College of Engineering Chengannur
Department of Computer Engineering
M. Tech. Computer Science (Image Processing)
03CS7903 Seminar II
Abstract of Proposed Seminar Topic

AUTOMATIC DEPTH EXTRACTION FROM 2D IMAGES USING A CLUSTER-BASED LEARNING FRAMEWORK

18/MCS/2019 CHN19CSIP02 Anushree Santosh

September 8, 2020

Keywords: depth extraction, 2D-to-3D conversion, machine learning, depth maps, clustering,offline processing,online processing.

Abstract

There has been a significant increase in the availability of 3D players and displays in the last years. Nonetheless, the amount of 3D content has not experimented an increment of such magnitude. To alleviate this problem, many algorithms for converting images and video from 2D to 3D have been proposed. Here, an automatic learning-based 2D- 3D image conversion approach is presented, based on the key hypothesis that color images with similar structure likely present a similar depth structure.

The presented algorithm estimates the depth of a color query image using the prior knowledge provided by a repository of color + depth images. The algorithm clusters this database attending to their structural similarity, and then creates a representative of each color-depth image cluster that will be used as prior depth map. The selection of the appropriate prior depth map corresponding to one given color query image is accomplished by comparing the structural similarity in the color domain between the query image and the database. The comparison is based on a K-Nearest Neighbor framework that uses a learning procedure to build an adaptive combination of image feature descriptors. The best correspondences determine the cluster,and in turn the associated prior depthmap. Finally, this prior estimation is enhanced through a segmentation-guided filtering that obtains the final depth map estimation. This approach has been tested using two publicly available databases, and compared with several state of the art algorithms in order to prove its efficiency.

2D-to-3D conversion process is typically performed in two main steps.The first one is the depth estimation from a given monocular image,and the second one is the Depth-Image Based Rendering(DIBR) of a new image or images to form a stereo pair,or a multi-view set of images. For this rendering step, there exists many algorithm that generate good quality results, while the depth estimation from single images is still a more challenging process. For this reason, paper focuses on inferring the depth information from 2D images.

There are two main approaches to 2D-to-3D image and video conversion depending on whether a human operator is needed or not. In the denominated semi-automatic methods, the intervention of a human operator in the process is needed to assign depth to different parts of the scene, creating a sparse depth map. Then, the sparse map is processed to create a dense depth map over the whole image or video. Alternatively,the human operator can assign a global variation to the whole scene to define a depth prior, which is then refined to make the different objects of the scene appear in the final depth map. The importance of the human actuation may vary from a small sketch that assigns depths to different regions of the scene up to an accurate delimitation of the objects in the scene in order to assign depth values to them.

In the case of automatic approaches, no human operation is needed to estimate the depth of the images or video sequences. An algorithm extracts the depth structure of the scene in an automatic way, representing an important saving in time and cost. For this purpose, many advanced depth extraction algorithms have been proposed that esti-

mate the depth structure from defocus, motion, or shading.The main problem of these methods is that they cannot be universally applied to all the images that compose a movie.

The proposed automatic 2D-3D image conversion algorithm belongs to this recent family of algorithms that adopts a machine learning philosophy, and consequently does not require a human intervention. The algorithm computes the depth map of a color image by means of a learning framework that uses a color + depth image database. First, an off-line stage processes the color + depth database to infer a set of representative depth scenarios, which will be used as prior depth maps.

This is accomplished by dividing the database into clusters using a structural similarity criterion in the color domain. The depth images relative to the color ones in each obtained cluster are combined to create a cluster representative in the depth domain (a prior depth map). Then, an on-line stage estimates the depth map of a query color image using the computed prior depth maps in the off-line stage. The best prior depth map for every color query image is selected by finding the cluster whose color images are more similar to the query image using a K-Nearest Neighbor framework.

The similarity criterion is based on distances between features vectors that encode the color structure of the query and database color images. For this purpose, a learning procedure is adopted to adaptively combine a pool of image feature descriptors. Finally,this prior estimation is refined through a segmentation-guided filtering that enhances the depth value transitions among different objects in the scene, assisted by a high level segmentation in the color image. As a result, an enhanced depth map of the color query image is obtained. This algorithm has been tested in two public and widely-used RGBD databases, along with other state-of-the-art algorithms to prove its performance.

The proposed automatic 2D-3D image conversion algorithm can be formulated as follows. 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 Dest of Q. The algorithm can be divided into two main modules: an offline processing, which adapts the DB and computes intermediate results for the posterior 2D-3D conversion process, and an online processing, which estimates the depth of each incoming query image Q shows.

The offline processing module performs two tasks. First, the database DB is clustered to group the images that represent similar scenarios. And second, some critical parameters of the system are learnt from DB to achieve the best performance in the conversion process.

The division of the color images of the database DB into clusters is done using the k-means algorithm and the correlation function. For this purpose,the clustering is performed on a feature based representation of the color images that captures the structural content. This feature based representation is based on a combination of four state-of-the-art image descriptors: Histogram of Oriented Gradients (HOG), Local Binary Patterns (LBP), the GIST descriptor, and the Speeded-Up Robust Features (SURF). The exact procedure to obtain the feature descriptor that describes the color image structure is as follows. Every color image is divided into 4 x 4 tiles. Then, the previous four image descriptors are independently calculated for each tile, and stacked together by concatenating all the tile descriptors per type.For example, hogn;m represents the HOG descriptor of the tile located in the n-th row and the m-th column. Finally, a super-descriptor is obtained by concatenating the previous resulting vectors.

Additionally, a set of weights w are considered for each one of the four descriptors with the purpose of assigning different relevance to each one according to how well they capture the structure of the scene, since one descriptor can characterize better one image than the others, and therefore it should have more relevance in the feature-based representation.Once 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 Neighbour strategy used to find the most similar color image in DB to the query image.

Regarding the second task of the off-line processing module, a learning strategy is carried out to find the optimal values of the weights w related to the different descriptors, and the number of candidate images n_c needed to generate a depth prior estimation D_{prior}.It starts by computing the feature descriptor f_{mix} for all the N_{Im} color images in DB. Then, the best parameters w and n_c are estimated from the descriptors and the depth images in DB by means of a Leave One Out learning strategy.

As a result, a feature descriptor $f_{testmix}$, a depth map D_{test}, the descriptors of the rest of images $f_{trainmix}$, and the rest of depth images D_{train} are obtained.To find the optimum values for the weights of the descriptors w_{opt}, and then compute the similarity between $f_{testmix}$ and each descriptor from the set $f_{trainmix}$ .For each combination of weights, all similarity values are sorted in descend order creating a vector of similarities. The depth maps in D_{train} are accordingly sorted, and then they are fused. The weights are computed as the similarities This process is repeated for every combination of values of w. Then, the highest score for S_d is searched and the particular values of w and $n_c$that produced this score are taken as the optimum values.

Next,The online processing of the algorithm can be divided into three main steps. The two first steps performs a hierarchical search to find similar images in DB to Q by doing a coarse search and then a fine search. The coarse search is performed to find the pre-computed cluster C_{match} in DB

that matches Q the best, using the correlation as a similarity metric. In the second step,a fine search is performed by computing the similarity between Q and all images inside C_{match}.Then, the n_c most similar images are selected. For this calculation, a weighted correlation is considered, using as a weighting value the one estimated in the training step. The value n_c of the selected images is also determined by the results of the training step.The final part of the algorithm is the fusion of the depth maps and the filtering of the result to get the final depth map estimation. With this purpose, a depth prior D_{prior} is first computed by fusing the depth maps associated to the selected images D_c. Then, a segmentation-based filtering is applied to D_{prior} to enhance the edges of the final depth estimation D_{test}.

The first step of the online part of the algorithm is a coarse search to find the clusters of images(in which the database was divided in the training stage) that best match the query image Q.As a result, the m nearest clusters C_{match} are selected. This value of the parameter is a trade-off 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 components are found.

Next in fine search ,the estimation for the best parameters of query image Q are computed by finding correlation of the query descriptor and the set of descriptors related to the images inC_{match}.Next, an exhaustive search is performed over the images in C_{match} to find those images that best match Q. This search is applied using correlation as the similarity metric and weighting the descriptors.As a result, the corresponding depth maps D_c to the selected images are obtained, which will be combined in the last part of the algorithm to estimate the final depth map.

Next in Depth Map Estimation,The depth maps of the color images with highest similarity score are fused to get a depth prior.This fusion is performed by computing the weighted average of the selected depth maps.The obtained depth estimation is refined using a new smooth filtering technique that reduces noise and artifacts. 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.

The presented approach learns how to combine different feature descriptors, as well as the number of candidate images to use from the training dataset. The algorithm also clusters the database according to their similarity to find similar images in a more efficient way. The clustering and the learning phases are implemented offline. Then, when a query image arrive for the conversion process, a hierarchical search (using the estimated clusters) is performed to find those images in the training database that are the most similar to the query image. The depth prior is built by fusing the depth maps of the structurally similar images applying a weighted average, and this prior is refined with a segmentation-based filtering to improve the contour of the objects in the scene.The approach achieves similar or higher results to the best algorithms in the state of the art, outperforming them for the most challenging cases, such as the indoor scenes and for the combination of indoor and outdoor scenes. Also, the database clustering makes the algorithm feasible when large databases are used.

References

[1] M. Wang Konrad, C. Wu P. Ishwar, and D. Mukherjee. Learning Based Automatic 2d to-3d Image and Video Conversion. *in IEEE Transaction on Image Processing*, 19(9):3485–3496 ., May 2013.

[2] S. Patil and P. Charles. Review on 2d-to-3d Image and Video Conversion Methods. In *International Conference on Computing Communication Control and Automation*, page 728–732, Feb 2015.

[3] R. Szeliski and P. Torr. *"Geometrically Constrained Structure from Motion: Points on Planes.* Morgan Kaufmann Publishers Inc. San Francisco, Second edition, 1996.