This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2017.2748458, IEEE Transactions on Multimedia



# Data Driven 2D-to-3D Video Conversion for Soccer

Kiana Calagari[1] Mohamed Elgharib[2] Piotr Didyk[3] Alexandre Kaspar[4] Wojciech Matusik[4] Mohamed Hefeeda[1]

[1]School of Computing Science, Simon Fraser University, Burnaby, BC, Canada
[2]Qatar Computing Research Institute, HBKU, Doha, Qatar
[3]MMCI, Saarland University, Saarbrücken, Germany
[4]Computer Science and Artificial Intelligence Laboratory, MIT, Cambridge, MA

***Abstract*—A wide adoption of 3D videos is hindered by the lack of high-quality 3D content. One promising solution to this problem is through data-driven 2D-to-3D video conversion. Such approaches are based on learning depth maps from a large dataset of 2D+Depth images. However, current conversion methods, while general, produce low-quality results with artifacts that are not acceptable to many viewers. We propose a novel, data-driven, method for 2D-to-3D video conversion. Our method transfers the depth gradients from a large database of 2D+Depth images. Capturing 2D+Depth databases, however, are complex and costly especially for outdoors sports games. We address this problem by creating a synthetic database from computer games and showing that this synthetic database can effectively be used to convert real videos. We propose a spatio-temporal method to ensure the smoothness of the generated depth within individual frames and across successive frames. In addition, we present an object boundary detection method customized for 2D-to-3D conversion systems, which produces clear depth boundaries for players. We implement our method and validate it by conducting user-studies that evaluate depth perception and visual comfort of the converted 3D videos. We show that our method produces high-quality 3D videos that are almost indistinguishable from videos shot by stereo cameras. In addition, our method significantly outperforms the current state-of-the-art method. For example, up to 20% improvement in the perceived depth is achieved by our method, which translates to improving the mean opinion score from Good to Excellent.**

***Index Terms*—2D-to-3D conversion, Depth estimation, 3D video**

## I. Introduction

Stereoscopic 3D (S3D) videos offer more engaging experience to viewers than traditional 2D videos, especially for sports games. Shooting sports games in 3D, however, is complex and costly, because it requires deploying and operating expensive 3D camera rigs. A more cost-effective approach is to convert regular 2D videos to 3D using automated methods. The 2D-to-3D conversion methods can also be used to convert previous events of historical importance, e.g., the previous FIFA World Cup final game. Converting 2D sports videos to high-quality 3D is, however, challenging, because of the high motion and complexity of the scenes in sports games. Current 2D-to-3D conversion methods, e.g., [32], [40], are designed for general videos and when applied to sports videos may introduce various visual artifacts that negatively impact the viewing experience of users.

In this paper, we propose a data-driven method for converting soccer 2D videos to 3D. The proposed method handles the temporal and spatial complexities of soccer videos. Unlike several previous methods, e.g., [23], [20], the proposed method is designed and optimized for sports videos and especially soccer videos. The key idea of the proposed method is to learn the depth information of a video frame from similar frames in a database of 2D+Depth soccer images. However, such databases are very costly to create, especially for outdoors sports games where depth information is harder to capture compared to indoor environments where simpler equipment (e.g., Microsoft Kinect) can be used to capture depth. In addition, sports games may contain numerous varieties of scenes and frame compositions, which requires large and diverse databases to cover. We address this problem by creating a synthetic database from computer games and showing that this synthetic database can effectively be used to convert real videos. Current computer games provide high-quality depth maps, which allows us to cost-effectively obtain a wide variety of shots from different teams, stadiums, seasons and camera angles.

The proposed method converts individual frames by dividing each into blocks and finding similar blocks in the database. It then transfers the depth gradient from the matched blocks. This, however, may not produce smooth depth within the frame and across successive frames. We present a spatio-temporal depth reconstruction method to address this problem.

We conduct extensive user studies to evaluate the performance of the proposed 2D-to-3D conversion method. In these studies, we use a diverse set of video segments and follow the ITU BT.2021 recommendations [6]. Our results show that: (i) 3D videos produced by our method are almost indistinguishable from original videos shot in 3D, (ii) our converted videos are rated Excellent by subjects, most of the time, and (iii) our method significantly outperforms the state-of-the-art method in the literature [20].

A preliminary conference version of this work appeared in [11]. The current paper extends [11] along two important aspects. First, it introduces a temporal smoothness method to control depth variations in successive frames. Second, it presents a detailed design for an object segmentation method tailored for 2D-to-3D video conversion. This method produces cleaner depth boundaries for players.

The rest of this paper is organized as follows. Section II summarizes the related works in the literature. Section III provides an overview of the proposed method. Section IV presents the proposed depth gradient based conversion method and Section V presents the object segmentation method. Section VI describes our subjective and objective evaluation, and Section VII concludes the paper.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2017.2748458, IEEE Transactions on Multimedia



## II. Related Work

Over the last few years, applications for 3D media have extended far beyond cinema and have become a significant interest to many researchers. Calagari *et al.* [12] propose a 3D streaming system that performs depth customization for a wide variety of 3D displays. Yang *et al.* [38] use the client viewing angle in a tele-immersive environment to prioritize the streaming of 3D content. Hefeeda *et al.* [16] provide content protection for 3D media. While such systems provide useful applications, the limited 3D content still remains a main bottleneck for the adoption of 3D technology. To tackle this issue, 2D-to-3D conversion techniques can be used. 2D-to-3D conversion has been explored by many researchers. However, previous methods are either semi-automatic [34], [41], [14], [26] or cannot handle complex motions [32], [21], [36], [23], [17], [20], [7], [40]. To the best of our knowledge, there has not been a 2D-to-3D conversion technique that is capable of handling the complex motions and the variety of scene structures that exist in soccer videos.

In 2D-to-3D conversion, the depth map of an image is estimated. Stereo image pairs can then be synthesised using this depth information. Traditional computer vision approaches such as depth from defocus or structure from motion can be used to compute the depth maps. Park *et al.* [32] estimate the depth using structure from motion. Zhang *et al.* [41], [40] propose a 2D-to-3D conversion system based on multiple depth cues including motion and defocus. A survey on automatic 2D-to-3D conversion techniques and depth cues can be found in [39]. In several of the previous works, strong assumptions are made on the depth distribution within a given scene. For example, the work in [21] classifies shots into long shots and other shots (e.g., medium shots, close-ups, etc.), where long shots are for large field view. Long shots are assigned a depth ramp for the field and a constant depth for the players. Similarly in [36], players are detected and a constant depth is assigned to them. This, however, causes the well-known 'card-board effect' where supposedly 3D objects appear flat on the screen.

Data-driven methods are an alternative way of computing depth maps. A relatively coarse depth estimation is provided in Hoiem *et al.* [17], where a scene is segmented into planar regions, and an orientation is assigned to each region. Konrad *et al.* [23], [22] use a database of image and depth map pairs to infer the depth of an input image. Their work is designed for still images and assumes that images with similar gradient-based features tend to have a similar depth. For a query image, the depth is estimated as the median over the depths of the most similar images from the database. In [22], geometrical differences between the query and a candidate match is compensated through SIFT-flow [28]. Karsch *et al.* [20] extended this approach to image sequences. They also use a large database of image and depth map pairs. Similar to [22], for a query frame, they find the most similar images in the database and warp the retrieved images to the query image using SIFT-flow. Finally, to estimate the final depth, the warped depth maps are combined by optimizing a cost function with spatial regularization in mind. The work in [20] is the closest to ours and we compare against it.

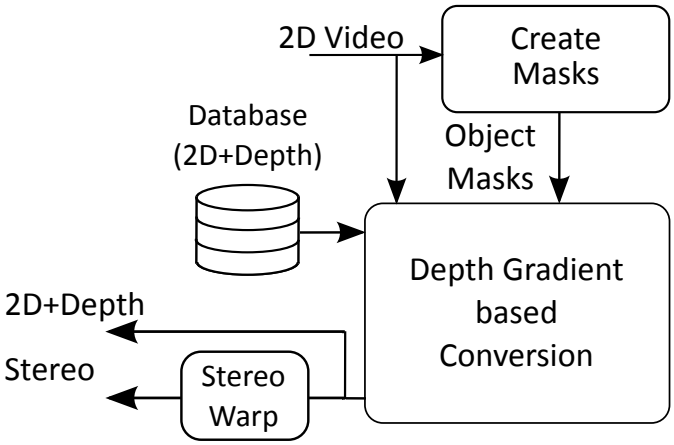



Fig. 1: The proposed 2D-to-3D conversion method.

There are a few commercial products that provide automated 2D-to-3D conversion, sold as stand-alone boxes (e.g., JVC's IF-2D3D1 Stereoscopic Image Processor, 3D Bee), or software packages (e.g., DDD's TriDef 3D). While the details of these systems are not publicly known, their depth quality is still an outstanding issue [39].

## III. System overview

An overview of the proposed 2D-to-3D conversion method is shown in Fig. 1. We infer depth from a database of synthetically generated depths. We collect this database from video games. With the high quality of current video games, which has come close to that of real videos, using a synthetic database offers two main advantages: 1) we can obtain a diverse database from different camera angles, teams, and stadiums; and 2) we can obtain accurate depth maps with perfect discontinuities. We discuss our synthetic database in Sec. IV-A.

For each query image, we transfer the depth gradients from the synthetic database to the query image by dividing the query into blocks and copying the depth gradients from the matching blocks in the database. This is quite different from previous approaches that use absolute depth over the whole frame [23], [20]. Our approach offers finer depth assignment to smaller objects (e.g., players), while requiring a much smaller database. This is because we match small blocks instead of the whole frame, and blocks have much less variety than frames.

After the depth gradients have been transfered, we recover the depth from these gradients by using Poisson reconstruction. Poisson reconstruction is a robust technique traditionally used to recover an image from its gradient information by solving a Poisson equation [33], [8]. We enhance the Poisson reconstruction formulation such that it utilizes temporal gradients in addition to spatial gradients. Our spatio-temporal Poisson reconstruction enables the generation of temporally smooth depth maps. Our depth estimation method is discussed in Sec. IV.

In order to maintain clear object boundaries, we create object masks and allow depth discontinuities on object boundaries by modifying the Poisson equation. We present two different methods for creating object masks, one for close-up shots and

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2017.2748458, IEEE Transactions on Multimedia



the other for non close-ups. In order to distinguish these two types of shots, we implemented a simple shot classification method. Sec. V discusses our object mask creation methods.

Finally, we use the stereo-warping technique in [20] to render the left and right stereo pairs using the 2D frames and their estimated depth. In this technique, a 2D frame is warped based on its estimated depth such that salient regions remain unmodified, while background areas are stretched to fill unoccluded regions.

## IV. Gradient-based Conversion

The core of our system is depth estimation from depth gradients; for an input 2D video, depth is inferred from our synthetic database. Fig. 2 outlines this process. For a 2D query frame, we first search the database for the $K$ nearest frames. Using these $K$ candidates we create a matching image block by block, where for each block we choose the best matching block from the $K$ candidates. We then copy the depth gradients from the matched blocks to the query frame. Finally the depth is reconstructed from these copied gradients by solving a Poisson equation. We now discuss each step in more detail.

### A. Synthetic Database

Many databases of RGBD (Red, Green, Blue and Depth) images [2], [1], [5] and videos [20], [3] have been created. The depth channel is acquired using techniques such as time-of-flight imaging [35] (e.g., using Microsoft Kinect). However, none of the current RGBD databases can be used for sports events. Acquiring depth maps for sports events is challenging since it requires the depth to be captured in sunlight conditions and in a highly dynamic environment. In order to address this challenge, we propose to create a Synthetic RGBD (S-RGBD) database from video games. Current video games have very high image quality and a large quantity of content can be easily generated from them.

To collect our S-RGBD data we use PIX [4], a Microsoft Directx tool, to extract image and depth information from the FIFA13 video game. PIX records all Directx commands called by an application. Each recorded frame can be rendered and saved by re-running these commands. In addition, PIX allows access to the depth buffer of each rendered frame. We extracted 16,500 2D+Depth frames from $40$ different sequences. Each sequence has a frame rate of about $10$ fps, and each extracted frame has a resolution of $1916 \times 1054$. These $40$ sequences cover a wide range of shots that can occur in a soccer match, including a variety of camera angles, color variations and motion complexities. Two of the $40$ sequences are designed to capture the common scenes throughout a full game. Each one has a duration of $6-7$ minutes. The rest of the sequences are shorter ($15-60$ seconds) and focus on capturing special and less common events such as behind the goal, close-ups, and zoomed on ground views. Our database includes different stadiums, teams, seasons and camera views.

### B. Block-based Matching

For each frame of the examined video we first identify the $K$ ($= 10$ in our work) most similar frames in our S-RGBD database by preforming visual search. The two main features used for visual search are: GIST [31] and Color. The former favors matches with overall similar structure, while the latter favors matches with overall similar color. For color, we use the hue channel in the HSV color space and create a normalized histogram of hue values with six equal-width bins. We then apply binary thresholding to represent only dominant colors (those with hue below 0.1 are ignored), and concatenate GIST and the thresholded color histogram to form the final image search descriptor. Fig. 3(b) shows $4$ samples of the $K$ candidates for the frame in Fig. 3(a).

Using the $K$ candidate images we construct a matched image, which is an image similar to the examined frame. The matched image provides a mapping between the examined frame and the candidates, where each pixel in the examined frame is mapped to a corresponding candidate pixel. While such mapping can be performed using a global approach by warping the candidates to the examined frame, such as [20], this requires strong similarity between the examined frame and the database. For example, an examined frame with 4 players requires the database to have an similar image. Therefore, we use a local approach instead, where similar images are constructed using block matching. This provides a more robust matching. For example, a good matching can be performed between two images even if they are acquired from different angles and different locations, and have a different number of players. This can be seen in the example in Fig. 3 where the images in Fig. 3(b) were used to create the high-quality matched image in Fig. 3(c), which may not have been possible using the global approach in [20]. One of the advantages of our local approach is that it can achieve good results without requiring a massive database, which is highly desirable since, as discussed in Sec. IV-A, creating an accurate 3D database is difficult.

For constructing the matching image, the examined frame is first divided into $n \times n$ blocks. In all of our experiments, we set $n$ to $9$ pixels. Each block of the examined frame is then compared against all blocks in the $K$ candidate images. We compare blocks based on their block descriptors. The block whose block descriptor has the least Euclidean distance with that of the examined block is chosen as the corresponding block. For block descriptor, we concatenate the SIFT descriptor calculated for the center of the block with the average color of the block. The average color is a three-dimension vector containing the average of R, G and B color channels separately. Note that the candidate images are all re-sized to the examined frame size. RGB values are normalized between 0-1. To capture more representative texture, the SIFT descriptor for each block is calculated on a larger patch of size $5n \times 5n$. Fig. 3(c) shows the matched image using our block matching approach. Notice that the horizontal playing field is matched with the horizontal field, the vertical advertisement boards are matched with vertical blocks, and the tilted audience are also matched with the audience.

Note that for a faster matching, we compute the image search descriptors, the block descriptors, and the depth map gradients for all frames in the database beforehand and store them as the database. Therefore, in practice, there is no need to actually store the RGB frames and depth maps of the database frames,

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2017.2748458, IEEE Transactions on Multimedia



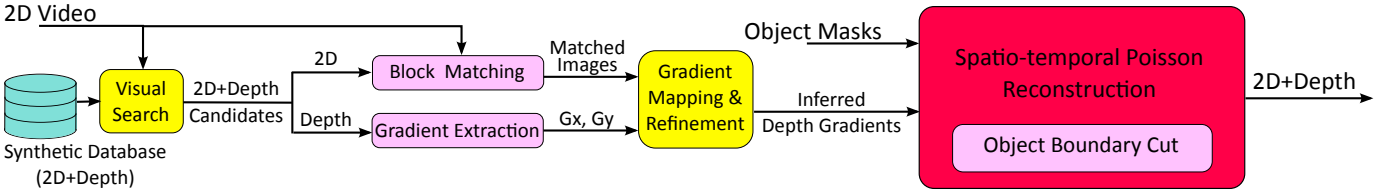



Fig. 2: The main components of the data-driven depth estimation method: For a 2D query frame, we first search the synthetic database for the nearest frames. Using these candidates we choose the best matching block for each query block. We then copy the depth gradients from the matched blocks and reconstruct depth from these gradients using a spatio-temporal Poisson formulation.

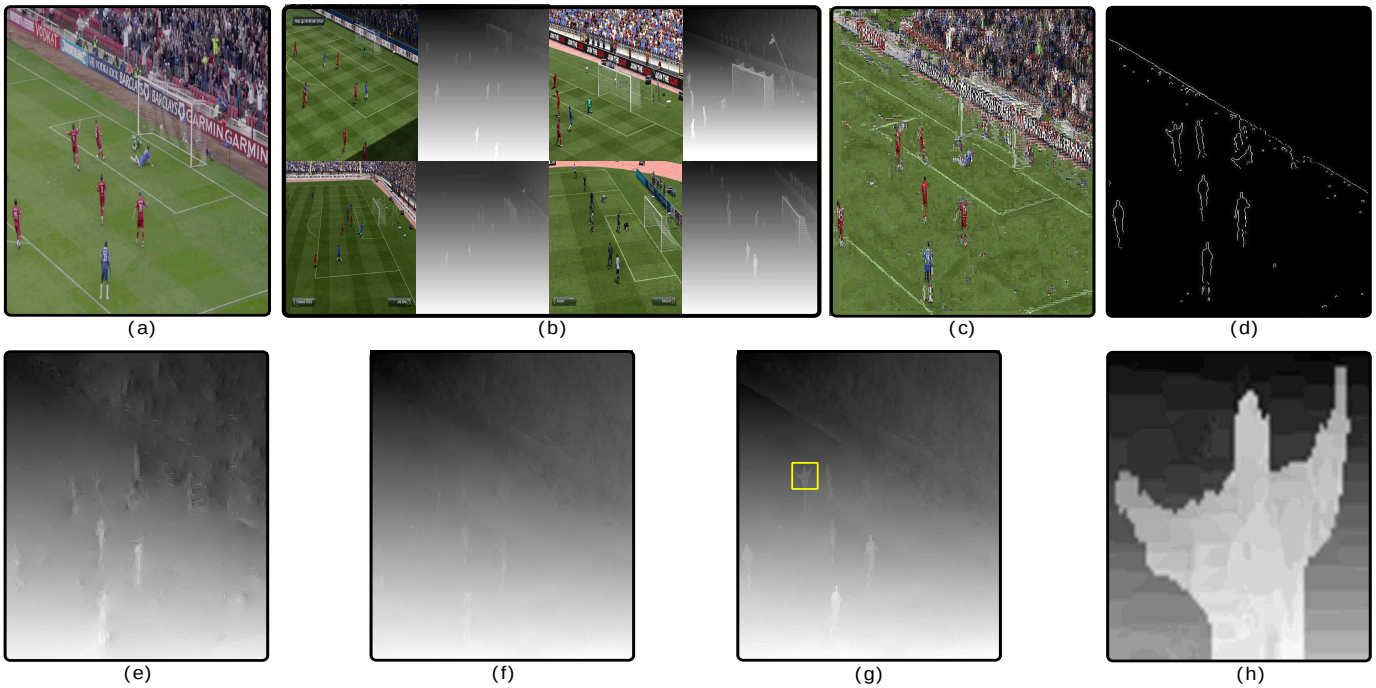



Fig. 3: The effect of different steps in the proposed depth estimation method: (a) Query, (b) Subset of $K$ matching candidates, (c) Created matched image, (d) Object boundary cuts, (e) Depth estimation using Poisson reconstruction, (f) Effect of gradient refinement, (g) Final depth with object boundary cuts, and (h) A zoomed and amplified version of the yellow block in $g$.

which saves a considerable amount of storage, in addition to reducing the processing time.

### C. Spatio-temporal Poisson Depth Estimation

To produce a smooth depth within and across all frames, we first copy depth gradients from the matched image to the query frame. We then use Poisson reconstruction to estimate the depth map from these copied gradients. However, in order to have a depth that is smooth through time and space we extend the Poisson reconstruction technique to a spatio-temporal formulation. In addition, gradient refinement and object boundary cuts are used to reduce artifacts and maintain clear depth discontinuities, respectively.

**Computing Depth Gradients:** Given a query frame and its matched image, we copy the corresponding depth gradients in blocks of $n \times n$ pixels from the matched image to the query frame. By depth gradients we refer to the first order spatial derivatives of the depth for both horizontal and vertical directions $(G_x, G_y)$.

**Poisson Reconstruction:** We reconstruct the depth values from the copied depth gradients using the Poisson equation:

$$\Big(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}\Big) D = \nabla \cdot G, \tag{1}$$

where $G = (G_x, G_y)$ is the copied depth gradient and $D$ is the depth we seek to estimate. $\nabla \cdot G$ is the divergence of $G$:

$$\nabla \cdot G = \Big(\frac{\partial G_x}{\partial x} + \frac{\partial G_y}{\partial y}\Big). \tag{2}$$

In the discrete domain, Eq. (1) and Eq. (2) become Eq. (3) and Eq. (4), respectively:

$$\begin{aligned} D(i, j+1) + D(i, j-1) - 4D(i, j) + \\ D(i+1, j) + D(i-1, j) = \nabla \cdot G(i, j). \end{aligned} \tag{3}$$

$$\begin{aligned} \nabla \cdot G(i, j) = G_x(i, j) - G_x(i, j-1) + \\ G_y(i, j) - G_y(i-1, j). \end{aligned} \tag{4}$$

To estimate $D$, we formulate the problem in the form of $Ax = b$, where $b = \nabla \cdot G$, $x = D$, and $A$ stores the coefficients

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2017.2748458, IEEE Transactions on Multimedia



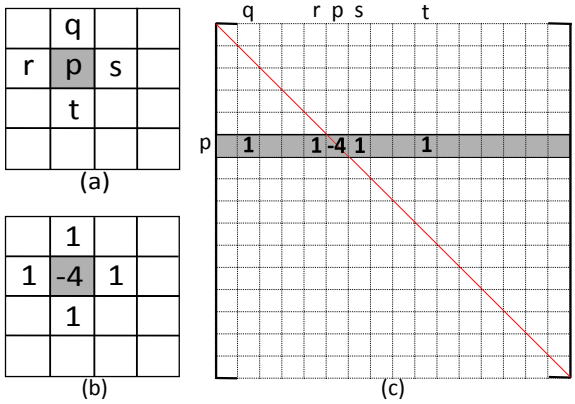



Fig. 4: Construction of matrix $A$ of the Poisson equation. (a) An example $4 \times 4$ image, showing a sample pixel $p$ and its neighbors. (b) The coefficients of Eq. (3) for pixel $p$. (c) The non-zero values in $A$ for the row corresponding to pixel $p$.

of the Poisson equation (Eq. (3)). For a query image of size $H \times W$, $A$ is a square matrix with size $HW \times HW$. Each row in $A$ corresponds to a pixel in the query frame, and the values in the row correspond to the coefficients of Eq. (3). Fig. 4 illustrates setting up $A$ for a small sample image. Note that since one or more neighbors do not exist for the image boundary pixels, the value of $\nabla \cdot G$ in these pixels is updated by removing the terms in Eq. (4) that refer to non-existing neighbors. Finally, given $Ax = b$, we solve for $x$. An example of the reconstructed depth ($x$) is shown in Fig. 3(e). In can be seen that the overall depth structure is captured, however, there are some artifacts present (see the lower right corner of Fig. 3(e)). Such artifacts are often caused by the inaccuracy in SIFT matching. For example, in Fig. 3(c) some field blocks are matched to non-field areas. If a query block from a region that is expected to have smooth depth (such as the field) is incorrectly matched to a reference block that has sharp changes in depth (such as player borders or the goal), small artifacts in the depth map can occur due to the sharp gradients that were transferred from the reference block. To avoid this problem, we perform gradient refinement, which reduces the large gradients before solving for $x$. Then using our object masks we impose depth discontinuities in the proper places. We describe these two steps in the following.

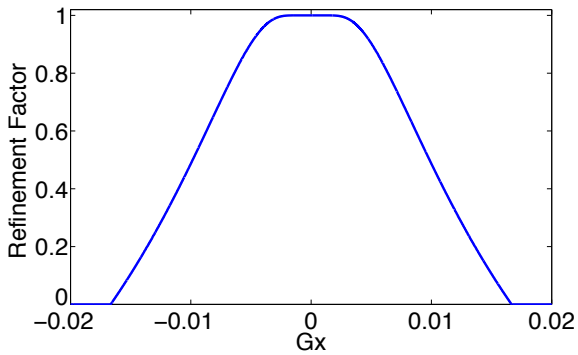



Fig. 5: The refinement factor of $G_x$ for $\alpha = 60$.

**Gradient Refinement:** To reduce the errors caused by incorrect block matchings, we multiply the depth gradients by a refinement factor:

$$G_x = G_x \times \max(\, 1 - e^{(1-\frac{1}{\alpha\,|G_x|})},\, 0\,)$$
$$G_y = G_y \times \max(\, 1 - e^{(1-\frac{1}{\alpha\,|G_y|})},\, 0\,) \quad (5)$$

This refinement exponentially reduces large gradients, which may be incorrectly estimated, while maintaining low gradients. Thus, it removes sharp artifacts while maintaining the rest of the image intact. The refinement strength is configured by the parameter $\alpha$. A high $\alpha$ can corrupt correct gradients, while a low $\alpha$ can allow artifacts. In our experiments, we set $\alpha$ to 60. Fig. 5 shows the refinement factor for $G_x$ when $\alpha$ is set to 60. It can be seen that while the factor is 1 for small values of $G_x$, it drops to zero as the gradient starts to grow. Fig. 3(f) shows the effect of gradient refinement on depth estimation for Fig. 3(a). In comparison to Fig. 3(e), artifacts are removed and depth is smoother.

**Object Boundary Cuts:** When performing Poisson reconstruction each pixel is connected to all its neighbors. This causes fading of most object boundaries, especially after gradient refinement where strong gradients are eliminated (see Fig. 3(f)). We solve this issue by modifying the Poisson equation on object boundaries and allowing depth discontinuities. To do so, we use object masks, whose creation is discussed in Sec. V. Given object masks, we first use the Canny edge detector to detect edges (see Fig. 3(d)). We then disconnect pixels from the object boundaries by preventing them from using an object boundary pixel as a valid neighbor. For each pixel neighboring a boundary pixel, the corresponding connection in $A$ is set to 0 and its $\nabla \cdot G$ value is updated accordingly. Hence, pixels adjacent to object boundaries are treated similar to image boundary pixels.

The object boundaries generated for Fig. 3(a) are shown in Fig. 3(d). The final estimated depth when cutting the object boundaries is shown in Fig. 3(g). The players in Fig. 3(g) are more visible compared to Fig. 3(f).

**Spatio-temporal Poisson Reconstruction:** While the discussed Poisson reconstruction technique produces plausible results, one of its main limitations is that it does not account for temporal smoothness. If the depth estimation is performed independently for each frame, the generated depth maps are not temporally smooth and can vary significantly between consecutive frames causing a flickering effect. While this limitation can be partially handled by temporally smoothing the depth maps during a post-processing phase, it is much more effective if we eliminate the problem from the source, and enforce temporal smoothness during the core depth estimation process. In order to do so, we enhance the Poisson reconstruction formulation such that it utilizes temporal gradients in addition to spatial gradients when reconstructing the depth. That is, instead of computing the depth of each frame independently, the information from the next and previous frames is also considered.

One of the main challenges, however, is utilizing this temporal information in the depth estimation process without limiting its parallelizable feature. Being parallelizable is an important aspect of our method, which enables processing different frames in parallel due to their independence. Considering temporal

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2017.2748458, IEEE Transactions on Multimedia



information, however, introduces dependence among frames. Therefore, in order to maintain the parallelizable feature, we determine a window around each frame and process each window independently. Within each window the depth maps of all frames are generated together and coherently. The final depth map for each frame is the average of all depth maps generated for that frame in different windows. While a bigger window size can achieve an overall better temporal coherence, it will significantly increase the computational complexity and decrease efficiency. Our experiments in Sec. VI-C show that a window size of 3 (one frame before and one after) yields good results, and not much gain can be achieved by further increasing the window size.

For each window, we perform block-based matching, depth gradient mapping and refinement for all frames. We then enforce temporal smoothness by modifying Eq. (3) as in Eq. (6) for each of the frames within the window. In Eq. (6), $D_{next}$ and $D_{pre}$ refer to the next and previous frames respectively, and $(i_c, j_c)$ refers to the corresponding pixel in the neighbouring frame. In order to identify the corresponding pixels between each two consecutive frames, we use optical flow [27], which computes the horizontal and vertical displacements for all pixels. For the first and last frames in the window for which one of the neighbours does not exist, the non-existing connection will be removed.

$$\begin{aligned} D(i,j+1) + D(i,j-1) - 6D(i,j) + \\ D_{next}(i_c,j_c) + D_{pre}(i_c,j_c) + \\ D(i+1,j) + D(i-1,j) = \nabla \cdot G(i,j). \end{aligned} \tag{6}$$

Temporal smoothness implies that the depth value of each pixel and its corresponding pixels in the next and previous frames should be similar. In other words, temporal smoothness implies that the temporal gradient should be set to zero. As a result, while the left hand side of Eq. (6) is an extension of Eq. (3) which includes temporal neighbours in addition to the spatial ones, $\nabla \cdot G(i,j)$ is still calculated using Eq. (4) which includes only the spatial gradients.

When formulating a solution in the form of $Ax = b$, we generate the matrix $A$ according to Eq. (6) such that it contains all frames in the window. Thus the size of $A$ will be $HWN \times HWN$, where $N$ is the window size. Finally, we concatenate $b$ and $x$ for all frames in the window, and solve for the depth maps $(x)$.

Note that since neither the optical flow nor the object masks are perfect, there is a chance that a pixel marked as an object (according to the mask) is recognized as a corresponding pixel to a non-object pixel in the neighbouring frame or vice versa. Establishing such temporal connections can cause fading of the object boundaries, as the two sides of the boundary will be connected through a temporal route. In order to solve this problem we first make sure that each two corresponding pixels have the same mask value before establishing a connection between them. Otherwise, we would remove the temporal connection by setting the corresponding connection in $A$ to 0.

**Creating the Final Output:** To form the final converted 2D+Depth output, we normalize the estimated depth maps in each window between $(0, 255)$ collectively and combine them with the query images. Our method produces a smooth depth that correctly resembles the depth of the players, field and spectators. Furthermore, the 'card-board effect', where a constant depth is assigned to each player, does not occur in our method. We show this by zooming-in on a player depth block in Fig. 3(g) and amplifying it by normalizing the depth values of the block to the range of $(0, 255)$. The zoomed and amplified version of the yellow marked block in Fig. 3(g) is shown in Fig. 3(h). The player in the marked block demonstrates the strength of our gradient-based approach in estimating small depth details. It can be seen that different body parts of the player have different depth values.

## V. Object Mask Creation

In order to have clear depth discontinuities on player boundaries, we delineate object boundaries. If object boundaries are not specified, the depth of players will blend with the ground, causing degradation in the depth quality. To detect object boundaries we first create object masks. These masks are created automatically in a pre-processing step where motion and appearance are used to detect objects. While object segmentation for videos with simple motion or static scenes can be performed using methods such as [19], it is rather challenging for videos with complex motion. Therefore, we propose two different methods for object detection: one for close-ups, and another for non close-ups. Close-ups are characterized by small playing areas and large player sizes, while non close-ups usually have a larger field of view. As a result, a shot classification step is required prior to object detection. Shot classification takes an input image sequence, finds the shot transitions and classifies each shot as either close-up or non close-up. Based on the type of shot the appropriate object detection method is then applied. The method for non close-ups is mainly based on global features such as the color of playing field, while for close-ups, local features such as feature point trajectories [29] are used. In this section we discuss each step in details.

### A. Shot Classification

Our shot classification stage has two main components: shot transition detection and shot classification. In shot transition detection, an input sequence is segmented into different shots by detecting the shot transitions. While there are several sophisticated techniques for handling shot transition detection [24], we designed a simple shot transition detection step suitable for our 2D-to-3D conversion method. Our implementation is designed to detect temporal impulsive changes in the frame structure. We predict each frame from its next frame using optical flow [27] and then estimate the global structure similarity between the original and predicted frame using SSIM (Structural Similarity) [37]. A frame is flagged as a shot transition if: the global SSIM value is smaller than a certain threshold (0.7 in our experiments), and it increases in the next frame by at least 0.1. In other words, if the similarity between the predicted and original frames is low but it increases considerably as we move to the next frame, then there is a high chance that there is a shot transition.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2017.2748458, IEEE Transactions on Multimedia



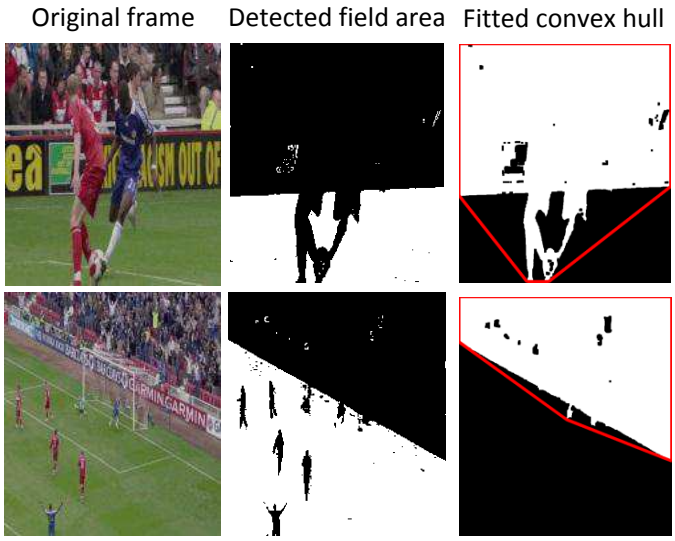



Fig. 6: Shot classification: An example of the detected field area, and the area covered by the fitted convex hull, for a close-up (top row) and non close-up (bottom row) frame. Red lines show the boundaries of the fitted convex hulls.

The second step is to differentiate between two types of shots: 1) close-ups and 2) non close-ups. A close-up is defined as a shot with a small field area and large players. We use a color-based approach to detect the field area. We train a Gaussian Mixture Model (GMM) [9] on samples collected from the playing fields and the white lines. In the test phase we estimate the log-likelihood of each pixel being generated by the learned GMM model. If the log likelihood is more than a threshold ($-15$ in our experiments), it is flagged as field area. The second discriminative cue for close-ups is player size. We exploit the observation that in close-ups, players often have a large size and the audience/ad banners are usually behind the player upper-body. Hence to measure the players size, first we invert the detected field so that white pixels indicate the non-field area. Then for each connected component, we fit a minimum convex hull and choose the largest one as the fitted convex hull for that frame. For closeups, the fitted convex hull takes a large portion of the field area due to its large player size. This does not happen in non close-ups. Fig. 6 shows an example for a close-up and a non close-up frame. The red lines show the boundaries of the fitted convex hulls. Finally, for the entire shot, we find the percentage of pixels detected as field ($A_1$) and the percentage of pixels covered by the fitted convex hulls ($A_2$). A close-up is then detected if $0.5(1-A_1)+0.5A_2$ is larger than a certain threshold ($0.3$ in our experiments).

Note that player segmentation is an important step in our shot classification. Player segmentation techniques require moderate color contrast between the foreground and background, which is the field in here. This was the case in the examined sequences and hence we did not experience much problems during shot classification. In addition, our selection of a shot classification threshold of 0.3 helped us in mitigating possible problems. One way to address the limitations of low players' color contrast is to incorporate structural information as silhouettes. This can be achieved by benefiting from the latest segmentation techniques through deep learning [18]. Another option is to train a CNN solution to directly classify the shots by implicitly learning deep features [15].

### B. Object Detection for Non Close-up Shots

Object masks for non close-up shots are a fusion of background subtraction and non-field areas. The latter is estimated during the shot classification step. However, relying only on field detection to detect players can have a high miss rate. This is often the case for players of similar color to the field. Hence to generate a more complete detection, we fuse the field detection results with that of background subtraction. Background subtraction is a well-known technique in video processing [20]. In this technique, first a homography is generated by warping all frames with respect to a reference frame (in our case it is the first frame). We use the method by Odobez *et al.* [30] with affine motion modelling to build our homography. This technique can accommodate a moderate amount of translational camera motion. The stationary background is then detected using temporal median filtering. Frame differencing between each frame and the stationary background is used to find the moving objects, which in our case are the players.

In order to further reduce player segmentation errors, we correct for possible misalignments between the frames and the stationary background through optical flow [27]. We perform frame differencing using the local SSIM values for each pixel and the motion computed by optical flow. A pixel is flagged as a moving object if: 1) the similarity between that pixel in the frame and the stationary background is low (SSIM less than $0.4$ in our experiments), and 2) the motion divergence in that pixel is high (higher than $0.01$ in our experiments). Motion divergence measures the rate of spatial changes in motion [13]. Hence it is low in regions with high similarity (such as the field) and high otherwise (for players). The final object mask is generated by a logical OR between the field detection based and background subtraction based approaches. Note that accurate background subtraction requires regular update of the underlying background model. However, we found through experimentation that this is more problematic when the examined scene undergoes strong global illumination variation such as an abrupt change in weather or lighting. This, however, was hardly the case in the processed sequences.

Fig. 7 shows an example of a non close-up object mask creation. Note that some detected players are small when using the field-only method, but are fully detected using the background subtraction approach (green boxes). The opposite is true as well (yellow box). The fusion of both approaches nevertheless brings the best of both worlds with all players being fully detected. Note that the use of multiple cues with conservative thresholds during background subtraction reduces the possibility of ghosting artifacts. Such conservative thresholds, however, could lead to players' under-segmentation. Nevertheless, when fused with the results of player segmentation with color thresholding (i.e. field detection), more complete player masks are generated.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2017.2748458, IEEE Transactions on Multimedia



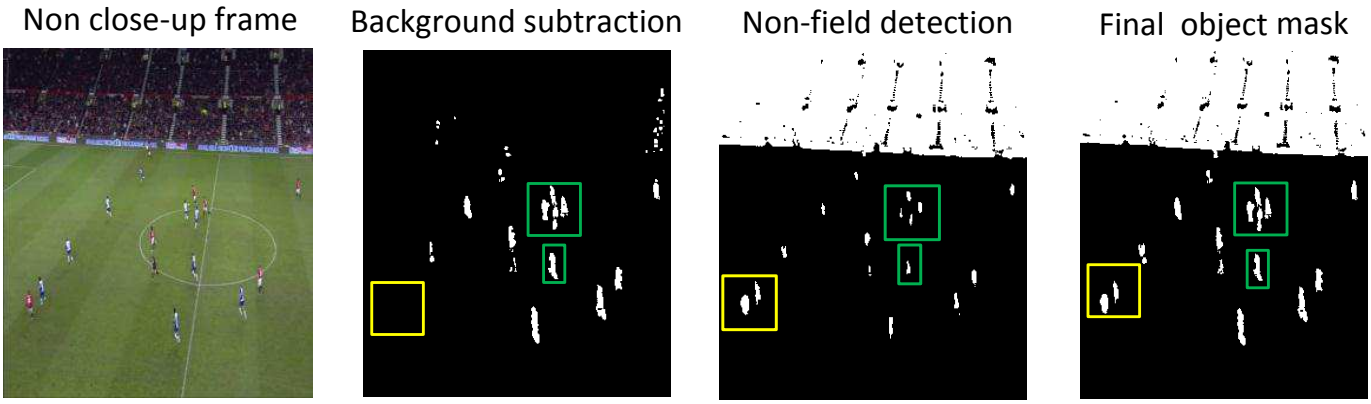



Fig. 7: An example of object mask creation for non close-ups. For a more complete detection of players, the final object mask is generated as a fusion of the non-field detection and the background subtraction approach. For instance, players in the yellow box are missing from the background subtraction approach, while the players in the green boxes are small when using the field-only method. However the final mask recovers the missing players and generates a more-complete object mask.

### C. Object Detection for Close-up Shots

In order to detect players in close-up shots we use a combination of frame-to-frame motion and feature point trajectories to obtain foreground and background matting strokes. Matting is then performed using these strokes and the generated mattes can be used as object masks after thresholding. However, in order to achieve cleaner results, we use field detection to remove possible mis-classified field areas.

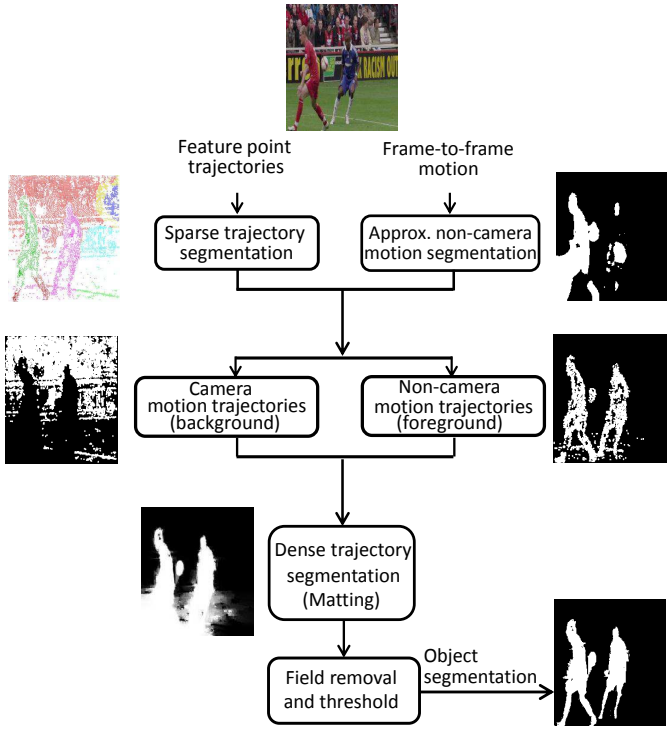



Fig. 8: For a close-up shot, a combination of feature point trajectories [29] and frame-to-frame motion [27] is used to generate background and foreground matting strokes. The method in [25] is then used to extract a dense players matte. Finally, field segmentation removes matting inaccuracies.

Frame-to-frame motion is estimated through the optical flow method in [27], which provides us with a color coded flow field. We fit a GMM [9] to the color coded flow field, and take the cluster with the most dominating Gaussian distribution as the camera motion segment. All other clusters are considered as the non-camera motion segment. This segmentation often has poor object boundaries and is not temporally coherent (see Fig. 8, approx. non-camera motion segmentation). Hence it can not be used directly as object masks. Instead we combine it with sparse trajectories segmentation to obtain foreground and background matting strokes.

Sparse trajectories segmentation is obtained through extracting feature point trajectories and segmenting them into different groups [29]. This generates a sparse labelling for different objects (see Fig. 8, sparse trajectory segmentation).

In order to combine sparse trajectories segmentation with non-camera motion segmentation we estimate the overlap of each trajectory segment with the non-camera motion segment. If there is at least a $30\%$ overlap, we label the trajectory segment as foreground (see Fig. 8, foreground), else background (see Fig. 8, background).

The feature point trajectories become the matting strokes and the method by Levin *et al.* [25] is used to extract a soft-mask of the players (see Fig. 8, matting). We then correct possible field mis-classifications by using the field detection of Sec. V-A. This generates cleaner player boundaries. Finally, we threshold the generated mattes by $0.3$ and get the final object masks (see Fig. 8, object segmentation).

## VI. Evaluation

All components of our proposed method have been implemented and compared against the closest system in the literature [20], and the ground-truth where available. For our experiments, both real and synthetic sequences have been considered.

Note that the few parameters in our method are experimentally tuned once for all sequences. Specifically, we set the number of candidate images $K$ to 10, the block size $n$ to 9, and the gradient refinement parameter $\alpha$ to 60.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2017.2748458, IEEE Transactions on Multimedia



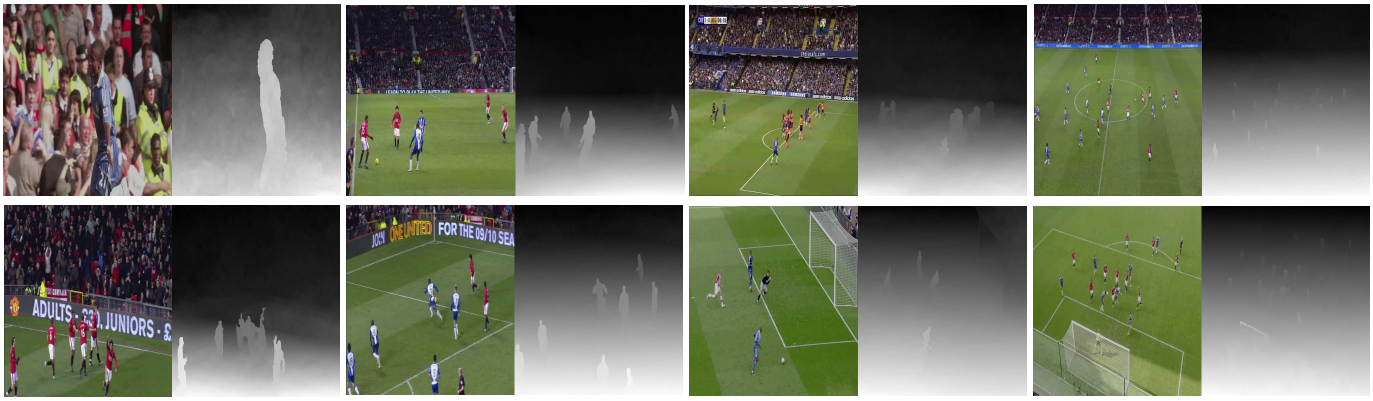


Fig. 9: Depth estimation for different types of shots using our method. Our method handles a wide variety of shots including Close-ups (e.g., top, left-most), Medium Shots (e.g., bottom, left-most), Bird's Eye View (e.g., bottom, right-most) and Long Shots (e.g., top, right-most).

### A. Examined Methods

Our 2D-to-3D conversion technique, which we refer to as DGC (short for Depth Gradient-based Conversion), is compared against several techniques as described below.

**Original 3D:** The original 3D-shot video that has been captured by stereo cameras. Results are compared subjectively.

**Ground-truth Depth:** Ground-truth depth maps are only available for synthetic sequences. As described in Sec. IV-A, they can be extracted from FIFA13 using PIX [4].

**DT:** The state-of-the-art method for data-driven 2D-to-3D conversion, Depth Transfer [20], trained on its own MSR-V3D database. MSR-V3D is available online and contains videos that have been captured using Microsoft Kinect.

**DT+:** Depth Transfer trained on our synthetic S-RGBD database. As stated in [20], capturing depth using Kinect is limited to indoor environments. This in addition to its erroneous measurements and poor resolution, limits Kinect's ability in generating a large soccer database. In order to have a rigorous comparison, we trained Depth Transfer on our soccer database and compared it against our technique.

**Depth from Stereo:** For an objective comparison of our method against the original side-by-side 3D, we need to approximate the ground-truth depth. We do so using the stereo correspondence technique in [10]. While stereo correspondence techniques do not always produce accurate results, they can sometimes capture the overall depth structure and thus be used for objective analysis.

### B. Subjective Experiments

To assess the visual 3D perception we perform several subjective experiments, and compare our method against the original 3D and DT+. We then demonstrate the benefits of our spatio-temporal Poisson reconstruction, especially for more temporally challenging scenes.

*1) Setup:* Our subjective experiments are conducted according to the ITU BT.2021 recommendation [6]. This recommendation suggests three primary perceptual dimensions for 3D video assessment: 1) Picture quality, in terms of pixel resolution and the impact of compression. This dimension is mainly affected by transmission and/or encoding. 2) Depth quality, which measures the amount of perceived depth. 3) Visual (dis)comfort, which measures any form of physiological unpleasantness due to 3D perception, e.g., headache, eye strain, and fatigue. In our experiments, we measure depth quality and visual comfort. We do not examine picture quality as we do not degrade it using compression or transmission. Note that we realize that artifacts may appear during stereo synthesis due to depth imperfections, but such artifacts will be captured by the depth quality and visual comfort dimensions.

The test sequences were displayed on a 55" Philips TV-set with passive polarized glasses. The lighting conditions were low. According to the ITU recommendations, we set the duration of each sequence to be between $10-15$ seconds, and the viewing distance to be around $3\,m$ for videos with a resolution of $1280\times720$ and around $2m$ for $1920\times1080$. We used static and dynamic random dot stereograms to test subjects' stereoscopic vision prior to the experiment. A stabilization phase was also performed before the actual experiments, where the subjects were asked to rate $4$ representative sequences with 3D qualities ranging from best to worst. While these representative sequences were not part of the actual test, this phase was useful in stabilizing the subjects' expectations and making them familiar with the rating protocol. The subjects were asked to ensure their full understanding of the experimental procedure prior to the actual test.

*2) Evaluation of our Technique:* For evaluating our 2D-to-3D conversion method, we show the subjects our converted sequences, and measure their average satisfaction. We assess depth quality and visual comfort for four real soccer sequences using the single-stimulus (SS) method of the ITU recommendations. We carefully created the four soccer sequences using clips from original 3D videos such that each includes a different category of shots: long shots, medium shots, close-ups, and bird's eye view. A long shot shows almost the entire field from a high camera position (Fig. 9, top right-most). In medium shots the camera is placed at a lower height and a smaller part of the field is visible (Fig. 9, bottom left-most). Close-ups have the camera zoomed on one or few players (Fig. 9, top left-most). In a bird's eye view the camera is placed above the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2017.2748458, IEEE Transactions on Multimedia



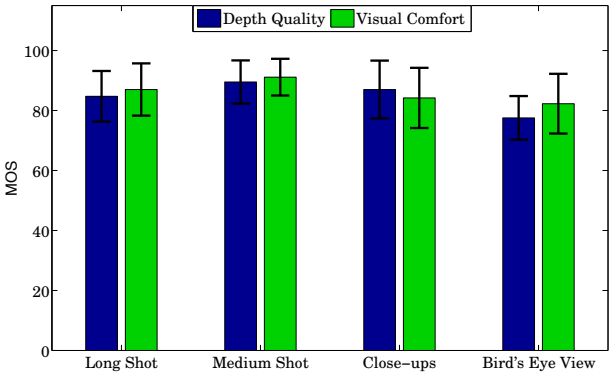



Fig. 10: Mean opinion scores of depth perception and visual comfort for different types of soccer scenes.

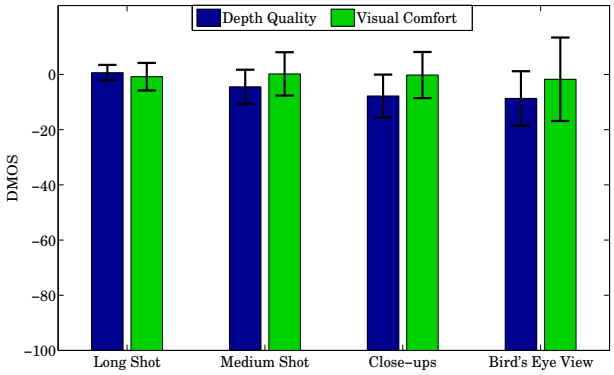



Fig. 11: Difference mean opinion score (DMOS) between our converted sequences and the original 3D. Zero implies that our converted sequence is the same as the original 3D.

field (Fig. 9, bottom right-most). Fifteen subjects participated in this study. The sequences are shown to subjects in random order. Before displaying each sequence, a $5\ sec$ mid-grey field is displayed which indicates the coded name of the sequence. The $10-15\ sec$ sequence is then displayed, followed by a $10\ sec$ mid-grey field which asks the subjects to vote. The standard ITU continuous scale is used for rating. For depth quality, the labels marked on the continuous scale are Excellent, Good, Fair, Poor, and Bad, while for visual comfort the labels are Very Comfortable, Comfortable, Mildly Uncomfortable, Uncomfortable, and Extremely Uncomfortable. We asked the subjects to mark their scores on these continuous scales. Their marks were then mapped to integer values between 0-100 and the mean opinion score (MOS) was calculated.

The MOS for all four sequences is shown in Fig. 10. For all sequences, DGC was rated in the range of Excellent by most subjects. Examples of estimated depth maps are shown in Fig. 9. Note how DGC can handle a wide spectrum of video shots, including different camera views and clutter.

In addition, in order to show the potential of our method on field sports other than soccer, we examined four real non-soccer sequences containing clips from Baseball, Tennis, Field Hockey and American Football. However, it is important to note that these sequences are only meant to show the *potential* of our method, as the soccer database was actually used for converting them. For a high quality conversion of such sequences a proper database should be designed. The results show that Field Hockey achieved the highest score (Excellent) as it resembles soccer the most, while the lowest score was for American Football (Good).

*3) Comparison against Original 3D:* We compare our converted videos against videos that are originally shot using stereo cameras. For this experiment, the Double Stimulus Continuous Quality Scale (DSCQS) method of the ITU recommendations is used. According to DSCQS, in order to assess the differences between each pair of sequences (original 3D and our converted 3D) properly, each pair should be observed by subjects at least twice prior to voting. Fifteen subjects participated in this study as well. The sequences were shown to them in random order without them knowing which is the original one. We then asked the subjects to rate depth quality and visual comfort for both sequences using the standard ITU continuous scale. Their marks are then mapped to integer values between 0-100 and used for calculating the Difference Opinion Score (= score for DGC - score for original 3D). Finally we calculate the mean of the difference opinion scores (DMOS).

A DMOS of zero implies that our converted 3D is judged the same as the original 3D, while a negative DMOS implies our 3D has a lower depth perception/visual comfort than the original 3D. The DMOS of the soccer sequences for both depth quality and comfort is shown in Fig. 11. It can be seen that our conversion achieves comparable quality to the original 3D. This is especially true for long shots which account for around $70\%$ of a full soccer game [12]. It is interesting to note that for some subjects our conversion was more comfortable than the original 3D. They reported that the popping out effect in original 3D was sometimes causing discomfort.

*4) Comparison against State-of-the-Art:* We compare our conversion technique against Depth Transfer (DT+) [20]. Similar to the previous experiments, the study is done with fifteen subjects, the DSCQS method is used, and the DMOS is calculated for both depth quality and comfort. For this experiment, we examined the close-up and medium shot sequences since they are the most challenging sequences for 2D-to-3D conversion due to their wide spectrum of camera angles, occlusion, clutter, and complex motion. Fig. 12 shows

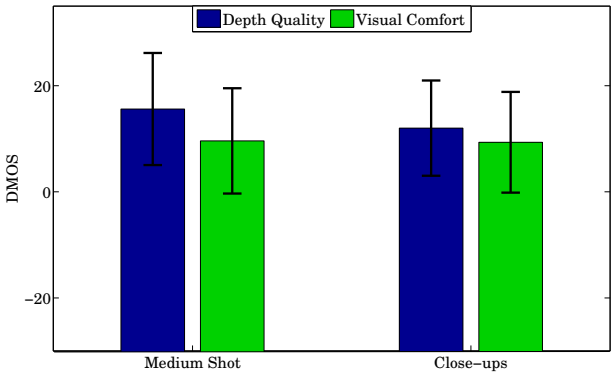



Fig. 12: Difference mean opinion score (DMOS) between our converted sequences and Depth Transfer DT+. Positive DMOS means that our technique is preferred over DT+.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2017.2748458, IEEE Transactions on Multimedia



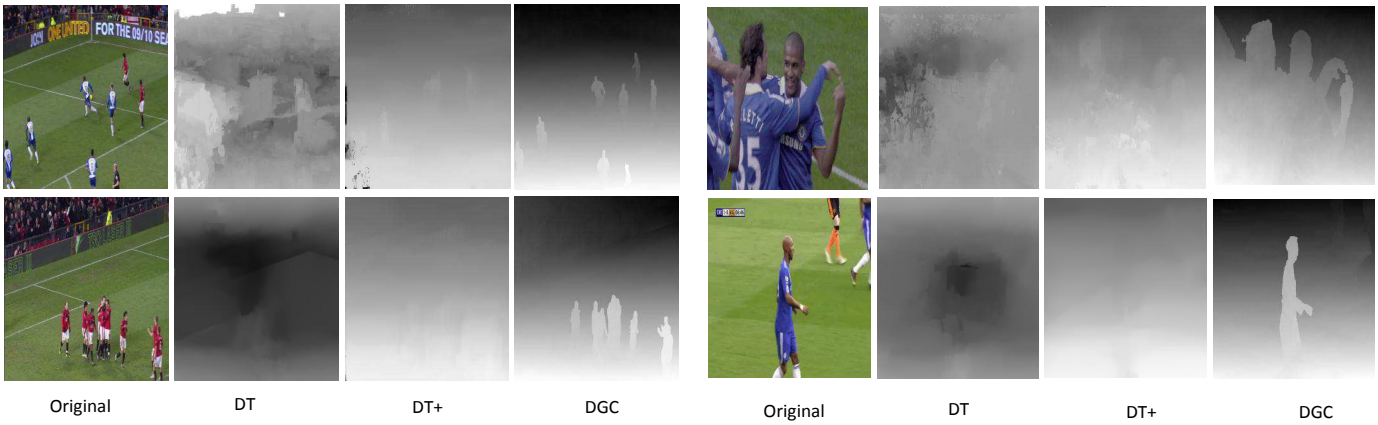


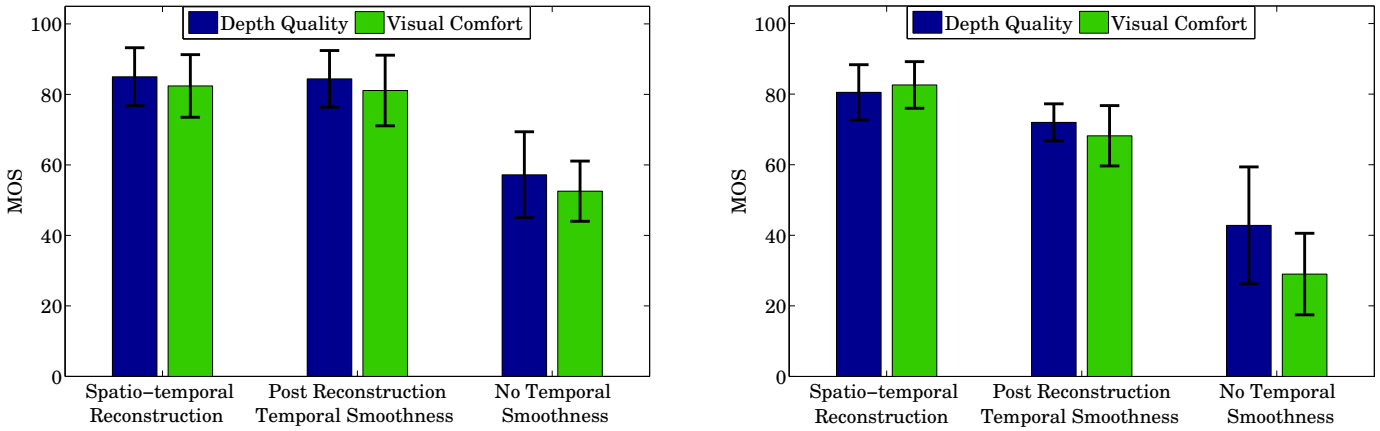

Fig. 13: Depth estimation for different sequences using: DT, DT+ and our method DGC. DT generates erroneous estimates, DT+ generates noisy measurements and does not detect players. Our technique outperforms both approaches.

DMOS for the medium shot and close-up against DT+. DT+ is outperformed by our method with an average of 12 points in close-ups and 15 points in medium shots. In addition, our technique was rated higher or equal to DT+ by all 15 subjects and the differences reported are statistically significant (p-value $< 0.05$). Fig. 13 shows some examples of extracted depth maps for DT, DT+ and our DGC. Note that as it can be seen in Fig. 13, the original implementation of Depth Transfer (DT) is much worse than DT+. Furthermore, it can be seen from Fig. 13 and Fig. 15 that the depth from DT+ can be very noisy sometimes, which in long term can cause eye strain.

*5) Effect of Spatio-temporal Poisson Reconstruction:* As discussed in Section IV-C, estimating depth independently for each frame may result in significant difference between the depth of consecutive frames. While simple shots may not suffer much from this problem and have a temporally smooth depth without the need of any further temporal enhancements, shots with complex and detailed texture, such as close-ups, may suffer from significant variations in the depth maps of successive frames. This may degrade the quality of depth perception and cause visual discomfort.

While temporally smoothing the depth maps during a post-processing phase works well for simple shots, it cannot completely overcome the problem for temporally complex shots. Our spatio-temporal Poisson reconstruction method, however, generates temporally and spatially smooth depth maps by utilizing temporal gradients in addition to spatial gradients during the depth calculation process. Thus, it can handle all types of shots and generate a comfortable and temporally smooth depth for all cases.

To assess the performance of our spatio-temporal method, we created two 10 $sec$ sequences. The first one is composed of various shots from the four soccer sequences used in the previous experiments, which are all rather simple to handle. We refer to this sequence as *Temporally Simple*. The second sequence is composed of various temporally complex soccer shots that are difficult to handle. In the figures, we refer to this sequence as *Temporally Complex*. The shots included in this




(a) *Temporally Simple* sequence

(b) *Temporally Complex* sequence

Fig. 14: Mean opinion scores of depth perception and visual comfort for two sequences of different temporal complexity, where three methods are compared: our spatio-temporal Poisson reconstruction, temporal smoothness as a post-process, and without temporal smoothness.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2017.2748458, IEEE Transactions on Multimedia



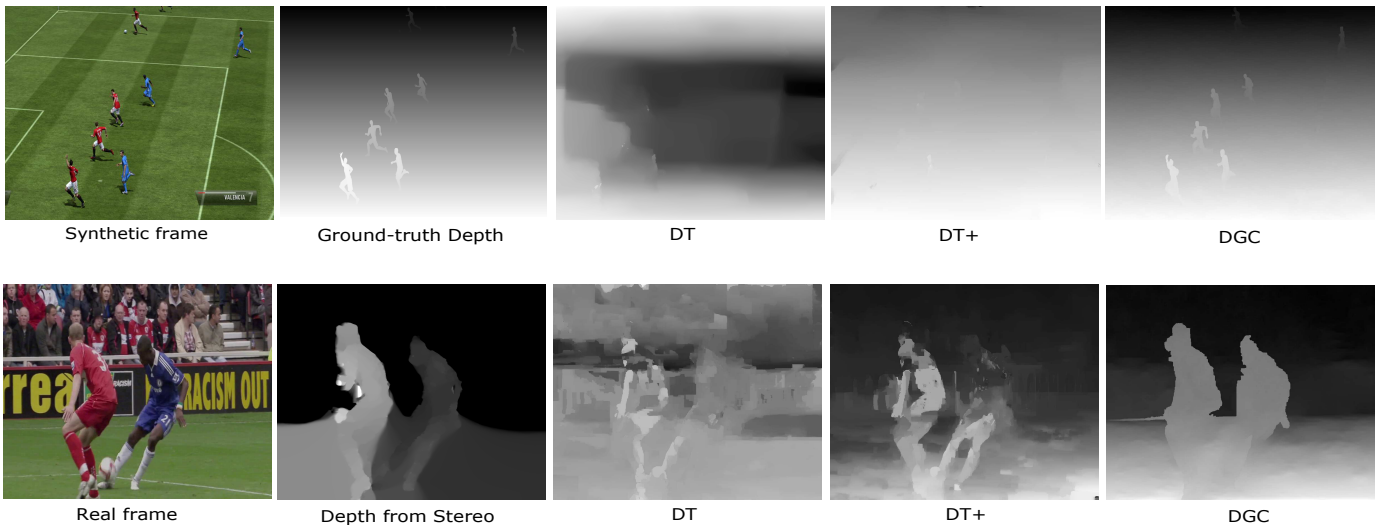



Fig. 15: Top row: Frame 3 of a synthetic sequence. Bottom row: Frame 24 of a real sequence. We show the depth extracted using: Ground-truth/Stereo Correspondence [10], DT, DT+ and DGC. Our technique DGC best resembles the Ground-truth/Stereo Correspondence in both sequences.

sequence were not included in the four sequences previously used.

We showed the subjects three versions of each sequence: 1) Without any temporal smoothness. 2) Temporal smoothness applied as a post-process on the depth maps generated by a regular (spatio) Poisson reconstruction. For this we use the temporal smoothness provided by Karsch *et al.* [20] as part of their stereo-warping technique. 3) Temporal smoothness integrated in the depth generation process using our proposed spatio-temporal Poisson reconstruction, without any further post-processing refinements. We then assess depth quality and visual comfort for all sequences using the single-stimulus (SS) method of the ITU recommendations.

Ten subjects participated in this study. We showed them the sequences in random order and they were asked to rate depth quality and visual comfort using the standard ITU continuous scale. Fig. 14(a) shows MOS for the three versions of the *Temporally Simple* sequence. It can be seen that while no temporal smoothness causes degradation in the comfort and thus the depth quality, it can be fully resolved by post-processing. As a result, there is very little difference between the results of our spatio-temporal reconstruction and that of post reconstruction smoothing. However, the benefits of our spatio-temporal reconstruction become more clear in the *Temporally Complex* sequence, where post-processing is unable to fully overcome the problem. Fig. 14(b) shows MOS for this sequence. It can be seen that our spatio-temporal reconstruction improves the comfort by an average of 14 points compared to post reconstruction smoothing, and enhances the quality from Good to Excellent. The differences reported in this figure are statistically significant (p-value $< 0.05$).

### C. Objective Experiments

We perform objective experiments on both real and synthetic sequences to measure the quality of our depth maps and compare it against the state-of-the-art. We then analyse

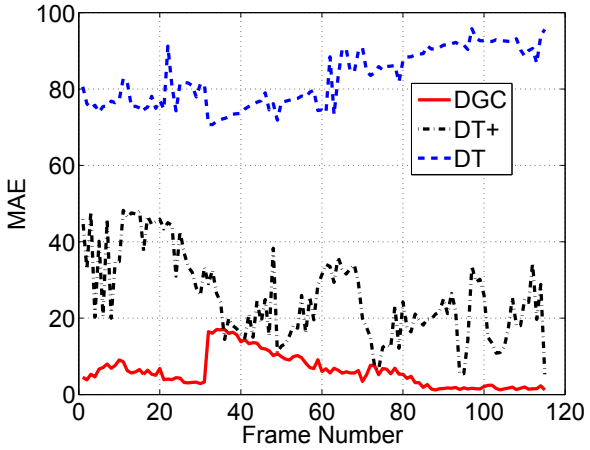



Fig. 16: An objective comparison between our DGC method and the closest method in the literature DT, and its extension DT+ on a synthetic soccer sequence.

the effect of our spatio-temporal Poisson reconstruction on temporal smoothness, and the effect of database size on depth quality.

*1) Comparison against State-of-the-Art:* For an objective comparison against state-of-the-art, we choose two sequences: a synthetic sequence and a real sequence. For the synthetic sequence we extract 2D+Depth for around 120 frames in the same way that the database was created (Sec. IV-A). In Fig. 15 (top) a frame of the synthetic sequence is shown followed by its ground-truth depth and estimated depth when using different methods (DT, DT+ and our DGC). All demonstrated depth maps are normalized and in the range of $(0-255)$. Results from DT are largely erroneous since the data in MSR-V3D hardly resembles soccer. While being trained on our database makes the results from DT+ significantly better, most players are yet not detected. Our technique DGC, however, manages to detect the players and generate a smooth depth that best resembles

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2017.2748458, IEEE Transactions on Multimedia



ground-truth. The Mean Absolute Error (MAE) against ground-truth for the whole synthetic sequence is shown in Fig. 16. As shown it the figure, the MAE of our method is much less than both DT and DT+.

Due to the absence of ground-truth depth for real sequences, performing objective analysis on them is challenging. In [20], Kinect depth was used as ground-truth. However, Kinect is incapable of capturing depth information in outdoor environments. As a result, Kinect cannot be used for generating ground-truth estimates for soccer matches. Instead, given a 3D-shot soccer sequence, we use stereo correspondence [10] to approximate the ground-truth depth map. Fig. 15 (bottom) shows a frame from one of the most challenging test sequences. Its extracted depth, though not perfect, captures the overall depth structure and can be used for inferring the quality of the converted depth maps. The estimated depth maps using DT, DT+ and our DGC are also shown in Fig. 15 (bottom). It can be seen that our technique (DGC) best resembles the ground-truth. In addition, our objective experiments over a range of 100 frames show that DGC reduces MAE 17% and 48% on average compared to DT+ and DT respectively. Figure is omitted due to space limitations.

*2) Effect of Spatio-temporal Poisson Reconstruction:* In order to demonstrate the advantage of our spatio-temporal Poisson reconstruction, we use the same two real and synthetic sequences (shown in Fig. 15). For each sequence, we generate the depth maps using a temporal window size of: *one* (without temporal smoothness), *three* (one frame before and one after), and *five* (two frames before and two after). Fig. 17 shows the average depth values for each frame of the synthetic sequence. It can be seen that without temporal smoothness the scene experiences sudden changes from frame to frame, but the changes become smoother as the window size increases. Also, without temporal smoothness the difference between the maximum and minimum average depth value is around 70, while with temporal smoothness it is reduced to around 15. Results for the real sequence (figure is omitted due to space limitations) also show that while without temporal smoothness the difference between the maximum and minimum average depth value is around 110, this value is reduced to around 50 when temporal smoothness is applied.

Fig. 18 shows the MAE between the depth of each frame in the real sequence and its previous frame. Each pixel is compared to its corresponding pixel in the previous frame, where the corresponding pixels are identified using optical flow. It can be seen that applying temporal smoothness significantly reduces the MAE. However, there is not much gain in increasing the window size from 3 to 5. MAE results for the synthetic sequence (figure is omitted due to space limitations) also show that the MAE is reduced from a maximum of 57 (without smoothness) to a maximum of 3 (window size of 5).

*3) Effect of Database Size:* To investigate the importance of our S-RGBD database size we examined six different database sizes: 1000, 2000, 4000, 8000, 13000 and 16000 images. For this experiment, a synthetic sequence with 120 frames was generated. This sequence includes a wide variety of shots that can occur in a soccer match. Results show that up to a size of 8,000 images, due to the absence of big enough data the performance fluctuates around an MAE of 30. Starting from 13,000 images there is a boost in performance which reduces MAE to around 20. However, the performance stabilizes around 16,000 images (Figure is omitted due to space limitations). Thus, a database of 16,500 images was used in our evaluation.

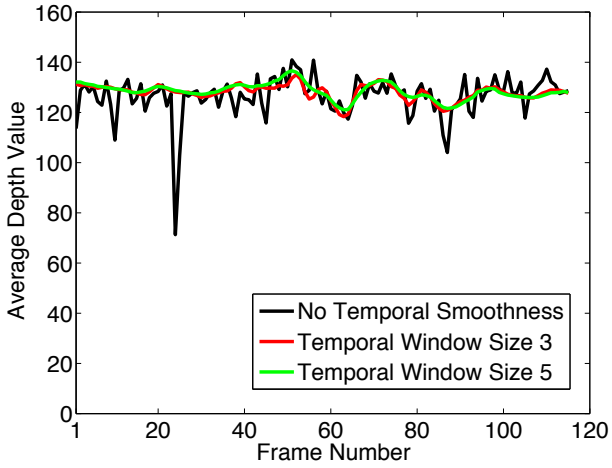



Fig. 17: The average depth values for each frame of a *synthetic* sequence, when using different temporal window sizes. While without temporal smoothness the scene experiences sudden changes of depth, the depth changes are much smoother when temporal smoothness is applied.

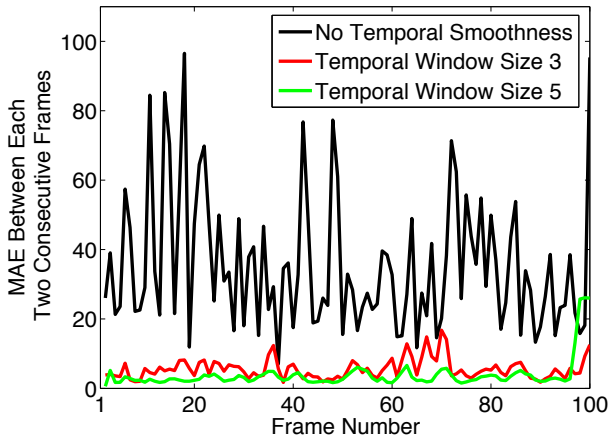



Fig. 18: Mean Absolute Error (MAE) between each two consecutive frames of a *real* sequence, when using different temporal window sizes.

## VII. Conclusions and Future Work

We presented a 2D-to-3D video conversion method for soccer videos that, unlike previous methods, can handle the motion complexities and the wide variety of scenes present in soccer matches. Our method transfers depth gradients from a synthetic database of soccer videos and estimates depth through a spatio-temporal Poisson reconstruction. We implemented our method and used both real and synthetic sequences for evaluating it. Our subjective and objective results show the capablity of our method in handling a wide spectrum of shots with different camera views, colors, motion complexities, occlusion, and

clutter. Our created 3D videos were rated Excellent by most subjects. In addition, our method outperforms the state-of-the-art both subjectively and objectively, in all real and synthetic sequences.

This paper contributes three *key findings* that can impact the area of 2D-to-3D video conversion, and potentially 3D video processing in general. First, domain-specific conversion can provide much better results than general methods. Second, transferring depth gradient on a block basis not only produces smooth natural depth when reconstructed using Poisson, but it also reduces the size of the required reference database. Third, synthetic databases created from computer-generated content can easily provide large, diverse, and accurate texture and depth references for various 3D video processing applications.

## VIII. Acknowledgments

This research was supported in part by NSERC (Natural Sciences and Engineering Research Council of Canada), the QCRI-CSAIL partnership, NSF grant IIS-1111415.

## References

[1] Berkeley 3-D object dataset. http://kinectdata.com/.

[2] Make3D. http://make3d.cs.cornell.edu/data.html.

[3] NYU depth dataset v2. http://cs.nyu.edu/~silberman/datasets/nyu_depth_v2.html.

[4] Performance Invistigator for Xbox (PIX). https://msdn.microsoft.com/en-us/library/windows/desktop/ee663275%28v=vs.85%29.aspx.

[5] RGB-D object dataset. http://rgbd-dataset.cs.washington.edu/.

[6] ITU-R BT.2021-1, Subjective methods for the assessment of stereoscopic 3DTV systems. Geneva, Switzerland, February 2015. International Telecommunication Union.

[7] L. J. Angot, W.-J. Huang, and K.-C. Liu. A 2d to 3d video and image conversion technique based on a bilateral filter. In *Proc. of SPIE 7526, Three-Dimensional Image Processing (3DIP) and Applications*, pages 75260D:1–10, San Jose, California, January 2010.

[8] P. Bhat, B. Curless, M. Cohen, and C. Zitnick. Fourier analysis of the 2D screened poisson equation for gradient domain problems. In *Proc. of European Conference on Computer Vision (ECCV'08)*, pages 114–128. Marseille, France, October 2008.

[9] C. Bouman. Cluster: An unsupervised algorithm for modeling gaussian mixtures. Technical report, Purdue University, 1998.

[10] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert. High accuracy optical flow estimation based on a theory for warping. In *Proc. of European Conference on Computer Vision (ECCV'04)*, pages 25–36, Prague, Czech Republic, May 2004.

[11] K. Calagari, M. Elgharib, P. Didyk, A. Kaspar, W. Matusik, and M. Hefeeda. Gradient-based 2D-to-3D conversion for soccer videos. In *Proc. of ACM Multimedia Conference (MM'15)*, pages 331–340, Brisbane, Australia, October 2015.

[12] K. Calagari, K. Templin, T. Elgamal, K. Diab, P. Didyk, W. Matusik, and M. Hefeeda. Anahita: A System for 3D Video Streaming with Depth Customization. In *Proc. of ACM Multimedia Conference (MM'14)*, pages 337–346, Orlando, Fl, November 2014.

[13] D. Corrigan, N. Harte, and A. Kokaram. Pathological motion detection for robust missing data treatment. *EURASIP Journal on Advances in Signal Processing*, 2008(153):1–16, 2008.

[14] M. Guttmann, L. Wolf, and D. Cohen-Or. Semi-automatic stereo extraction from video footage. In *Proc. of International Conference on Computer Vision (ICCV'09)*, pages 136–142, Kyoto, Japan, September 2009.

[15] A. Hassanien, M. A. Elgharib, A. Selim, S.-H. Bae, M. Hefeeda, and W. Matusik. Large-scale, fast and accurate shot boundary detection through spatio-temporal convolutional neural networks. *CoRR*, abs/1705.03281, 2017.

[16] M. Hefeeda, T. ElGamal, K. Calagari, and A. Abdelsadek. Cloud-based multimedia content protection system. *IEEE Transactions on Multimedia*, 17(3):420–433, 2015.

[17] D. Hoiem, A. A. Efros, and M. Hebert. Automatic photo pop-up. *ACM Transactions on Graphics*, 24(3):577–584, 2005.

[18] S. Hong, H. Noh, and B. Han. Decoupled deep neural network for semi-supervised semantic segmentation. In *Proc. of Advances in Neural Information Processing Systems (NIPS'15)*, pages 1495–1503, Montreal, Canada, December 2015.

[19] H. Jiang, G. Zhang, H. Wang, and H. Bao. Spatio-temporal video segmentation of static scenes and its applications. *IEEE Transactions on Multimedia*, 17(1):3–15, 2015.

[20] K. Karsch, C. Liu, and S. B. Kang. Depth transfer: Depth extraction from video using non-parametric sampling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(11):2144–2158, 2014.

[21] J. Ko. 2D-to-3D Stereoscopic Conversion: Depth Estimation in 2D Images and Soccer Videos. Master's thesis, Korea Advanced Institution of Science and Technology (KAIST), 2008.

[22] 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. In *Proc. of SPIE 8288, Stereoscopic Displays and Applications XXIII*, pages 82880F:1–12, Burlingame, California, January 2012.

[23] J. Konrad, M. Wang, P. Ishwar, C. Wu, and D. Mukherjee. Learning-based, automatic 2D-to-3D image and video conversion. *IEEE Transactions on Image Processing*, 22(9):3485–3496, 2013.

[24] G. G. Lakshmi Priya and S. Domnic. Walsh-hadamard transform kernel-based feature vector for shot boundary detection. *IEEE Transactions on Image Processing*, 23(12):5187–5197, 2014.

[25] A. Levin, D. Lischinski, and Y. Weiss. A closed-form solution to natural image matting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(2):228–242, 2008.

[26] M. Liao, J. Gao, R. Yang, and M. Gong. Video stereolization: Combining motion analysis with user interaction. *IEEE Transactions on Visualization and Computer Graphics*, 18(7):1079–1088, 2012.

[27] C. Liu. *Beyond Pixels: Exploring New Representations and Applications for Motion Analysis*. PhD thesis, Massachusetts Institute of Technology, 2009.

[28] 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, 2011.

[29] P. Ochs, J. Malik, and T. Brox. Segmentation of moving objects by long term video analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(6):1187–1200, 2014.

[30] J.-M. Odobez and P. Bouthemy. Robust multiresolution estimation of parametric motion models. *Journal of Visual Communication and Image Representation*, 6(4):348–365, 1995.

[31] 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.

[32] M. Park, J. Luo, A. C. Gallagher, and M. Rabbani. Learning to produce 3D media from a captured 2D video. *IEEE Transactions on Multimedia*, 15(7):1569–1578, 2013.

[33] P. Pérez, M. Gangnet, and A. Blake. Poisson image editing. *ACM Transactions on Graphics*, 22:313–318, 2003.

[34] R. Phan and D. Androutsos. Robust semi-automatic depth map generation in unconstrained images and video sequences for 2d to stereoscopic 3d conversion. *IEEE Transactions on Multimedia*, 16(1):122–136, 2014.

[35] A. Saxena, S. H. Chung, and A. Y. Ng. Learning depth from single monocular images. In *Proc. of Advances in Neural Information Processing Systems (NIPS'05)*, pages 1161–1168, Vancouver, Canada, December 2005.

[36] L. Schnyder, O. Wang, and A. Smolic. 2D to 3D conversion of sports content using panoramas. In *Proc. of IEEE Conference on Image Processing (ICIP'11)*, pages 1961–1964, Brussels, Belgium, September 2011.

[37] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004.

[38] Z. Yang, W. Wu, K. Nahrstedt, G. Kurillo, and R. Bajcsy. Viewcast: View dissemination and management for multi-party 3D tele-immersive environments. In *Proc. of ACM Multimedia Conference (MM'07)*, pages 882–891, Augsburg, Bavaria, Germany, September 2007.

[39] L. Zhang, C. Vázquez, and S. Knorr. 3D-TV content creation: automatic 2D-to-3D video conversion. *IEEE Transactions on Broadcasting*, 57(2):372–383, 2011.

[40] Z. Zhang, Y. Wang, T. Jiang, and W. Gao. Visual pertinent 2d-to-3d video conversion by multi-cue fusion. In *Proc. of IEEE International Conference on Image Processing (ICIP'11)*, pages 909–912, Brussels, Belgium, September 2011.

[41] Z. Zhang, C. Zhou, Y. Wang, and W. Gao. Interactive stereoscopic video conversion. *IEEE Transactions on Circuits and Systems for Video Technology*, 23(10):1795–1808, 2013.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2017.2748458, IEEE Transactions on Multimedia



**Kiana Calagari** received her BSc and MSc degrees in Electrical Engineering from Sharif University of Technology, Iran, in 2010 and 2012, respectively. She is currently finalizing her PhD studies in Computing Science at Simon Fraser University, Canada. Her research interests cover various aspects of multimedia systems and applications including interactive video streaming, customizing 3D videos, and generating immersive content such as 3D videos and VR content.

**Mohamed Elgharib** received the BAI and PhD from Trinity College Dublin in 2008 and 2011. He then moved to Boston University where he worked in video surveillance. He is now with Qatar Computing Research Institute working on visual data utilization for image manipulation. Through out his career he worked on several problems covering a wide spectrum of topics in computer vision and computer graphics. Some of his notable works include motion magnification, sports video processing, painting style transfer and video reflection removal. They have been featured on several news outlets including MIT main page, MIT news, and BBC News. His paper in SIGGRAPH'16 was selected for the back cover of the proceedings.

**Piotr Didyk** is an independent research group leader at the Excellence Cluster for Multimodal Computing and Interaction at Saarland University in Germany where he is a head of Perception, Display, and Fabrication Group. Prior to this, he spent two years as a postdoctoral associate at Massachusetts Institute of Technology. In 2012, he obtained his Ph.D. from the Max Planck Institute for Informatics and the Saarland University for his work on perceptual displays. His research interests include novel display technologies and computational fabrication.

**Alexandre Kaspar** obtained a B.Sc. and a M.Sc. from the Ecole Polytechnique Fdrale de Lausanne (EPFL) in Switzerland respectively in 2011 and 2014. He is currently a Ph.D. student at the Massachusetts Institute of Technology, USA. His research interests include computer graphics, vision and fabrication.

**Wojciech Matusik** is an Associate Professor of Electrical Engineering and Computer Science at the Computer Science and Artificial Intelligence Laboratory at MIT, where he leads the Computational Fabrication Group. Before coming to MIT, he worked at Mitsubishi Electric Research Laboratories, Adobe Systems, and Disney Research Zurich. He studied computer graphics at MIT and received his PhD in 2003. He also received a BS in EECS from the University of California at Berkeley in 1997 and MS in EECS from MIT in 2001. His research interests are in direct digital manufacturing and computer graphics. In 2004, he was named one of the worlds top 100 young innovators by MITs Technology Review Magazine. In 2009, he received the Significant New Researcher Award from ACM Siggraph. In 2012, Matusik received the DARPA Young Faculty Award and he was named a Sloan Research Fellow.

**Mohamed Hefeeda** received his Ph.D. from Purdue University, USA in 2004, and M.Sc. and B.Sc. from Mansoura University, Egypt in 1997 and 1994, respectively. He is a Professor in the School of Computing Science, Simon Fraser University, Canada, where he leads the Network Systems Lab. His research interests include multimedia networking over wired and wireless networks, mobile multimedia, and network protocols. In 2011, he was awarded one of the prestigious NSERC Discovery Accelerator Supplements (DAS), which are granted to a select group of distinguished researchers in all Science and Engineering disciplines in Canada. He has co-authored more than 100 refereed papers and multiple granted patents.