IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received May 6, 2021, accepted June 8, 2021, date of publication June 28, 2021, date of current version July 6, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3092777

Segmentation-Based Seam Cutting for High-Resolution 360-Degree Video Stitching

TAEHA KIM[1], SEONGYEOP YANG[1], BYEONGKEUN KANG[2], HEEKYUNG LEE[3], JEONGIL SEO[3], AND YEEJIN LEE[1]

[1]Department of Electrical and Information Engineering, Seoul National University of Science and Technology, Seoul 01811, South Korea
[2]Department of Electronic and IT Media Engineering, Seoul National University of Science and Technology, Seoul 01811, South Korea
[3]Immersive Media Research Section, Electronics and Telecommunications Research Institute, Daejeon 34129, South Korea

Corresponding author: Yeejin Lee (yeejinlee@seoultech.ac.kr)

This work was supported by the Electronics and Telecommunications Research Institute (ETRI) Grant through the ICT Research and Development Program of MSIT/IITP by the Immersive Media Research Laboratory under Grant 2018-0-00207.

ABSTRACT We present a novel segmentation-based seam cutting algorithm to generate visually plausible high-resolution 360-degree video efficiently. While the demand for an efficient video stitching algorithm for generating immersive videos has increased, it has received limited attention in the literature. Furthermore, stitched videos often suffer from distorted objects, temporal inconsistency and time constraints. Thus, in this paper, we propose an efficient seam finding algorithm that preserves objects from distortion, minimizes temporal inconsistency, and reduces processing time. One of the fundamental steps in image and video stitching is the estimation of seam boundary. To do this, the proposed algorithm leverages a convolutional neural networks-based instance segmentation algorithm that provides more accurate object regions. It computes energy surfaces considering the regions and then estimates seam boundary by discovering a minimal energy path with minimal computations. We validate the proposed algorithm using real-world high-resolution 360-degree sequences. The experimental results verify that the proposed algorithm can produce seam boundaries that avoid objects with better temporary consistency. The proposed algorithm reduces the number of pixels passed through objects by approximately 30% on average compared to the existing algorithms. The qualitative comparisons furthermore demonstrate that the proposed algorithm consistently produces more perceptually pleasing results.

INDEX TERMS Video stitching, image stitching, seam estimation, 360-degree video, instance segmentation, deep neural network.

I. INTRODUCTION

Video stitching is a crucial technique to generate immersive videos since omnidirectional scenes are usually captured by using multiple cameras [1], [2]. It creates a combined, usually wider field of view, video from a collection of videos with overlapping fields of view. A typical video stitching pipeline consists of feature extraction, correspondence matching, image registration, and image composition [3]–[5]. By feature extraction and correspondence matching, corresponding points are found between images from different views. Then, given the corresponding points, the image registration process aligns the images by transforming them into one coordinate system. Finally, as the aligned images have overlapping regions, the image composition step determines the color information in overlapping regions and their nearby locations.

While image registration has been extensively studied to enhance the quality of image alignments [6]–[10], it is still a challenging problem when input images are captured with large parallax, lens distortion, scene motion, and exposure difference [3]. Input images should be captured by cameras that are sufficiently far away from the scene to reduce parallax [3], [11], [12]. It, however, limits many potential applications and usages that apply video stitching only to images with small parallax, lens distortion, scene motion, and exposure difference. Thus, many image composition algorithms, such as seam cutting [13]–[17], and advanced image blending [4], [18] algorithms, have been proposed to

The associate editor coordinating the review of this manuscript and approving it for publication was Tomasz Trzcinski.

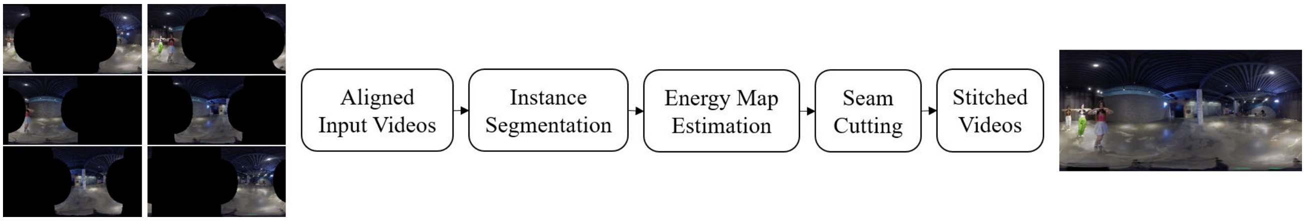


FIGURE 1. Overview of the proposed framework. The proposed framework focuses on the seam cutting algorithm that utilizes instance segmentation-based energy map estimation to minimize visual artifacts. The framework does not have any constraint on the number of input videos.

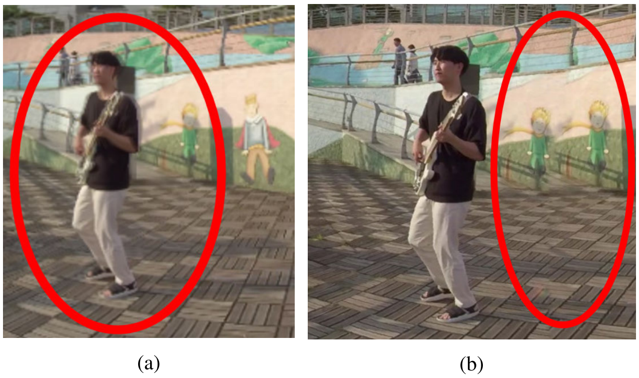


FIGURE 2. Examples of visual artifacts on the objects in the stitched images. The object boundary is blurred in (a), and the object on the wall is duplicated in (b).

relieve registration artifacts and produce visually plausible stitched images.

In addition to the challenges of image stitching, video stitching further suffers from visual artifacts caused by temporal inconsistency and time constraints. Additional visual artifacts are induced by transitions of seam location or light condition variations across successive frames. Considering processing time, as modern camera systems capture high-resolution videos at a high frame rate, the demand for efficient video stitching algorithms has been increasing. Nevertheless, efficient video stitching has received limited attention in the literature [19]–[22].

To this end, we proposed a new video stitching algorithm taking visual artifacts, spatiotemporal consistency, and time constraints into account. The proposed framework induces fewer visual artifacts by applying an efficient and effective seam finding algorithm. The overall framework of the proposed algorithm is explained in Figure 1. As one of the severe visual artifacts of video stitching is the distortion of objects (See Figure 2), we propose an energy map estimation method that leverages convolutional neural networks(CNN)-based instance segmentation to preserve the structures of objects. Moreover, to improve temporal consistency, we develop an efficient seam finding algorithm that includes a decision algorithm for seam updating and a spatiotemporal smoothing process, suppressing undesirable seam changes over frames. As a side note, image/video stitching is a typical ill-posed problem without ground-truth stitched images/videos. This makes it challenging to evaluate performance properly and to accomplish a quantitative performance analysis. Thus, we present a way to assess how well seam locations are determined and preserve contents in the stitched results. We then compare the performance of the proposed algorithm with the current state-of-art seam finding algorithms based on it. In summary, the contributions of this work are as follows:

- Propose an efficient seam finding algorithm for stitching 360-degree videos using instance segmentation and temporal smoothing process.
- Present a way to evaluate the performance of seam finding algorithms.
- Validate the performance of the proposed algorithm using both real-world videos and common benchmark videos.

The remainder of this paper is organized as follows. In Section II, we present related works regarding image/video stitching and instance segmentation. We then introduce the proposed algorithm including energy map estimation algorithm in Section III-A and seam finding algorithm in Section III-B. In Section IV, we present experimental results to demonstrate the effectiveness and efficiency of the proposed algorithm. Lastly, we conclude with a summary of key observations in Section V.

II. RELATED WORKS

A. *IMAGE AND VIDEO STITCHING*

1) IMAGE REGISTRATION

Many advanced methods have been studied for fixing the aforementioned artifacts by estimating more accurate homography and/or by applying spatially varying warping. Instead of using a single global homography, Gao *et al.* [11] proposed a dual-homography (DH) warping method by assuming that the scene contains a ground plane and a distant plane. The estimated two homographies were blended using the per-pixel weight that controls the contribution of each homography. Because a global affine transformation only well-fits to the regions of dense correspondences, Lin *et al.* [23] replaced the pre-computed global affine transformation with a smoothly varying affine (SVA) transform over the rest of the scene. The warp method, known as as-projective-as-possible (APAP) [24], also allows local deviation from the global transformation. An input image is partitioned into small cells, then the local homography

TABLE 1. The summary of the approaches and strengths/limitations of previous seam finding algorithms.

Method	Formulation	Strength	Limitation
Graphcut [14]	Min-cut Problem (Max-flow Problem)	Providing an approximately optimal solution to seam boundary	Computational expensive Memory intensive
Weaving [15]	Min-cut Problem (Max-flow Problem)	Faster than classic graph cuts Flexible to editing	Risk to produce improper seam by user interaction
Perception [16]	Min-cut Problem (Max-flow Problem)	Generating perceptually better stitching results	Risk to produce seams low-frequency region in objects
Object-centered [17]	MRF Inference Problem	Relieving object-related errors	Many hyperparameters to control energy function
Radiance [18]	Min-cut Problem (Max-flow Problem)	Invariant to large exposure difference	Need subsets of input images to create a panorama reference image

for each cell is refined by the weighted feature points. More recently, mesh-driven approaches were proposed aiming at improving local alignment quality. Zhang *et al.* [25] designed a mesh-based framework to optimize alignment by adding line preserving, orientation, and loop closure constraints. Another mesh-based approach proposed in [26] determines a total energy function by simultaneously optimizing the as-projective-as-possible warp and the quasi-homography warp [27] including alignment, distortion, and saliency. The seam-guided local alignment (SEAGULL) scheme iteratively optimizes local alignment by performing seam-guided feature reweighting [13]. The method in [28] used a global similarity prior (GSP) constraining edges away from the overlapping region to generate natural stitched images. To ensure more robust alignment, Li *et al.* [12] proposed a robust elastic warping method using TPS [29] model and a Bayesian feature refinement model based on robust elastic warping.

2) IMAGE COMPOSITION

Despite the studies to construct better alignment functions, the artifacts of non-ideal camera settings cannot be concealed only by better registration. Therefore, post-processing techniques have been extensively explored to mitigate registration artifacts between images, such as seam cutting and image blending. A general approach for finding optimal seam is discovering the path that minimizes certain important energy functions [14], [15]. Graphcut is a widely used seam selection algorithm that maximizes Markov random field (MRF) likelihood based on the similarity of pixels between the reference and warped images [14]. Based on the Graphcut algorithm, Eden *et al.* [18] defined the energy functions in radiance space. The functions prefer values with large signal-to-noise ratios and unsaturated/normally exposed in the scenes with large motions and exposure differences. Zhang *et al.* [25] defined the energy function as the combination of the alignment error and color difference and optimized it using the Graphcut algorithm. An energy map was modified based on the human eye's perception defined by saliency and color difference [16]. As advanced object detection techniques developed [30], [31], Herrmann *et al.* composited the energy using the detected object information [17]. Table 1 summarizes the approaches to find seams and pros/cons of previous seam finding algorithms.

3) VIDEO STITCHING

Compared to image stitching, video stitching has received limited attention in the literature. The straightforward approach is extending image stitching to video stitching. El-Saban *et al.* reduced the processing time by tracking interest points over multiple frames [19]. Jiang and Gu proposed the content-preserving warping algorithm that optimizes local alignment and image composition in the spatiotemporal domain [20]. To alleviate alignment artifacts in multi-view videos, Lee and Sim estimated the parameters of ground plane homography, fundamental matrix, and vertical vanishing points using the appearance and activity-based feature matches [21]. Recently, the work in [22] adapted a deep learning framework to stitch videos inspired by pushbroom cameras.

B. INSTANCE SEGMENTATION

Instance segmentation task aims to classify each pixel to the corresponding class and object instance [32]–[35]. It is closely related to object detection and semantic segmentation. Object detection task aims to detect all the objects in the given image by localizing them using bounding boxes and by classifying them to the corresponding classes [31], [36]–[39]. The goal of the semantic segmentation task is classifying each pixel in the given image to the corresponding class [40]–[43]. Object detection has a limitation on localizing objects at pixel-level while semantic segmentation cannot separate object instances. Instance segmentation overcomes the limitations by fusing the two tasks although it cannot segment stuff.

He *et al.* presented a framework for object instance segmentation by extending an object detection method [31] by adding a branch for predicting an object mask [32]. For real-time instance segmentation, Bolya *et al.* proposed a fully convolutional model [34]. The framework consists

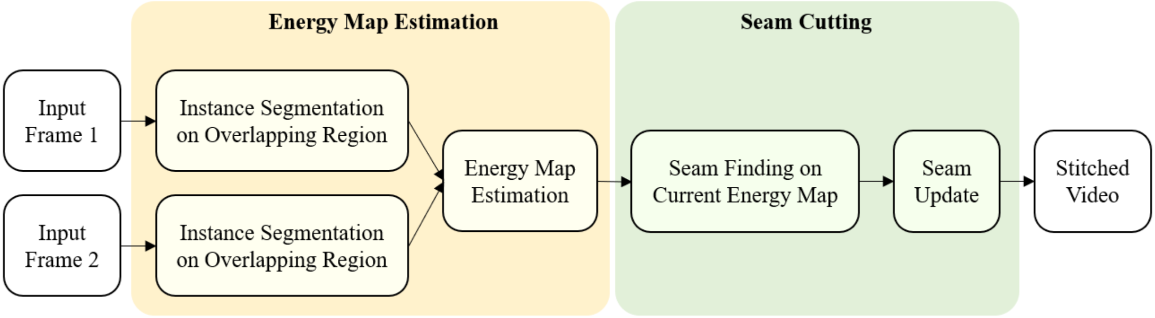

Energy Map Estimation
Seam Cutting
Input Frame 1
Input Frame 2
Instance Segmentation on Overlapping Region
Instance Segmentation on Overlapping Region
Energy Map Estimation
Seam Finding on Current Energy Map
Seam Update
Stitched Video

FIGURE 3. Description of the proposed algorithm.

of generating a set of prototype masks and predicting per-instance mask coefficients. Wang *et al.* presented a single-shot instance segmentation framework that utilizes uniform grids [35], [44] similar to YOLO [38]. For each grid cell, the method predicts a semantic class and the corresponding instance's mask.

III. PROPOSED METHOD

Existing seam finding algorithms do not necessarily produce visually pleasing results and suffer from visual artifacts. In particular, objects in overlapping regions are often distorted, cropped, duplicated, or occluded, as shown in the examples of Figure 2. In addition to these spatial-visual artifacts, video stitching inherently faces temporal coherence issues. Specifically, seam boundaries frequently change over frames, yielding blinking outputs in consecutive sequences. In order to overcome these limitations, we design a new approach that creates seam boundaries preserving contents in the stitched regions and maintains temporal consistency.

In this section, we provide the details of the proposed seam finding algorithm. We first discuss the energy map that enables detecting dominant objects in the overlapping regions. Besides, the function is designed for maintaining spatial-temporal consistency in videos. Then, we describe how to find seam boundaries that bypass dominant objects efficiently. The overall pipeline of the proposed algorithm is given in Figure 3.

A. ENERGY MAP ESTIMATION

After the initial registration, individual frames from different views are blended to produce a single seamless frame. Given two aligned t-th RGB frames I_0^t and I_1^t as inputs, and the stitched frame I^t, a seam finding problem can be thought of as selecting a discrete label $L^t(i,j) \in \{0,1\}$ for all pixel locations (i,j) in the overlapping regions of two frames. The label determines which frame provides the pixel value at the pixel location (i,j), for example, $L^t(i,j) = 0$ indicates that the pixel at the location (i,j) is from the frame $I_0^t(i,j)$. This transition can be determined by the locations of dominant objects in the overlapping region. As such, we would like to create seam boundaries that avoid dominant objects.

Figure 4 describes the steps of the proposed energy map estimation algorithm. The object energy map is initially computed by the instance segmentation. The initial energy map is then combined with the energy associated with the previous frame.

1) OBJECT ENERGY

For object energy, instance segmentation results can be represented by a binary map $S^t(i,j)$, which indicates whether the pixel belongs to objects:

$$S^t(i,j) = \begin{cases} 1, & \text{if } (i,j) \in \Omega_o \\ 0, & \text{otherwise,} \end{cases} \tag{1}$$

where Ω_o is the set of pixel locations belonging to the objects in the overlapping regions. Allowing misalignment error in the overlapping regions, in practice, $S^t(i,j)$ is computed as the union of pixels that belong to any objects detected in either I_0^t or I_1^t:

$$S^t(i,j) = S_0^t(i,j) \vee S_1^t(i,j), \tag{2}$$

where the operation $\vee$ denotes a logical disjunction; S_0^t and S_1^t indicate the instance segmentation results corresponding to I_0^t and I_1^t, respectively. Note that any instance segmentation methods can be used to detect dominant object regions. In our experiment, we used a convolutional neural networks-based instance segmentation method YOLACT [34] in overlapping regions because of its efficiency.

2) TEMPORALLY CONSISTENT ENERGY

In the video stitching process, the stitched outputs often suffer from temporal inconsistency due to the presence of dynamic elements or varying exposures across frames. Therefore, to preserve the spatial-temporal consistency, we propose an efficient temporal smoothing process based on the feedback scheme. The proposed smoothing process maintains energy information over an arbitrary number of frames while reducing computational cost and memory requirements. Moreover,

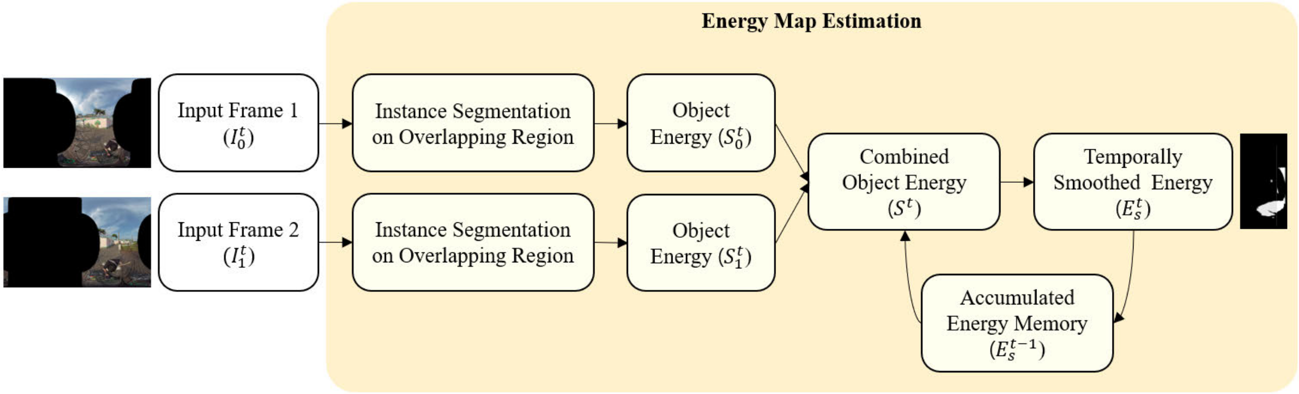

Energy Map Estimation
Input Frame 1 (I_0^t)
Input Frame 2 (I_1^t)
Instance Segmentation on Overlapping Region
Instance Segmentation on Overlapping Region
Object Energy (S_0^t)
Object Energy (S_1^t)
Combined Object Energy (S^t)
Temporally Smoothed Energy (E_s^t)
Accumulated Energy Memory (E_s^{t-1})

FIGURE 4. **Description of the proposed energy map estimation algorithm. Given two input images, the proposed algorithm estimates spatial-temporally consistent object energy function.**

the contribution of each frame to the output energy map can be easily controlled by adjusting an attenuation factor.

We formulate the temporally smooth energy map $E^t(i,j)$ at the pixel location (i,j). The function consists of object energy augmented with the energy associated with previous frames, as follows:

$$E_s^t(i,j) = \alpha E_s^{t-1}(i,j) + (1-\alpha)S^t(i,j), \tag{3}$$

where $E_s^{t-1}(i,j)$ is the energy map of the $(t-1)$-th frame. The attenuation constant α decides the contribution of the previous energy map to form the current energy map. Owing to the recurrent structure of (3), the energy map of E_s^t is interpreted as the weighted sum of a set of object energies (*i.e.*, $\{S^0(i,j), \cdots, S^t(i,j)\}$). That is, the current energy map memorizes objects moving through the previous frames. This is an efficient structure to combine energy maps over an arbitrary number of frames and smooth temporal changes without storing previous frames. However, since memorizing long-term object energy may produce invalid seam boundaries by accumulating the trajectories of all objects passing through the overlap, we regulate the contribution of the energy map of previous frames as:

$$E_s^t(i,j) = \begin{cases} E_s^t(i,j), & \text{if } E_s^t(i,j) \geq \tau \\ 0, & \text{Otherwise,} \end{cases} \tag{4}$$

where τ is the threshold that defines a time window to accumulate energies. Object movement history is truncated after the defined time window by τ. In our implementation, we would like to form the smoothing process as a weighted average that weights the energy of the current frame more and gradually downweights the energy of the temporally distant frames. Thus, τ was set to $\alpha^N(1-\alpha)$, where N is the number of frames to maintain energies, and α was set to 0.1.

In addition to finding temporal coherency, we further alleviate spatial-temporal misalignment by applying dilation operations. This can help robustly handling object deformation artifacts that appear in the overlapping regions. Since dilating object energy implicitly gives the prediction of object movement, this leads seams to be formed avoiding objects that will appear in the future frames. Furthermore, the dilation would compensate for possible unstable object segmentation. We have found that the contribution of the temporal smoothing energy is maximal when segmentation over frames is unstable.

B. SEAM FINDING

Given the energy map (3) and (4), seam boundaries are computed as the minimal path of (4). A set of seam boundaries can be built by connecting pixels with consistent windings. Solving this minimal path problem even for an image is typically a computationally expensive and memory-intensive process [15], [20]. Thus, in order to accelerate the seam finding process for videos, we develop a way to efficiently approximate optimal seam boundaries in the spatial-temporal domain. The details of the seam finding steps are given in Figure 5.

1) SEAM BOUNDARY

The proposed algorithm finds the minimal energy seam boundary that stretches from the top of the overlapping region to the bottom, moving left or right by a few pixels from one row to the next. While traversing rows from the top to the bottom (*i.e.*, $i = 2, \cdots, H$) where the size of the overlapping region is $H \times W$, only the pixels in the row directly under are examined to efficiently find seam with minimal computations. Moreover, the next seam path through the energy surface is searched on a set of adjacent pixels of the previous seam location. Defining a function $V^t(i)$ as a seam path with the lowest energy on the surface of E_s^t in the overlapping region, this can be accomplished by using the greedy search [45], as follows:

$$V^t(i) = \underset{k}{\operatorname{argmin}} \left\{ E_s^t(i,k) \right\}. \tag{5}$$

To avoid abrupt seam changes over each row and to expedite the processing time, we only searched $k \in \{j-p, j, j+p\}$. j is the seam location of the previous row in the horizontal line, and p is the offset from j.

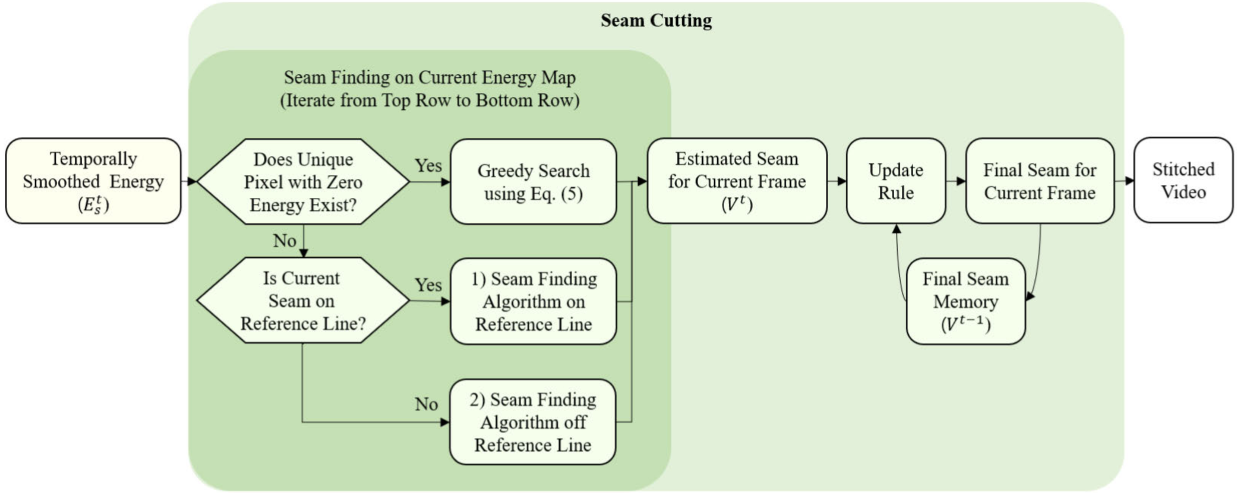

Seam Cutting
Seam Finding on Current Energy Map
(Iterate from Top Row to Bottom Row)
Temporally Smoothed Energy (E_s^t)
Does Unique Pixel with Zero Energy Exist?
Yes
No
Greedy Search using Eq. (5)
Is Current Seam on Reference Line?
Yes
No
1) Seam Finding Algorithm on Reference Line
2) Seam Finding Algorithm off Reference Line
Estimated Seam for Current Frame (V^t)
Update Rule
Final Seam for Current Frame
Final Seam Memory (V^{t-1})
Stitched Video

FIGURE 5. **Description of the proposed seam finding algorithm. Given the estimated energy map, the proposed algorithm finds seam boundaries by examining preattentive object locations behind the current row.**

In (5), we assign a minimal adjacent pixel as the seam only if the minimal pixel is unique and has never belonged to objects (*i.e.*, $E_s^t(i, V^t(i)) = 0$). Otherwise, the seam path is computed depending on whether the seam location of the previous row is on the reference line. The details are explained below in *a)* Seam Finding on Reference Line and *b)* Seam Finding off Reference Line. This is to form a seam boundary avoiding a contiguous group of object pixels once the dominant image is determined in the local area. Moreover, setting a reference line is effective to improve the temporal consistency in background regions by using the same reference for every frame. In practice, we used the vertical path connecting the midpoint of each row as the reference.

The details of the steps, when the minimal pixel is not unique and/or has belonged to objects, are as follows:

a) Seam Finding on Reference Line: When the current seam is on the reference line, we compute $V^t(i)$ taking into account the energy map behind the current row y, making it preventing possible seam creation that passes through objects. We examine the energies behind the y-th row on the left side E_L^t and the right side E_R^t from the reference line, where $\lceil W/2 \rceil$ is the horizontal location of the reference line and $\lceil \cdot \rceil$ is a ceiling operation. These energies are approximately estimated by summing E_s^t considering its distance from the reference line to the pixels of objects. Then, $V^t(i)$ is determined by finding the lower energy between E_L^t and E_R^t. Specifically, the preattentive energies E_L^t and E_R^t are defined as

$$E_L^t(y) = \sum_{j=1}^{\lceil W/2 \rceil - 1} \sum_{i=y+1}^{H} \left(E_s^t(i,j) + E_p(i,j)\right)$$

$$E_R^t(y) = \sum_{j=\lceil W/2 \rceil + 1}^{\lceil W/2 \rceil + \lfloor W/2 \rfloor} \sum_{i=y+1}^{H} \left(E_s^t(i,j) + E_p(i,j)\right), \qquad (6)$$

$$E_p(i,j) = (\beta_1\, |\text{i-y}| + \beta_2\, |j - \lceil W/2 \rceil|)\, \mathbb{1}\left(E_s^t(i,j) > 0\right), \qquad (7)$$

where $\mathbb{1}\{\cdot\}$ is an indicator function. The β_1 and β_2 control the contribution of the preattentive energy E_s^t by the distance from the reference line. Then, the seam is determined toward the direction that has minimal preattentive energy by comparing E_L^t and E_R^t:

$$V^t(y+1) = \begin{cases} \lceil W/2 \rceil - p, & \text{if } E_L^t(y) < E_R^t(y) \\ \lceil W/2 \rceil, & \text{if } E_L^t(y) = E_R^t(y) \\ \lceil W/2 \rceil + p, & \text{Otherwise.} \end{cases} \qquad (8)$$

The energy function (6) penalizes the pixels on high energy regions and the pixels far from the current seam location. This implies that the pixels on the path that objects passed through are less likely to be selected as the seam.

b) Seam Finding off Reference Line: When the current seam pixel (y, j) is off the reference line, we approach a simple seam path search strategy to find a minimal distance path, reducing processing time. In order to prevent the seam path from forming off the overlapped, we treat the seam path moves toward the reference line:

$$V^t(y+1) = \begin{cases} \min\{|\lceil W/2 \rceil - k|\}, \\ \quad \text{if } \min\left\{E_s^t(y+1,k)\right\} = 0 \\ \max\{|\lceil W/2 \rceil - k|\}, \\ \quad \text{Otherwise,} \end{cases} \qquad (9)$$

where $k \in \{j-p, j, j+p\}$.

Figure 6 illustrates the example of the proposed seam path finding algorithm, where each node corresponds to a pixel. The black nodes conceive the object's energy, and the white nodes are the pixels that have no object energy. The adjacent pixels in (5) are marked in the blue boxes. The seam path moves toward the left-most pixel in Figure 6(a) since the minimal energy is unique and zero. In contrast, the minimal seam

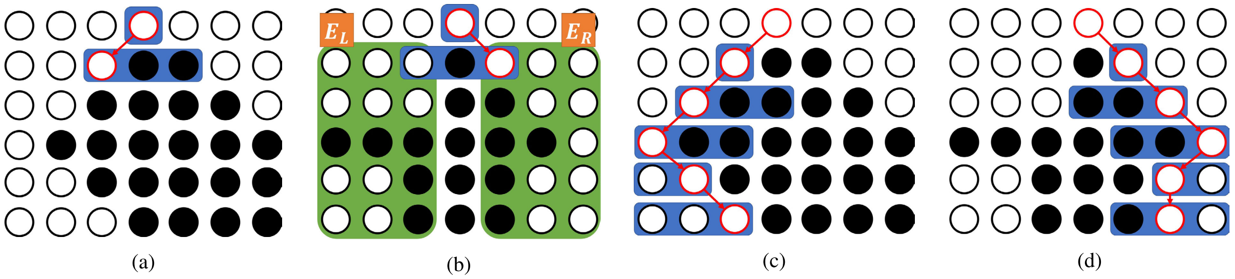


FIGURE 6. Examples of seam findings. Each node corresponds to a pixel, and the black nodes are the pixels that conceive the object's energy. The candidate pixels for seam are marked in blue areas in the case that p is 1. The areas shaded with green are the pixels to be considered for the E_L and E_R computation. The (a) and (b) are examples when the current seam is on the reference line. They are distinguished whether the candidate pixel is unique or not. The (c) and (d) show the examples when the current seam is off from the reference line.

path is computed by comparing the preattentive energies E_L^t and E_R^t in (b). In the examples of (c) and (d), the seam path moves toward the reference line avoiding contiguous object regions as explained in *b).*

2) UPDATE RULE

Although the initial seam is found in the temporally smoothed energy surface, undesirable seam changes could occur across frames. To prevent incurring unnecessary seam changes over frames, seam boundaries are updated only if necessary in our proposed video stitching pipeline.

To do this, we consider two indicators which correspond to energy variation and location changes along the seams. The energy variation I_e is defined, as follows:

$$I_e = \frac{|S_c - S_p|}{S_c}, \tag{10}$$

where S_c and S_p are the sum of energies in the current seam path and the previous seam path, respectively,

$$S_c = \sum_{i=1}^{H} E_s^t(i, V^t(i)),$$
$$S_p = \sum_{i=1}^{H} E_s^t(i, V^{t-1}(i)), \tag{11}$$

and V^{t-1} denotes the set of seam pixels at the $(t-1)$-th frame. Large I_e indicates that the sum of energies on the current seam's pixels differs largely from the sum of energies from the previous seam's pixels on the current energy map. It shows the suitability of changing seam from previous to current. In addition, the location change I_ℓ is measured by taking the differences of the seam location from the middle of the overlapping regions:

$$I_\ell = \left(\sum_{i=1}^{H}\left(V^t(i) - \lceil W/2 \rceil\right)\right)\left(\sum_{i=1}^{H}\left(V^{t-1}(i) - \lceil W/2 \rceil\right)\right). \tag{12}$$

A negative I_ℓ indicates that the current seam shifts the moving direction from the previous seam, whereas a positive I_ℓ implies no shift in the moving direction of seams.

Given two indicators, the seam is updated to $V^t(i)$ only if the following condition $\mathbb{U}$ is satisfied:

$$\mathbb{U} = ((I_e > \epsilon) \vee (I_\ell \geq 0)) \wedge \left(S_c < S_p\right), \tag{13}$$

where ϵ is the constant that measures energy variation. Otherwise, the seam is not updated and maintains the previous location.

IV. RESULTS AND DISCUSSION

A. REAL-WORLD DATA RESULTS

1) SETUP

In order to validate the proposed algorithm, we used the video sequences captured using Insta360 Pro 2 with a 6-camera setting, which results in 6 overlapping fields of view (See Figure 1). The sensor resolution of the camera is 7680×3840. The test sequences contain more than 3, 000 frames, and the frame rate of the test sequences is 60 fps. The sequences contain several moving objects, which is important for validating content preservation in the stitched videos. All test sequences were initially aligned using VRWorks [46]. The energy function in (1) was generated using YOLACT [34] with Resnet50-FPN [47] as the backbone. We temporally smoothed the energy function in (4) over 7 frames (*i.e.*, $N = 7$), and spatially smoothed it by applying 23×13 dilation kernel. The offset p in (5) was 1, and ϵ in (13) was set to 1. All experiments were performed on Intel i7-10700K 8-core processor with 64GB of memory and two NVIDIA GTX 1080 Ti.

The proposed algorithm was compared against the current seam computation algorithms – the algorithm in [48] ("Gradient"), the algorithm in [14] ("GraphCut") , and the algorithm in [16] ("Perception"). Although the evaluation of the stitching algorithms not designed for seam computation does not necessarily represent the seam computation performance accurately, we included the visual comparison against the state-of-art algorithms because it has become commonplace in the literature. The compared algorithms were Automatic Panoramic Image Stitching algorithm ("Autostitch") [4], natural image stitching with the global similarity prior algorithm ("GSP") [28], and robust elastic warping ("Robust ELA") algorithm [12]. Notice that we considered more state-of-art algorithms for comparisons rather than the earlier mentioned algorithms. However, most of them often failed to align the test sequences, which is probably because they were developed mainly for planar images.

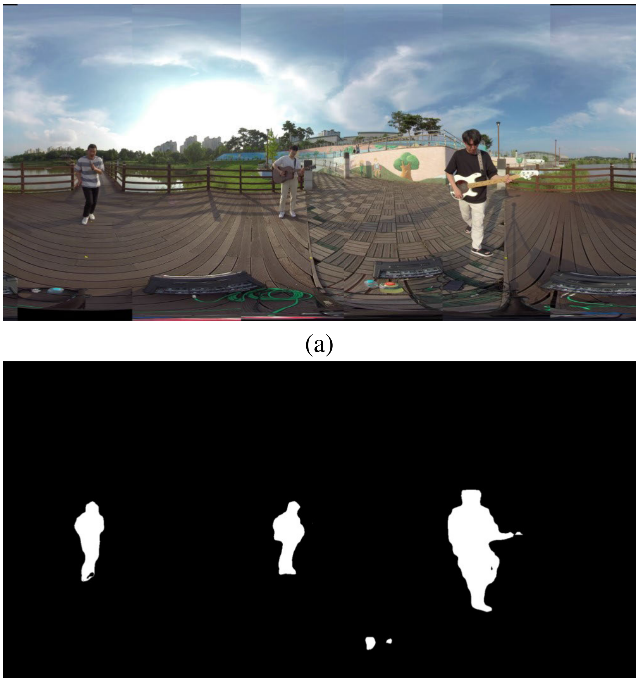


FIGURE 7. Example of an input image and the corresponding ground truth image. (a) Input image. (b) Ground truth image.

2) QUANTITATIVE EVALUATION

The proposed seam computation algorithm was designed, aiming at preserving foreground objects in the stitched regions. To ensure the purpose of content preserving, we evaluated the seam computation performance by counting the number of frames and pixels that seam boundaries pass through objects in the overlapping regions. By leveraging instance segmentation algorithms for quantitative comparison, we segmented object regions in the stitched sequence without applying any seam cutting and blending at the full resolution. Then, we manually inspected segmentation results and excluded the frames that had any incorrect or incomplete object segments. Figure 7 shows an example of a pair of the input image and ground truth. We note that any instance segmentation method can be used for generating ground truth.

3) RESULTS

The performance of seam computation is tabulated in Table 2 in terms of i) the number of frames including seams passed through objects in any overlap among 6 overlaps and ii) the average number of seam pixels on objects per overlap. The number of error frames produced by the proposed algorithm is reduced by about 85%, 85%, and 88% relative to Gradient, Graphcut, and Perception algorithms on average over all test videos. And the average number of error pixels per overlap produced by the proposed algorithm is reduced by about 20%, 20%, and 49% relative to Gradient, Graphcut, and Perception algorithms on average over all test videos' error frames. These results verify that the proposed algorithm can produce the seam that preserves dominant objects in the overlapping regions.

TABLE 2. The comparisons of seam computation performance for different algorithms. The performance is measured in terms of the number of frames, including seams passed through objects (# frames) and the average number of seam pixels on objects (# pixels) among 7680 × 3840 pixels.

Test Seq.	Error Metric	Ours	Gradient [48]	Graphcut [14]	Perception [16]
Video1	# frames	**3**	67	69	94
	# pixels	401.00	**191.45**	212.91	288.00
Video2	# frames	**34**	77	79	148
	# pixels	**424.35**	542.78	539.92	859.00
Video3	# frames	**10**	111	100	127
	# pixels	**280.30**	400.47	419.27	818.00
Video4	# frames	**10**	125	132	114
	# pixels	**147.80**	539.26	515.94	631.00

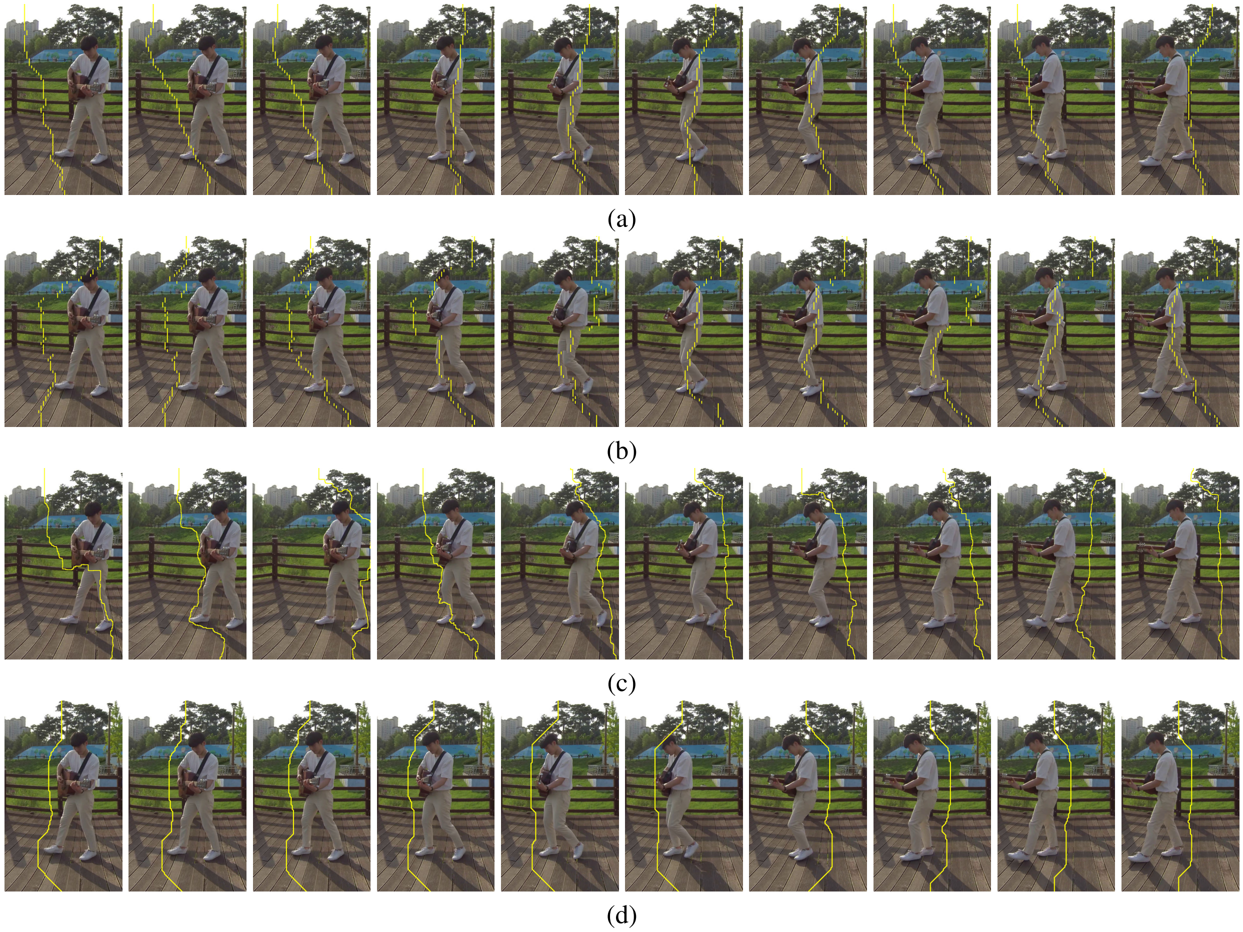
TABLE 3. The comparisons of seam computation time. The proposed algorithm and the comparison algorithms were implemented using C++, except for the Perception algorithm implemented using MATLAB.

	Gradient [48]	Graphcut [14]	Perception [16]	Proposed
Average processing time per frame (ms)	14.4	166.8	318619.8	**8.7**

We also provide a visual comparison of seam changes to demonstrate the effectiveness of temporal consistency, as shown in Figure 8. The interval between frames shown in the figure was set to 100 *ms*. By comparisons, the seams produced by Gradient algorithm [48] and Graphcut algorithm [14] tend to pass through the object in the overlapping region. This is partly due to the fact that they only use low-level features of the input images, not semantics. The seam produced by the proposed algorithm well maintains temporal consistency while other algorithms yield frequent seam changes across frames since they were designed without considering temporal consistency.

Table 3 reports the processing time of seam computations for different algorithms. The comparing algorithms were implemented using C++ except for the Perception algorithm implemented using the MATLAB API. As demonstrated in Table 3 and Figure 8, the proposed algorithm improves time efficiency due to its structural simplicity with better visual quality. The proposed algorithm requires $\mathcal{O}(\log(n))$ on computational complexity whereas Graphcut and Perception algorithms require $\mathcal{O}(n \log(n))$ [14]; the Gradient algorithm requires $\mathcal{O}(n)$, where n is the number of pixels in the overlapping region examined.

In addition to the quantitative and qualitative comparison of the seam computation algorithms, the visual comparisons against Autostitch, GSP, and Robust ELA algorithms are shown in Figure 12. As observed in the example figures, the proposed algorithm generates visually plausible results, whereas Autostitch and GSP often produce objects that suffer


(a)

(b)

(c)

(d)

FIGURE 8. Visual comparison of the estimated seam from 961-th frame to 1015-th frame for the different algorithms every 6 frames. (a) Gradient [48]. (b) Graphcut [14]. (c) Perception [16]. (d) Proposed. The seam produced by the proposed algorithm well maintains temporal consistency while other algorithms yield frequent seam changes across frames.

from the ghosting effect in the overlapping regions. The objects in the results generated by Robust ELA are likely to be structurally and perspectively distorted.

B. ABLATION STUDY

To validate the effectiveness of the spatial-temporally consistent energy function and seam update rule, we study the individual contribution of each step described in Section III-A and Section III-B by disabling one or more of them. They are finding seam based on the energy function in (4) without any temporal smoothing and update rule ("O"), finding seam based on the energy function in (4) with temporal smoothing ("O+T"), finding seam using the energy function of (4) and the update rule in (13) ("O+U"), and finding seam by the proposed algorithm ("O+T+U").

TABLE 4. The ratio of dominant image transition within a given interval (*ms*) for different combinations of the proposed algorithm. The combinations are finding seam based on the energy function without any temporal smoothing process ("O"), finding seam based on the energy function with temporal consistency filtering ("O+T"), finding seam using the energy function with the update rule in (13) ("O+U"), and finding seam by the proposed algorithm ("O+T+U").

	$[0, 50]$	$(50, 100]$	$(100, 200]$	> 200	$[0, 200]$
O	27%	15%	17%	41%	59%
O + U	24%	11%	15%	50%	50%
O + T	7%	5%	18%	70%	30%
O+T+U	1%	0%	12%	87%	13%

We measured how frequently dominant image transition occurs within a short interval – $50ms$, $100ms$, and $200ms$ on four different combinations. The dominant image is the image that contributes to more than 50% of an overlapping region between two overlapping images. The temporal consistency is then measured in terms of the ratio of dominant image transition that happened given time interval over the total number of dominant image transitions across entire frames, which is reported in Table 4. The 59% of dominant image changes took place within $200ms$ when the seams were produced by the energy function that only includes object energy ("O"), yielding undesirable blinking in the output videos. The ratio of image transitions is lessened to 50% with update rule ("O+U") and to 30% with temporal smoothing process ("O+T"). Only

TABLE 5. **The ratio of dominant image transition with different numbers of frames accumulated (N) to form the proposed energy function.**

N / Interval (ms)	0	1	3	5	7	9	11
$(0, 50]$	25%	13%	12%	4%	1%	1%	1%
$(50, 100]$	12%	11%	5%	7%	0%	0%	2%
$(100, 200]$	15%	17%	17%	16%	13%	10%	8%

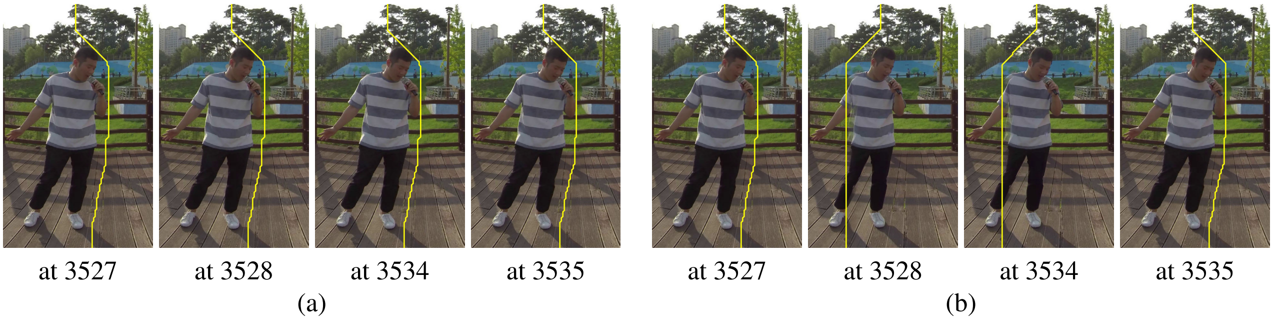


FIGURE 9. **Visual comparisons of the dominant image transition. (a) Computed seam boundaries by the proposed algorithm ("O+T+U"). (b) Computed seam boundaries by the proposed algorithm without update rule ("O+T"). The dominant image transition occurred over the short frames when the update rule was not applied. The numbers below images are the frame numbers.**

13% of image transition occurred within 200*ms* when applying both the temporally smoothing process and the update rule ("O+T+U"). The results verify that the proposed algorithm can generate the spatial-temporally consistent seam across frames.

We also studied the effectiveness of the length of the time window to accumulate energies, which is associated with the number of frames N. Table 5 reports the ratio of the dominant image transition given intervals (50*ms*, 100*ms*, 200*ms*) depending on the length of the time window. The seam changes within a short interval frequently occur without the temporal smoothing process (*i.e.*, $N = 0$). The ratios of dominant image transitions decrease as N increases up to $N = 9$, which implies that applying the temporal smoothing process is advantageous in preventing abrupt seam changes. The ratio of the interval 100*ms* slightly increases when $N = 11$. This indicates that accumulating the trajectory of moving objects for a long time would hurt temporal coherency because the energy function can be affected by the objects already passed.

In order to thoroughly examine the effectiveness of the update rule, we examined the seam changes of the case with update rule ("O+T+U") and the case without update rule ("O+T"). The examples of the produced seam by two cases are displayed in Figure 9. As demonstrated in the figure, the dominant image transition could occur without the update rule (See Figure 9(b)), proving the effectiveness of the design of the update rule.

C. BENCHMARK DATA RESULTS

1) SETUP

As publicly available datasets for multiview 360-degree images are limited, we modified a common benchmark video dataset, the KITTI [1] multiview-extension [49], [50] for evaluation. The test sequences were created from 100 training videos of the KITTI multiview dataset with 21-view stereo pairs. Each video consists of 21 frames per view, and the resolution of each camera is 375 × 1242. To delicately register the image pairs, we refined image alignment and obtained feature points using the oriented fast and rotated binary robust independent elementary features (ORB) [51]. The corresponding points were then matched by using the fast library

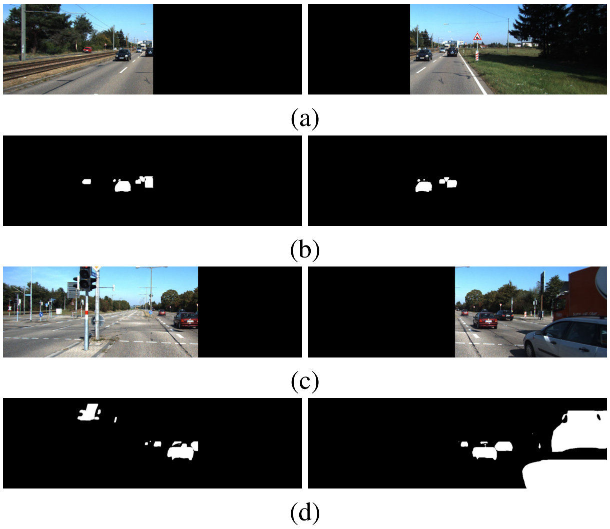


FIGURE 10. **Example of input images from the KITTI Multiview-Extension in Section IV-C. (a) Input images from different views. (b) Ground-truth images of (a). (c) Input images from different views. (d) Ground-truth images of (c).**

[1] Karlsruhe Institute of Technology and Toyota Technological Institute

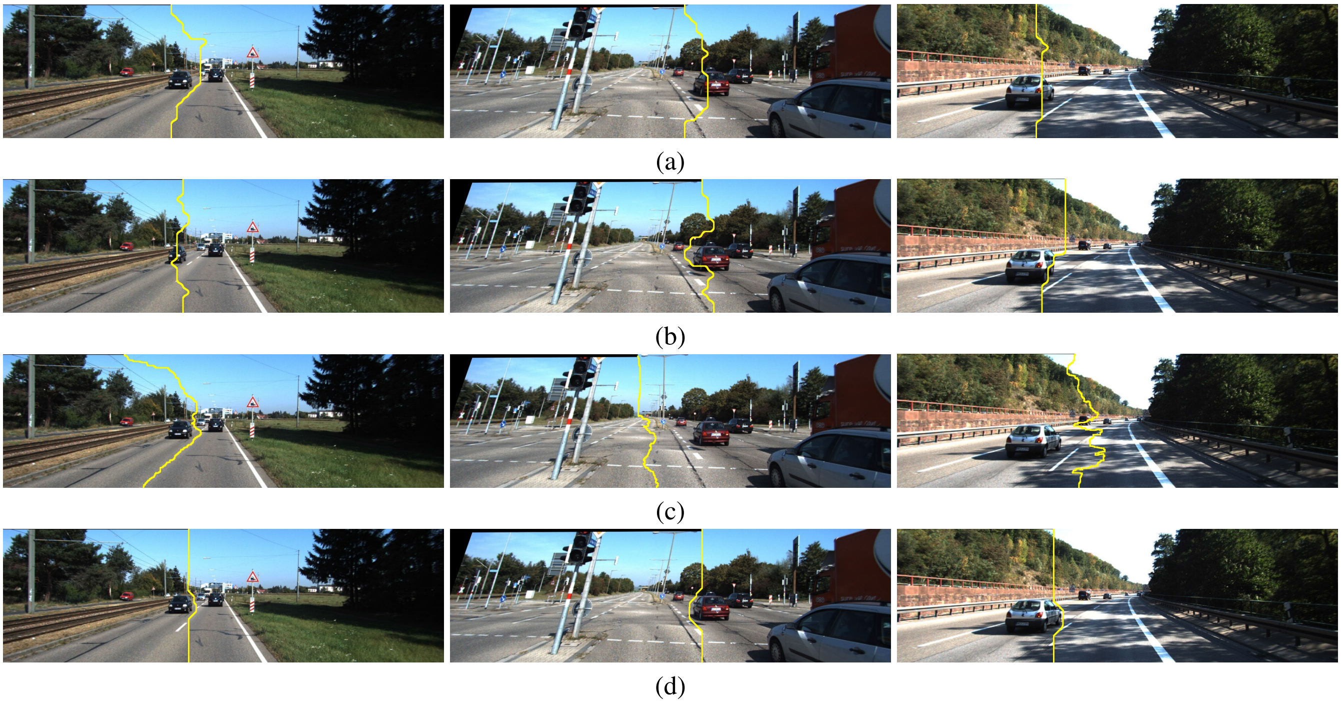


FIGURE 11. Visual comparison of the computed seam of the KITTI multiview-extension dataset for different algorithms. (a) Gradient [48]. (b) Graphcut [14]. (c) Perception [16]. (d) Proposed. The proposed algorithm produces seams that preserve objects in the overlaps.

for approximate nearest neighbors (FLANN) algorithm [52] and random sample consensus (RANSAC) [53]. We then used the 8-point algorithm [54] to find homography. The estimated homography at the first frame was applied to the remaining frames to keep temporal consistency for each test video. The resolution of each test video differs from the registration results. There are approximately 460, 000 pixels per frame in the stitched image region on average.

Ensuring the purpose of developing the proposed algorithm to find a seam that preserves contents, we selected overlapping regions centered at dominant objects after registration. Since the KITTI dataset was collected on an autonomous driving platform for depth estimation, its image pairs typically have wide scenes and huge overlaps. Note that the overlapping regions are almost 97.11% of the entire image region on average. This is undesirable to our video stitching problem, in which the primary goal is to generate stitched videos preserving dominant objects in the overlaps. Hence, the test frames were cropped so that the overlaps contain dominant objects in the scene, as shown in Figure 10.

The energy function in (1) was generated using YOLACT [34] with Resnet50-FPN [47] as the backbone. The energy function in (4) was smoothed with $N = 1$, and spatially smoothed it by applying 23×13 dilation kernel. The offset p in (5) was 1, and ϵ in (13) was set to 1.

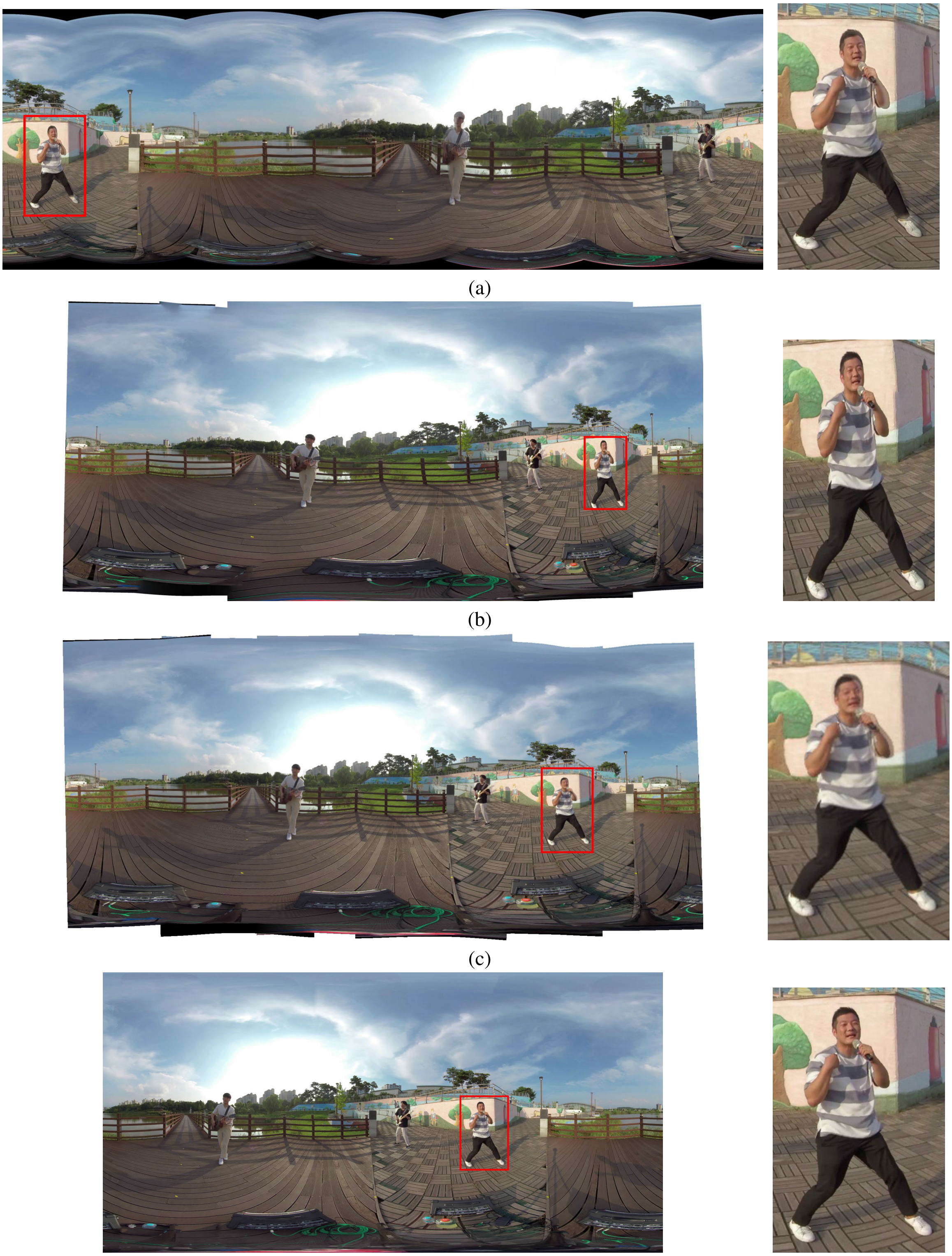
2) RESULTS

For quantitative evaluation, we generated ground truth object maps following the way described in Section IV-A (See Figure 10) and evaluated seam computation performance with the metrics used for the real-world test videos. We considered the algorithm in [48] (“Gradient”), the algorithm in [14] (“GraphCut”) , and the algorithm in [16] (“Perception”). Table 6 shows the quantitative evaluation results in terms of the number of frames including seam passed through objects and the average number of seam pixels on the objects. The number of error frames produced by the proposed algorithm is reduced by about 27%, 17%, and 5% relative to Gradient, Graphcut, and Perception algorithms. The average number of error pixels is reduced by about 43%, 45%, and 60% relative to Gradient, Graphcut, and Perception algorithms.

TABLE 6. The comparisons of seam computation performance for different algorithms in the KITTI multiview-extension [49], [50]. The performance is measured in terms of the number of frames including seam passed through objects (# frames) and the average number of seam pixels on objects (# pixels) over 2100 frames. The average total number of pixels is 460, 000 in the stitched image region.

Error Metric	Ours	Gradient [48]	Graphcut [14]	Perception [16]
# frames	**662**	907	801	697
# pixels	**31.1**	54.91	56.86	76.95

In addition to quantitative evaluation, we also visually demonstrate the performance of the proposed algorithm with computed seams against the compared algorithms in Figure 11. Although the Gradient and Graphcut algorithms were sensitive to detect the regions where the color variations


(a)

(b)

(c)

(d)

FIGURE 12. Visual comparison of stitching algorithms for the real-world dataset. (a) Autostitch [4]. (b) Robust ELA [12]. (c) GSP [28]. (d) Proposed. The proposed algorithm generates visually plausible results, whereas other algorithms often produce objects with the ghosting effect and structure distortion.

were distinct, they were likely to produce seams on dominant objects due to the lack of semantic information. The Perception algorithm could place seams on the small objects that were considered as nonsalient regions. In contrast, the proposed algorithm could produce seams that preserve object structure.

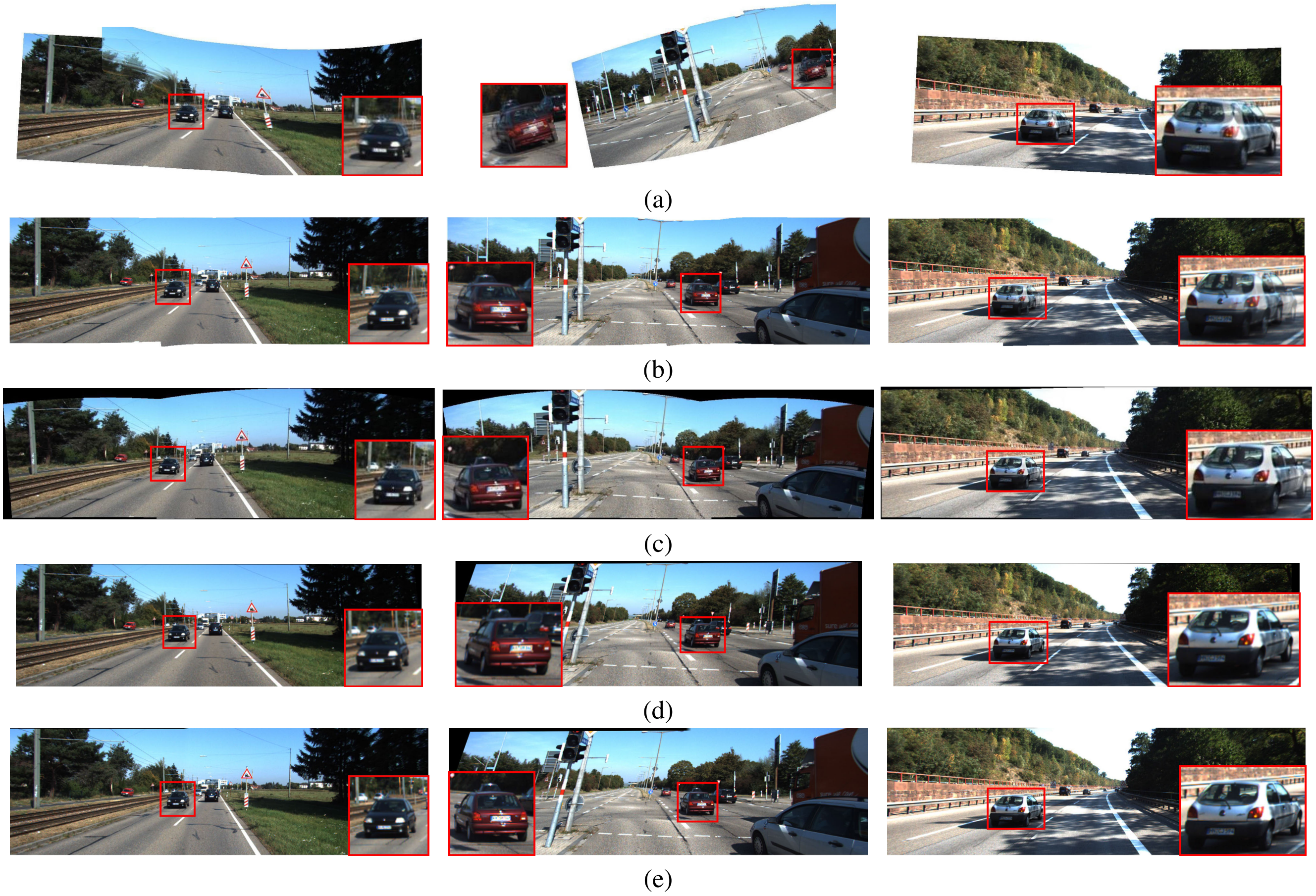


FIGURE 13. Visual comparison of stitching algorithms for the KITTI multiview-extension dataset. (a) Robust ELA [12]. (b) GSP [28]. (c) Autostitch [4]. (d) Photomerge from Photoshop [55]. (e) Proposed. The proposed algorithm can consistently generate visually pleasing results, whereas the stitched results by other algorithms often suffer from visual artifacts.

Figure 13 shows qualitative comparison against the state-of-art stitching algorithms and commercial software– Robust ELA [12], GSP [28], Autostitch [4], and Photomerge from Photoshop [55]. In this experiment, we used the test videos without registration as inputs to comparing algorithms except for Photomerge and the proposed algorithm. As demonstrated in the examples, the proposed algorithm could consistently produce visually pleasing stitched images while the stitched images generated by other algorithms suffered from visual artifacts. For example, the objects in the stitched results suffered from ghosting effect (See Figure 13(a) and (b)), the object in Figure 13(c) was occluded by the road and the object in Figure 13(d) was structurally distorted.

V. CONCLUSION

We proposed an efficient seam computation algorithm for 360-degree high-resolution multi-view videos. With the advancement of instance segmentation algorithms, the energy function was chosen to high weight to dominant objects in the overlapping regions. In order to maintain temporal consistency, the energy function was associated with the predicted moving directions of objects. Due to its structural simplicity, the proposed algorithm finds seam boundaries with minimal computations. The experimental results using the real-world video sequences and the benchmark dataset verified that the proposed seam computation algorithm reduces visual artifacts and produces visually pleasing results.

REFERENCES

[1] X. Zhang, Y. Zhao, N. Mitchell, and W. Li, "A new 360 camera design for multi format VR experiences," in *Proc. IEEE Conf. Virtual Reality 3D User Interfaces (VR)*, Mar. 2019, pp. 1273–1274.

[2] J. Tan, G. Cheung, and R. Ma, "360-degree virtual-reality cameras for the masses," *IEEE MultimediaMag.*, vol. 25, no. 1, pp. 87–94, Jan. 2018.

[3] R. Szeliski, "Image alignment and stitching: A tutorial," *Found. Trends Comput. Graph. Vis.*, vol. 2, no. 1, pp. 1–104, 2007.

[4] M. Brown and D. G. Lowe, "Automatic panoramic image stitching using invariant features," *Int. J. Comput. Vis.*, vol. 74, no. 1, pp. 59–73, Apr. 2007.

[5] C. Herrmann, C. Wang, R. S. Bowen, E. Keyder, M. Krainin, C. Liu, and R. Zabih, "Robust image stitching with multiple registrations," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 53–67.

[6] L. G. Brown, "A survey of image registration techniques," *ACM Comput. Surveys*, vol. 24, no. 4, pp. 325–376, Dec. 1992.

[7] A. M. K. Siu and R. W. H. Lau, "Image registration for image-based rendering," *IEEE Trans. Image Process.*, vol. 14, no. 2, pp. 241–252, Feb. 2005.

[8] M. Unser and P. Thevenaz, "Optimization of mutual information for multiresolution image registration," *IEEE Trans. Image Process.*, vol. 9, no. 12, pp. 2083–2099, Dec. 2000.

[9] Y. He, K.-H. Yap, L. Chen, and L.-P. Chau, "A nonlinear least square technique for simultaneous image registration and super-resolution," *IEEE Trans. Image Process.*, vol. 16, no. 11, pp. 2830–2841, Nov. 2007.

[10] G. Caner, A. M. Tekalp, G. Sharma, and W. Heinzelman, "Local image registration by adaptive filtering," *IEEE Trans. Image Process.*, vol. 15, no. 10, pp. 3053–3065, Oct. 2006.

[11] J. Gao, S. J. Kim, and M. S. Brown, "Constructing image panoramas using dual-homography warping," in *Proc. CVPR*, Jun. 2011, pp. 49–56.

[12] J. Li, Z. Wang, S. Lai, Y. Zhai, and M. Zhang, "Parallax-tolerant image stitching based on robust elastic warping," *IEEE Trans. Multimedia*, vol. 20, no. 7, pp. 1672–1687, Jul. 2018.

[13] K. Lin, N. Jiang, L.-F. Cheong, M. Do, and J. Lu, "SEAGULL: Seam-guided local alignment for parallax-tolerant image stitching," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2016, pp. 370–385.

[14] V. Kwatra, A. Schödl, I. Essa, G. Turk, and A. Bobick, "Graphcut textures: Image and video synthesis using graph cuts," *ACM Trans. Graph.*, vol. 22, no. 3, pp. 277–286, Jul. 2003.

[15] B. Summa, J. Tierny, and V. Pascucci, "Panorama weaving: Fast and flexible seam processing," *ACM Trans. Graph.*, vol. 31, no. 4, pp. 1–11, Aug. 2012.

[16] N. Li, T. Liao, and C. Wang, "Perception-based seam cutting for image stitching," *Signal, Image Video Process.*, vol. 12, no. 5, pp. 967–974, Jul. 2018.

[17] C. Herrmann, C. Wang, R. S. Bowen, E. Keyder, and R. Zabih, "Object-centered image stitching," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 821–835.

[18] A. Eden, M. Uyttendaele, and R. Szeliski, "Seamless image stitching of scenes with large motions and exposure differences," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, vol. 2, Jun. 2006, pp. 2498–2505.

[19] M. El-Saban, M. Izz, and A. Kaheel, "Fast stitching of videos captured from freely moving devices by exploiting temporal redundancy," in *Proc. IEEE Int. Conf. Image Process.*, Sep. 2010, pp. 1193–1196.

[20] W. Jiang and J. Gu, "Video stitching with spatial-temporal content-preserving warping," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2015, pp. 42–48.

[21] K.-Y. Lee and J.-Y. Sim, "Stitching for multi-view videos with large parallax based on adaptive pixel warping," *IEEE Access*, vol. 6, pp. 26904–26917, 2018.

[22] W.-S. Lai, O. Gallo, J. Gu, D. Sun, M.-H. Yang, and J. Kautz, "Video stitching for linear camera arrays," in *Proc. Brit. Mach. Vis. Conf. (BMVC)*, 2019, pp. 1–12.

[23] W.-Y. Lin, S. Liu, Y. Matsushita, T.-T. Ng, and L.-F. Cheong, "Smoothly varying affine stitching," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2011, pp. 345–352.

[24] J. Zaragoza, T.-J. Chin, M. S. Brown, and D. Suter, "As-projective-as-possible image stitching with moving DLT," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 2339–2346.

[25] G. Zhang, Y. He, W. Chen, J. Jia, and H. Bao, "Multi-viewpoint panorama construction with wide-baseline images," *IEEE Trans. Image Process.*, vol. 25, no. 7, pp. 3099–3111, Jul. 2016.

[26] T. Liao and N. Li, "Single-perspective warps in natural image stitching," *IEEE Trans. Image Process.*, vol. 29, pp. 724–735, 2020.

[27] N. Li, Y. Xu, and C. Wang, "Quasi-homography warps in image stitching," *IEEE Trans. Multimedia*, vol. 20, no. 6, pp. 1365–1375, Jun. 2018.

[28] Y.-S. Chen and Y.-Y. Chuang, "Natural image stitching with the global similarity prior," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2016, pp. 186–201.

[29] R. Sprengel, K. Rohr, and H. S. Stiehl, "Thin-plate spline approximation for image registration," in *Proc. 18th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc.*, vol. 3, Oct. 1996, pp. 1190–1191.

[30] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "SSD: Single shot multibox detector," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2016, pp. 21–37.

[31] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Adv. Neural Informat. Process. Syst. (NIPS)*, 2015, pp. 91–99.

[32] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2961–2969.

[33] X. Chen, R. Girshick, K. He, and P. Dollár, "TensorMask: A foundation for dense object segmentation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 2061–2069.

[34] D. Bolya, C. Zhou, F. Xiao, and Y. J. Lee, "YOLACT: Real-time instance segmentation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 9157–9166.

[35] X. Wang, R. Zhang, T. Kong, L. Li, and C. Shen, "SOLOv2: Dynamic and fast instance segmentation," in *Proc. Adv. Neural Informat. Process. Syst. (NIPS)*, 2020, pp. 17721–17732.

[36] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 580–581.

[37] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Region-based convolutional networks for accurate object detection and segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 1, pp. 142–158, Jan. 2016.

[38] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 779–788.

[39] S. Tripathi, G. Dane, B. Kang, V. Bhaskaran, and T. Nguyen, "LCDet: Low-complexity fully-convolutional neural networks for object detection in embedded systems," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jul. 2017, pp. 411–420.

[40] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 3431–3440.

[41] S. Zheng, S. Jayasumana, B. Romera-Paredes, V. Vineet, Z. Su, D. Du, C. Huang, and P. H. S. Torr, "Conditional random fields as recurrent neural networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1529–1537.

[42] B. Kang, Y. Lee, and T. Q. Nguyen, "Depth-adaptive deep neural network for semantic segmentation," *IEEE Trans. Multimedia*, vol. 20, no. 9, pp. 2478–2490, Sep. 2018.

[43] B. Kang and T. Q. Nguyen, "Random forest with learned representations for semantic segmentation," *IEEE Trans. Image Process.*, vol. 28, no. 7, pp. 3542–3555, Jul. 2019.

[44] X. Wang, T. Kong, C. Shen, Y. Jiang, and L. Li, "SOLO: Segmenting objects by locations," *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2019, pp. 649–655.

[45] T. Cormen and C. Leiserson, *Introduction to Algorithms* (MIT Electrical Engineering and Computer Science Series). Seoul, South Korea: Kyobobook, 2001.

[46] (2018). *Calibrating Stitched Videos With VRWorks 360 Video SDK*. [Online]. Available: https://developer.nvidia.com/blog/calibrating-videos-vrworks-360-video/

[47] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[48] (2019) OpenCV. https://docs.opencv.org/master/d7/d09/classcv_1_1detail_1_1SeamFinder.html

[49] M. Menze, C. Heipke, and A. Geiger, "Object scene flow," *J. Photogramm. Remote Sens.*, vol. 140, pp. 60–76, Jun. 2018.

[50] M. Menze and A. Geiger, "Object scene flow for autonomous vehicles," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 3061–3070.

[51] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: An efficient alternative to SIFT or SURF," in *Proc. Int. Conf. Comput. Vis.*, Nov. 2011, pp. 2564–2571.

[52] M. Muja and D. Lowe, "Flann-fast library for approximate nearest neighbors user manual," Dept. Comput. Sci., Univ. British Columbia, Vancouver, BC, Canada, vol. 5, 2009.

[53] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, Jun. 1981.

[54] R. Hartley and A. Zisserman, *Computation of the Fundamental Matrix F*, 2nd ed. Cambridge, U.K.: Cambridge Univ. Press, 2004, pp. 279–309.

[55] (2020). *Create Panoramic Images With Photomerge*. [Online]. Available: https://helpx.adobe.com/photoshop/using/create-panoramic-images-photome rge.html

TAEHA KIM is currently pursuing the M.S. degree in electrical and information engineering with the Seoul National University of Science and Technology, Seoul, South Korea. His research interests include computer vision and machine learning with a focus on image generation, reconstruction, and semantic segmentation.

SEONGYEOP YANG is currently pursuing the M.S. degree in electrical and information engineering with the Seoul National University of Science and Technology, Seoul, South Korea. His research interests include computer vision and machine learning with a focus on image segmentation, re-identification, and object detection.

BYEONGKEUN KANG received the B.S. degree in electrical and electronic engineering from Yonsei University, Seoul, Republic of Korea, in 2013, and the M.S. and Ph.D. degrees in electrical and computer engineering from the University of California at San Diego, La Jolla, CA, USA, in 2015 and 2018, respectively. He was a Postdoctoral Fellow with the Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, USA, from 2018 to 2019. He is currently an Assistant Professor with the Seoul National University of Science and Technology, Seoul. His current research interests include semantic segmentation, object detection, and human–machine interaction.

HEEKYUNG LEE received the B.S. degree in computer engineering from Yeungnam University, in 1999, and the M.S. degree in information and communication engineering from KAIST-ICC, in 2002. In 2002, she joined the Electronics and Telecommunications Research Institute (ETRI), South Korea, where she is currently serving as a Senior Member of Engineering Staff. She participated in TV-Anytime standardization and IPTV GSI Metadata standardization. She involved in the development of gaze tracking technology. She is also working on 360VR, AR, and MR. Her research interests include personalized service via metadata, HCI, gaze tracking, bi-directional advertisement and video content analysis, and VR/AR/MR.

JEONGIL SEO was born in Goryoung, South Korea, in 1971. He received the Ph.D. degree in electronics from Kyungpook National University (KNU), Daegu, South Korea, in 2005, for his work on audio signal processing systems. He worked as a Member of Engineering Staff with the Laboratory of Semiconductor, LG-Semicon, Cheongju, South Korea, from 1998 to 2000. He has been working as the Director of the Immersive Media Research Section, Electronics and Telecommunications Research Institute (ETRI), Daejeon, South Korea, since 2000. His research interests include image and video processing, audio processing, and realistic broadcasting and media service systems.

YEEJIN LEE received the Ph.D. degree in electrical and computer engineering from the University of California at San Diego, La Jolla, CA, USA, in 2017. She was a Postdoctoral Fellow in radiology with the University of California at Los Angeles, Los Angeles, CA, USA, from 2017 to 2018. She is currently an Assistant Professor with the Seoul National University of Science and Technology, Seoul, Republic of Korea. Her current research interests include computer vision, color image processing, and machine learning.