poggi, and S. Mattoccia. Generative adversarial networks for unsupervised monocular depth prediction. In *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, September 2018.
 - [50] Y. Almalioglu, M. R. U. Saputra, P. P. B. d. Gusmão, A. Markham, and N. Trigoni. Ganvo: Unsupervised deep monocular visual odometry and depth estimation with generative adversarial networks. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 5474–5480, 2019.
 - [51] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 886–893 vol. 1, June 2005.
 - [52] T. Ojala, M. Pietikainen, and D. Harwood. A comparative study of texture measures with classification based on featured distributions. *Pattern Recognition*, 29(1):51 – 59, 1996.

- [53] A. Oliva and A. Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. *International Journal of Computer Vision*, 42(3):145–175, 2001.
- [54] H. Bay, A. Ess, T. Tuytelaars, and L. van Gool. Surf: Speeded up robust features. *Computer Vision and Image Understanding*, 110(3):346–359, 2008.
- [55] D. ping Tian et al. A review on image feature extraction and representation techniques. *International Journal of Multimedia and Ubiquitous Engineering*, 8(4):385–396, 2013.
- [56] J. Macqueen. Some methods for classification and analysis of multivariate observations. In *In 5-th Berkeley Symposium on Mathematical Statistics and Probability*, pages 281–297, 1967.
- [57] Mahesh and M. .. V. Subramanyam. Automatic feature based image registration using sift algorithm. In *Computing Communication Networking Technologies (ICCCNT), 2012 Third International Conference on*, pages 1–5, July 2012.
- [58] F. Durand and J. Dorsey. Fast bilateral filtering for the display of high-dynamic-range images. *ACM Trans. Graph.*, 21(3):257–266, jul 2002.
- [59] P. Arbelaez, M.I. Maire, P. Arbeláez, C. Fowlkes, J. Malik, M. Maire, C. Fowlkes, and J. Malik. Contour Detection and Hierarchical Image Segmentation. *Transactions on Pattern Recognition and Machine Intelligent*, 33(5):898–916, 2011.
- [60] C. Liu, J. Yuen, and A. Torralba. Sift flow: Dense correspondence across scenes and its applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(5):978–994, May 2011.
- [61] D. Hoiem, A. Efros, and M. Hebertl. Recovering surface layout from an image. *International Journal of Computer Vision*, 75(1):151–172, Oct 2007.
- [62] G. Heitz, S. Gould, A. Saxena, and D. Koller. Cascaded classification models: Combining models for holistic scene understanding. In D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, editors, *Advances in Neural Information Processing Systems*, volume 21, pages 641–648. Curran Associates, Inc., 2009.
- [63] A. Cherian, V. Morellas, and N. Papanikolopoulos. Accurate 3d ground plane estimation from a single image. In *2009 IEEE International Conference on Robotics and Automation*, pages 2243–2249, 2009.
- [64] C. Li, A. Kowdle, A. Saxena, and T. Chen. Toward holistic scene understanding: Feedback enabled cascaded classification models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(7):1394–1408, 2012.
- [65] Z. Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Trans. on Image Process.*, 13(4):600–612, April 2004.
- [66] N. Silberman and R. Fergus. Indoor scene segmentation using a structured light sensor. In *IEEE International Conference on Computer Vision Workshops (ICCV Workshops), 2011*, pages 601–608, Nov 2011.