Shoot360: Normal View Video Creation from City Panorama Footage

Anyi Rao
MMLab, CUHK
Hong Kong, China
anyirao@ie.cuhk.edu.hk

Linning Xu
MMLab, CUHK
Hong Kong, China
linningxu@ie.cuhk.edu.hk

Dahua Lin
MMLab, CUHK
Hong Kong, China
dhlin@ie.cuhk.edu.hk

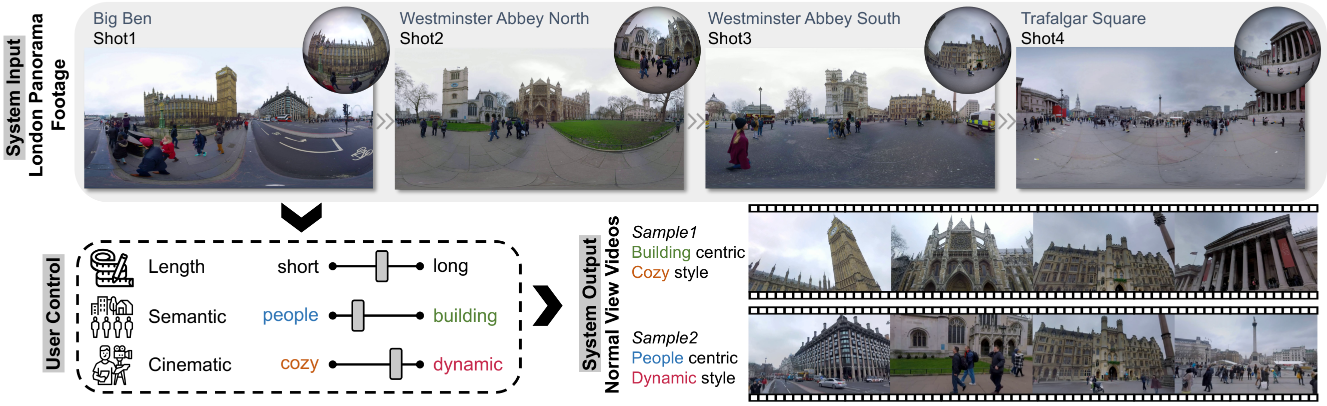

Figure 1: Given a collection of recorded panorama footage across the city, *Shoot360* takes in friendly high-level user control and produces high-quality normal view videos with desired content preference and cinematic styles.

ABSTRACT

We present *Shoot360*, a system that efficiently generates multi-shot normal view videos with desired content presentation and various cinematic styles, given a collection of 360 video recordings on different environments. The core of our system is a three-step decision process: 1) It firstly semantically analyzes the contents of interest from each panorama environment based on shot units, and produces a *guidance* that specifies the semantic focus and movement type of its output shot according to the user specification on content presentation and cinematic styles. 2) Based on the obtained *guidance*, it generates video *candidates* for each shot with shot-level control parameters for view projections following the filming rules. 3) The system further *aggregates* the projected normal view shots with the imposed local and global constraints, which incorporates the external knowledge learned from exemplar videos and professional filming rules. Extensive experiments verify the effectiveness of our system design, and we conclude with promising extensions for applying it to more generalized scenarios.

CCS CONCEPTS

• **Information systems** → Multimedia content creation.

KEYWORDS

video editing, panorama videos, neural networks

ACM Reference Format:
Anyi Rao, Linning Xu, and Dahua Lin. 2022. *Shoot360*: Normal View Video Creation from City Panorama Footage. In *Special Interest Group on Computer Graphics and Interactive Techniques Conference Proceedings (SIGGRAPH '22 Conference Proceedings), August 7–11, 2022, Vancouver, BC, Canada.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3528233.3530702

1 INTRODUCTION

To record what we have experienced during a city trip provides valuable memory for us. With the advent of portable 360 cameras, we can record almost everything that happened around us during the day trips. Interestingly, the panorama footage recorded by 360 cameras can roughly capture the entire 3D environment, providing us with a constrained 3D studio to creatively produce various kinds of normal view videos with accurate control. Still, the video editing process requires significant time and effort. The abundant source of materials contained in the panorama footage could bring difficulty for novice users to select and assemble an ideal video. The process of specifying the keyframes for each shot[1] in *spatial* dimension and connecting multiple shots in *temporal* dimension via professional software, such as Adobe Premiere and Insta Studio, is generally harsh for novices to get desired videos in time.

[1] A *shot* is an unbroken sequence of frames recorded from the same camera.

SIGGRAPH '22 Conference Proceedings, August 7–11, 2022, Vancouver, BC, Canada

ACM ISBN 978-1-4503-9337-9/22/08…$15.00
https://doi.org/10.1145/3528233.3530702

It is in great demand to develop an automatic system that can produce high-quality normal view videos from panorama footage with only a few adjustable preferences. Existing methods are mostly performed on single-shot videos, focusing on the salient person within the frame, neglecting other semantic elements, and not containing cinematic styles. Contrarily, our paper focuses on a more complete video creation pipeline that jointly considers the control of semantic elements and cinematic styles to bring out a multi-shot video that is harmonious both in content delivery and cinematic styles. A systematic comparison between our systems and existing works is shown in Tab. 1.

Shoot360 supports the creation of normal view videos from panorama footage with user specifications on the overall content preference and cinematic style as the input constraints. The system firstly brings out instruction *guidance* specifying the semantic focus (*e.g.*, person, building instance) and movement type (*e.g.*, static, zoom, rotate) of each shot based on the semantic analysis of the given footage and user preference. Instead of performing a per-frame decision process, the system determines the meaningful semantic instances at the shot level. Based on the selected contents, it then applies shot movement types inspired from professional filming to generate *candidates* to easily present cinematic styles. Finally, combining the criteria learned from exemplar normal view video data with incorporated professional filming rules, a full video is *aggregated* with local and global optimization goals to ensure its content deliver accuracy, visual smoothness and aesthetics.

The key contributions of our work are summarized as follows:

- An automatic pipeline to create normal view videos from panorama footage that allows user-friendly high-level control with content semantic preference and cinematic styles, which are interpretable and effective.
- A data-driven video ensemble strategy that learns the frame composition and shot aggregation from exemplar videos and incorporates filming rules.

Extensive experiments on user study and quantitative evaluations show the effectiveness of our tool. Novice users can easily obtain video creations that match their expectations, while the professionals can significantly save creation time to achieve desired effects with the aid of *Shoot360*.

Assumptions. In this paper, we mainly consider panorama footage recorded with stationary cameras, where each panorama shot records events that happened around a specific place. This assumption eliminates the ambiguity about the 3D scene environment derived from a moving camera if the camera trajectory information is unavailable. In addition, we mainly test videos for travel videos depicting city scenes, as they constitute a large portion of published online 360 videos, and are also rich and diverse in the contents. The relaxations on these two assumptions to accommodate general videos are discussed in the last section of the paper.

2 RELATED WORK

Video creation for various scenarios. Given the inconvenience of frame-based video processing for ordinary people, the ability to automatically create a video with desired styles and contents is of great importance. Besides focusing on several key steps in video processing [Liao et al. 2020; Shin et al. 2016], many researchers seek high-level automatic ways with easy-to-use user interactions [Arev et al. 2014; Truong et al. 2016] when creating videos. Wang et al. [2019] and Chi et al. [2020] create videos from themed text and web pages. Leake et al. [2017] and Truong and Agrawala [2019] focus on dialogue scenes and social conversations. Our system also belongs to this line of research. Targeting video creation from city travel panorama footage, it provides a high-level control interface for users to output desired contents.

Table 1: Comparison on different approaches’ features.

Methods	Scenario	User control	Semantic	Cinematic	Shot-level control	Multi-shot source
Su and Grauman [2017]	daily life	-	saliency	-	-	-
Hu et al. [2017]	daily life	-	saliency	-	-	-
Truong et al. [2018]	event, party	viewpoints	face & pose	-	✓	-
Truong and Agrawala [2019]	conversation	objects	face	-	✓	-
Wang et al. [2020]	daily life	viewpoints	saliency or people	-	-	-
Ours	city	semantic & cinematic	people & building	✓	✓	✓

Frame composition and shot aggregation. The quality of the generated video critically depends on the frame composition [Chang and Chen 2009; Pan et al. 2021] and shot aggregation. As for frame composition, researchers leverage the mutual relations [Li et al. 2020] and composition rules [Tu et al. 2020] to achieve better aesthetic quality. To handle shot aggregation, existing approaches typically rely on heuristics [Bruckert et al. 2021], Hidden Markov Models [Leake et al. 2017], dynamic programming [Wang et al. 2019] to encode cinematographic rules. Compared to the above, rather than processing each frame independently, we take a whole shot into account and compose frames based on semantic focus and cinematic concerns. On shot aggregation, we instruct the system *what to do* with the knowledge learned from real data, and *what should not do* following the filming rules.

Camera control for cinematography. How to control the camera is the key to both content selection and presentation with consolidated filming styles. Ideally, the control is performed in a 3D environment, either in real or virtual environments [He et al. 1996; Huang et al. 2019; Wu et al. 2018]. In virtual environments [Sitzmann et al. 2017], based on toric space [Lino and Christie 2015], Jiang et al. [2020] design an example-driven camera controller that learns camera behaviors from an example film clip. In real scenarios, most of the researchers target aerial settings [Xie et al. 2018; Yang et al. 2018] and formulate it as a robot control problem. The above are active control that requires either virtual environments or drone equipment, which is hard for an ordinary user to apply into practice. In this work, we focus on “passive control" where the output video comes from a post-round of filming in the recorded panorama environment.

Panorama to videos. From the perspective of system input and output, the most relevant task to ours is Pano2Vid [Su and Grauman 2017], which aims to capture normal field-of-view (NFOV) videos from panoramic videos. Most existing works focus on tracking the saliency parts in the video, which is usually a single shot [Hu et al. 2017; Kang and Cho 2019; Lai et al. 2017; Wang et al. 2020]. Su and Grauman [2017] start by sampling glimpses for each frame,

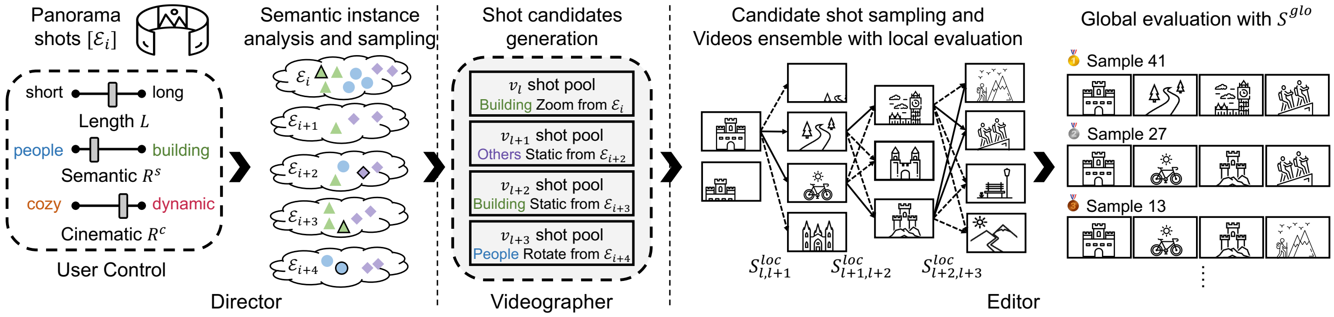

Figure 2: Overall pipeline of *Shoot360*. 1) The system firstly analyzes the semantic elements (people ●, building ▲, others ◆) contained in the panorama shots and combines the user specifications on the overall content coverage, cinematic styles and video length to sample the instruction guidance (*role of Director*). 2) Each instruction determines the source panorama shot, semantic focus, and movement type of the corresponding normal view shot. The system then brings out candidate videos for each shot following the instructions (*role of Videographer*). 3) Finally, by jointly associating the learned criteria from exemplar videos and hand-crafted criteria from filming rules, it samples shots from the candidates and aggregates the final videos in a local-to-global way (*role of Editor*).

and then score these candidates with a trained discriminator to construct a camera trajectory maximizing the scores. Later, Truong et al. [2018] explore multiple people scenes with face detection to generate multiple shots. Along with this effort, researchers focus on using similar techniques to improve the VR viewing experience [Liu et al. 2019; Pavel et al. 2017] or guide the videographers [Huang et al. 2020]. Our work focuses on the semantic meaningful units (*e.g.*, people and building in city scenarios) and aims to create a long video with multiple shots while maintaining the overall harmony with filming rules, which significantly relieves the burden of video creation for inexperienced users.

3 METHODOLOGY

Preliminaries for panorama data. The input to our system is a set of panorama shots recorded from various environments at different time. Each panorama image can be freely transferred to a normal view image via *gnomonic projection* [Snyder 1987] specified by a triplet of control parameters $q = (x, y, f)$. Here, the tuple $(x, y), x \in [-180°, 180°), y \in [-90°, 90°)$ denotes the horizontal/vertical position within the frame, revealing the viewing direction and the projection center; and $f = (f_x, f_y)$ denotes the field of view (FOV) along the x and y direction, depicting the range of content to be included in the projected image, where the ratio of f_x/f_y is fixed to the output video resolution. The creation process of each single normal view shot can be treated as the acquisition of a sequence of optimal projection parameters for the frames within a selected panorama shot. To generate a multi-shot video, each normal view shot is ordered according to the time order of its corresponding panorama shot.

System overview. To make the system self-contained and interpretable, we separate the creation process into three components, as shown in Fig. 2, reflecting the human film-making process with the roles of "*director*", "*videographer*", and "*editor*".

In Sec. 3.1, we introduce how the system analyzes the contents of interest from panorama shots with detected semantic instances, and combines the user's high-level control to generate *instruction guidance* for the entire video. Each instruction guidance specifies the source panorama, semantic focus and movement type for each shot, which plays the role of "director". Given one instruction, Sec. 3.2 explains how the "videographer" module applies corresponding movement type to the semantic elements for each panorama shot and creates a normal view video candidate pool. Finally, combining the prior knowledge learned from exemplar videos and the professional filming rules on shot design, the "editor" module outputs the final video by performing a selection and aggregation from the candidates with the aim to maximize a designed editing score for the full video, as elaborated in Sec. 3.3, to meet the users' requirements and looks natural, aesthetic, and meaningful.

3.1 Director Instruction Guidance

To provide simple and flexible user controls with high-level manipulation, we propose a semantic and cinematic instruction guidance module, which analyzes the panorama shots and generates a top-down instruction for overall content presentation, cinematic style, and video length.

3.1.1 Panorama shots analysis. With instance segmentation, each panorama shot $\mathcal{E}_i, i \in [0, I)$ obtains K_i^b buildings and K_i^p people, which forms the basic understanding of the semantic elements for each shot. [2] Applying this process for all shots, we acquire a total of K^b buildings and K^p people among I panorama shots. Though the kinds of semantic elements of interest may vary among different types of videos, in our experimental city travel scenario, we focus on two major semantic elements: *people* and *buildings*, which are of primary interest for a broad range of audiences and are also ubiquitous in many travel videos.

[2] For each panorama shot, instance segmentation is applied to its **keyframes**, which are average sampled. It densely samples the field of view in the keyframe, projects them to normal view, applies instance segmentation to avoid distortion, and restores the instance bounding box's position and size in panorama [Armeni et al. 2019].

3.1.2 Length control. It is a key feature for users to control the output video length. Instead of simply changing the player speed, we control the number of shots to achieve more natural length control. Each $\mathcal{E}_i$ is allowed to contribute different numbers of normal view shots to the final video. In practice, we restrict to generate $L \in [0, 2I]$ normal view shots from I panorama shots for efficiency and quality.

3.1.3 Content semantics control. In order to deliver the content information effectively in a visual-pleasant way, usually, each produced video shot should have its own semantic focus, *e.g.*, people, buildings or others. To control the overall content semantics percentage in the final video and meet the needs for various editing styles, we define the content semantics ratio $R^s = (R^s_p, R^s_b, R^s_o)$, $\sum R^s = 1$. $L \times R^s_p$ person instances and $L \times R^s_b$ building instances are sampled as the semantic focuses from the whole K^p people and K^b buildings respectively to form the final normal view videos with L shots. The rest $L \times R^s_o$ shots do not focus on people or buildings.

3.1.4 Cinematic style control via movement types. The underlying cinematic style of a video can significantly effect the viewing experience. Inspired from filming expert knowledge [Giannetti and Leach 1999], some basic cinematic styles such as “cozy" and “dynamic" can be implemented with different shot movement type ratio within a whole video, though more fancy styles should additionally consider the color tone, accompanied music etc, which is out of the scope of this paper. Similar to the above, we define the movement type ratio for three commonly used styles: static, zoom-in/out and rotate, $R^c = (R^c_s, R^c_z, R^c_r)$, $\sum R^c = 1$.

3.1.5 Instruction on each shot. With determined L, R^s, we sample semantic instances from the whole K^p person and K^b building instances, and each instance corresponds to one normal view shot. The shot order of normal view videos follows the original chronological order in the panorama shots. Through the above process, the source panorama shot and the semantic focus of each normal view shot are specified. Furthermore, we sample possible arrangements of each shot's movement type to obey the constraint of R^c.

3.2 Videographer Candidate Generation

Once the semantic focus, movement type, and source panorama shot of each normal view shot are specified, for each shot, there are still lots of candidates satisfying the requirements can be produced. To ease the control of the applied movement type and semantic focus, we lift the control on projection parameters from frame-level to shot-level, and apply three movement types, including *static*, *zoom in/out*, and *rotate* shots [Rao et al. 2020], to generate a limited number of candidates for each normal view shot in high quality. This treatment not only allows for the easy incorporation of professional filming techniques but also reduces the parameter space of control.

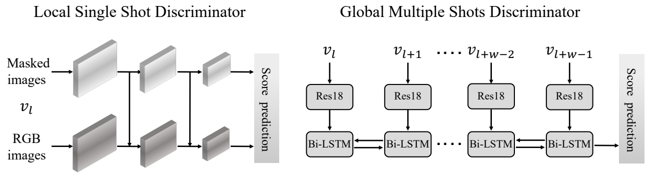

Figure 3: Network architecture for the discriminators.

3.2.1 Shot-level projection parameter. Recall that each normal view shot v_l, $l \in [0, L)$ can be uniquely controlled by a list of projection parameters $[q^t_l]$, $t \in [0, T_l)$ corresponding to T_l panorama frames, which are randomly sampled from the associated panorama shot and usually last for $1 \sim 20$s. Instead of controlling each frame, we perform on the temporal unit *shot*, which is easier to achieve more consistent visual aesthetics and effective content delivery. Hence, the parameter of each shot v_l can be represented as a tuple $(q^0_l, q^{T_l-1}_l, T_l, \alpha)$, where $(q^0_l, q^{T_l-1}_l)$ indicates the camera starting and ending point, T_l denotes the shot length, and α is the movement rhythm parameter. The projection for intermediate frames then follows an interpolation with the help of the easing function to achieve smoothness and simplicity,

$$q^t_l = \left(1 - \text{ease}\left(\frac{t}{T_l - 1}; \alpha\right)\right) q^0_l + \text{ease}\left(\frac{t}{T_l - 1}; \alpha\right) q^{T_l-1}_l$$
$$\text{ease}(\cdot; \alpha) = \begin{cases} \dfrac{\alpha^{(\cdot)} - 1}{\alpha - 1}, & \alpha \in (0, 1) \cup (1, \infty); \\ (\cdot), & \alpha = 1. \end{cases} \tag{1}$$

Note that in this easing function, α and T_l implicitly control the movement speed within a shot, α controls the movement rhythm. A large α makes the shot “first slow and then fast" and vice versa, and we use $\alpha \in \{0.1, 1, 10\}$ in our experiments. Given a fixed $(q^0_l, q^{T_l-1}_l)$, a large T_l slows the overall speed and vice versa.

3.2.2 Shot movement types to present semantics. 1) *Static shot* means that the camera is fixed to its semantic focus instance's centroid with no movement or rotation. The triplet control parameters $q = (x, y, f)$ keep the same in that shot. 2) In *zoom-in/out shot*, the camera gradually approaches/moves away from the centroid of the semantic focus instance. The FOV value f monotonous decreases for zoom in shot and increases for zoom out. It is also applied to the centroid of each instance. 3) *Rotate shot* keeps the f fixed while changing (x, y) values. It starts from the bottom (or top) of the instance and ends at the top (or bottom) for one instance or moves from one instance to another[3]. 4) Additionally, to allow generating shots without any semantic focuses, we apply “random" generation, which aims to simulate the view transitions when a person stands in a spot and casually looks around. Static, zoom and rotate shots take random positions (x, y) and follow the same strategy as above.

In the end, for each shot that is specified with semantic focus, movement type and source panorama shots, *Shoot360* enumerates all the possible instances, T_l, α to build up its candidates.

3.3 Editor Video Ensemble

With the prepared candidates of each shot for a given full video's instruction, the next step is to find an appropriate selection and combination of normal view shots from their corresponding candidate pool to assemble the final video. For each individual shot, it

[3] For the FOV value f, we set $f_x = 45°$ for static and rotate, f_x changes from $25°/45°$ to $45°/65°$ for zoom-in/out, $f_y = 1.77 f_x$ to meet the requirements of output resolution.

requires correct semantic focus and aesthetics for successive shots, which pursues harmony in content delivery and cinematic styles. We therefore design the ensemble evaluation metrics at three levels: individual shots v_l, neighboring shot pairs (v_l, v_{l+1}), and the full video $V = [v_l]$, $l \in [0, L)$. The criteria come from 1) well-trained discriminators with expert knowledge that are implicitly learned from massive exemplar videos, and 2) explicitly modeled filming rules in hard standards.

3.3.1 Data-driven discriminators. To evaluate whether the instruction-guided generated shots follow the real data patterns in terms of semantic elements and shot compositions, we resort to discriminator networks as the critic that learn to assign higher scores to the generated videos that are similar to the real exemplar videos.

Fig. 3 shows the network architecture of our *local* and *global* discriminators. 1) The local one handles a single shot with semantic focus on people and building separately and outputs S_l^p and S_l^b. It is implemented by a two-branch fusion convolutional network taking the keyframes as input, with ResNet18 [He et al. 2016] as backbones. 2) The global discriminator jointly considers a sequence of w shots to investigate the multiple shots combination and outputs $S_{l;w}^g$. It is implemented by a sequential discriminator that takes ResNet18 as the backbone and Bi-LSTM [Huang et al. 2015] as the classification head. Each shot is represented with five keyframes. Besides the RGB images, the local discriminator requires masked images [4] to better capture the framing composition for corresponding semantics.

3.3.2 Filming rule. While the criteria learned from real data can capture the high-level fidelity of the produced video, it is necessary to incorporate expert filming knowledge [Giannetti and Leach 1999] into the generated videos to ensure visual continuity and aesthetics among consecutive shots. The generated video gets a higher score $S_{l,l+1}^c$ if the following rules are satisfied, and vice versa. 1) The movement of frames is encouraged to be consistent with people's movement. 2) The consecutive shots are not recommended to have contradictory frame moving directions. 3) Furthermore, in terms of visual content, two neighboring shots are discouraged from sharing similar content, which might bring jump feelings for the audiences. The details are specified in the supplementary.

3.3.3 Aggregation with local and global constraints. With the above evaluation metrics, the goal turns to retrieve the optimal candidate from each shot pool to compose a video similar to real data patterns and obey filming rules. To speed up the aggregation, we divide the above criterion into the *local* and *global* stages. The local stage aims to maximize the local criteria S^{loc}, which only relies on two neighboring shots. Specifically,

$$S_{l,l+1}^{loc} = \sum_{j=l}^{l+1} (I_j^p S_j^p + I_j^b S_j^b) + \sum_{j=l}^{l+1} S_{l,l+1}^c, \\ I_j^{p/b} = \begin{cases} 1, & \text{if j-th shot's semantic focus is people/building} \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

Beam search [Reddy 1977] is applied here to achieve possible output videos maximizing S^{loc}. The searching process only keeps the most promising N nodes at each time step that reduces the memory requirements with a complexity of $O(NL)$.

In practice, each L, R^s and R^c specified in the control step will generate at most M full video's instructions. Finally, all these $N \times M$ samples are ranked with the global criterion to acquire the highest ones for selection. The global criterion jointly considers the score from the global discriminator and the filming rules for the entire shot sequence, forming the following score,

$$S^{glo} = \sum_{l=0}^{\frac{2(L-1)}{w}} S_{\frac{lw}{2};w}^g + \sum_{l=0}^{L-2} S_{l,l+1}^c, \tag{3}$$

where S^g is calculated within a w-long sliding window via a $w/2$ stride. [5] The final output is the one among $N \times M$ samples that achieves the highest S^{glo}.

3.4 Practical User Input

The aforementioned system opens direct control on overall length L, content semantics/shot movement type percentage R^s/R^c, which implements the high-level user specification on content presentation and cinematic styles. As some of these controls may not be straightforward for common users to master, we conduct user studies on practical interface design, which is specified in the supplementary. A key observation is that providing cinematic style presets is more friendly to common users as they are not experts on controlling professional shot movement types. Hence, the length and semantics control is instantiated as "adjustable bars" and the cinematic styles are implemented as "selection boxes" that provide constraints on shot movement types. To be specific, *cozy* style requires more static shots $R_s^c + R_z^c \geq 0.5$ and *dynamic* style needs more rotate shots $R_r^c + R_z^c > 0.5$. Clearly, it is potentially useful to develop more cinematic styles as presets that put constraints on other factors. For instance, *informative* style present more normal view video's shots to deliver richer information with $L > I$. Besides the above control parameters, we append an additional useful function in the interface to specify favored shots. It allows users to keep the desired shots in the final video, and the system will fix these shots in the video ensemble process.

4 EXPERIMENTS

4.1 Setup

4.1.1 Data. We test *Shoot360* on a gallery of 20 recorded 360 travel footage, covering various cities with different cityscapes. In each video, the shots are taken from multiple environments, including views such as famous landmark spots, busy streets, squares, and public gardens. The panorama footage comes from the YouTube VR Gorilla channel. The exemplar normal view videos come from YouTube Expedia travel guide video repositories, where we collected 228 videos with high numbers of likes[6].

4.1.2 Implementation Details. In the analysis of panorama shots, object detection for person and building was employed to label the semantic content information for each shot. Specifically, we apply Mask R-CNN [He et al. 2017] with ResNet50 backbone pretrained

[4] Masked image is the element-wise production of people/building segmentation mask and the RGB image.

[5] We set $N = 15$, $M = 1,000$, $w = 8$ in our experiments.

[6] https://www.youtube.com/c/Vrgorilla1, https://www.youtube.com/c/Expedia

Table 2: Processing time statistics of each step.

City	Input		*Shoot360*			Output	
	#Shot.	duration	candidate	ensemble w/ preview	render	#shots	duration
Amster.	24	12:40s	4.2s	25.1s	2:16s	12	51.3s
Barcelona	13	5:48s	3.6s	13.8s	1:39s	8	32.1s
Cuba	60	14:05s	9.3s	50.7s	3:26s	20	90.5s
London	16	5:13s	3.8s	14.8s	1:43s	8	41.1s
Paris	32	19:09s	5.8s	48.5s	2:35s	20	78.1s
Rome	17	6:03s	5.1s	15.7s	1:44s	7	38.3s

on Cityscapes [Cordts et al. 2016] to acquire each person instance segmentation mask. The building analysis is implemented by a Deeplab v3+ [Chen et al. 2018] with ResNet50 backbone pretrained on Ade20K [Zhou et al. 2019]. The discriminators output scores between $[0, 1]$ and are trained with $5,000$ real samples (label as 1) and $5,000$ fake samples (label as 0) using cross-entropy loss. Real samples are sampled from the collected exemplar videos and fake samples come from the random camera shooting in the panorama. The details are specified in the supplementary. *Shoot360* is tested on a laptop with an NVIDIA 1080 GPU, and the processing statistics of the selected panorama footage videos are shown in Tab. 2.

4.2 Evaluation of Modules

4.2.1 Candidate generation. To show whether the included contents of generated video indeed follow the instruction guidance, we apply people and building segmentation to count the frame coverage of semantic contents. We compare our candidate generation against "random" generation as specified in Sec. 3.2. Each method is tested on ten pieces of panorama footage, and for each footage we test ten different instructions covering all the semantic focuses and shot movement types. We sample the generated videos per second to check whether each frame contains the content in the instruction, and report the average content coverage percentage over all generated videos. From the quantitative results, our semantic-focused candidate generation (86.5%) holds 70% higher coverage than the random generation (11.7%), which verifies its effectiveness.

4.2.2 Video ensemble. To study the effectiveness of each score term adopted in the ensemble process, we conduct ablation studies on score terms and the results are shown in Tab. 3. We invite 10 professional filming major college students to form our *professional judge panel* to conduct pairwise quality comparisons with five scales, *i.e.*, much better to much worse between the full and ablated models. For each pair, we generate in total ten normal view videos from different panorama footage. Observed from Tab. 3, we find out that all the score terms play critical roles in the production of final high-quality videos, which proves the effectiveness of each score.

4.3 Comparison to Pro Manual Editing

To show how our tool performs in practice, we compare the video creation time and quality between the usage of Insta360 studio and *Shoot360* with Adobe Premiere Pro (shorten as Pr), which serves as an alternative video emsembler, in professional editing scenario.

Table 3: Pairwise quality comparison on various ablated models that do not use score terms in video ensemble.

Full vs.	much better	better	similar	worse	much worse
w/o S^b	71%	22%	7%	0%	0%
w/o S^p	67%	24%	9%	0%	0%
w/o S^c	82%	18%	0%	0%	0%

Table 4: User time cost and quality comparison with professional manual creation. Time is in the HMS format and quality score is in seven-point Likert scale very good (7)-neutral (4)-very bad (1).

User groups	Settings	User time cost	Quality score		
			Q_1^P	Q_2^P	Q_3^P
Professional	Insta360 studio	1:48:02s	5.8	5.7	5.6
	Ours cand. + Pr	0:43:19s	5.6	5.5	5.7
	Ours cand. + ensem. + Pr	0:24:40s	5.6	5.5	5.6
Novice	Ours cand. + ensem.	0:08:29s	5.2	5.5	5.2

Details are specified as, 1) Using Insta360 studio to generate each shot with keyframe annotation per 0.5 seconds and then connect each shot. 2) Ours cand. + Pr means manually selecting videos from our generated candidates and applying the trim connection with Pr. 3) Ours cand. + ensem. + Pr means that based on the above, it takes our assembled video as initialization for further editing.

The aforementioned settings are conducted by six professional video editors (two for each) with more than five years' experience. In addition, we invite three novices to use *Shoot360* to edit the same footage without any other tools. We report the average results of time cost and quality in Tab. 4. All the shots are connected with cut-in/out that do not hold additional transition effects. For fair comparisons, we ask them to create a one-minute video from the same twenty London panorama videos and count the time until they are satisfied with the outputs.

4.3.1 Time cost. As shown in the time column of Tab. 4, for the professional group using our generated candidates can save time to create videos with similar qualities, and taking our video ensemble results as initialization can further speed up the process, which saves 5× time. The novices take less than ten minutes to interact with our interface to edit a video with generally satisfying outputs.

4.3.2 Quality. To further show the quality of the generated videos, we prepare the following questions to evaluate the results from different perspectives on the professional judge panel (the same as Sec. 4.2). Q_1^P: Do you feel comfortable about the **video content**? Q_2^P: Is the **cinematic style** feeling of the video right? Q_3^P: How well is the **overall pace** of the video?

The results are summarized in the score columns in Tab. 4 and all items' variances are below 0.03. Notably, the videos created by novices achieve a very similar score on cinematic feeling Q_2^P to the professionals' creations. We consider this an indication that

Table 5: Survey statistics on the usage experience among the non-professional judge panel using seven-point Likert scale very good (7) - neutral (4) - very bad (1).

Item	Q_1^N	Q_2^N	Q_3^N	Q_4^N	Q_5^N
Score	6.0 ± 0.7	4.1 ± 0.9	6.4 ± 0.4	6.3 ± 0.5	6.2 ± 0.3

Shoot360 is able to perform sufficiently professionally for most ordinary users.

4.4 Usage Experience Statistic

The aforementioned experiments show the satisfying performance of our tools among professional's eyes. To get to know how ordinary users rate the usage experience, we conduct another user survey among 20 users aged between 20 to 50 without professional video editing skills as *non-professional judge panel* to try the tool and answer the following questions. Q_1^N: How would you rate the generated videos compared to your own travel videos? Q_2^N: How do you compare the results of *Shoot360* and Professionals? Q_3^N: How would you rate the usability of our tool? Q_4^N: Would you like to share the results with your friends? Q_5^N: Are you willing to use *Shoot360* if you take a 360 camera along a trip?

As shown in Tab. 5, the first two questions are about the quality of the generated video, we got 6.0 on Q_1^N and 4.1 on Q_2^N. It is a bit surprising that these novice users rate similar scores to ours and those created by professionals. In terms of the usability Q_3^N, most users feel generally good, and the standard deviation among users is low. Additionally, most of the participants are willing to share their created videos with friends, and get inspired to take a portable 360 camera in their future trips and create videos with *Shoot360*.

4.5 Comparison to Baseline

We pairwise compare our method with Pano2Vid [Su and Grauman 2017], default viewpoints on ten different city panorama footage following the same criteria Q_1^P, Q_2^P, Q_3^P on the professional judge panel (the same as Sec. 4.2.2/4.3) and Q_1^N on the non-professional judge panel (the same as Sec. 4.4). The results are shown in Tab. 6.

4.5.1 Settings. Since there is no work that shares the exact same setting as ours, we try our best to adapt and enhance Pano2Vid [Su and Grauman 2017] as our baseline. This is an improved version based on its previous work from the same authors [Su et al. 2016], though these two tools share the same name. It is designed for single-shot panorama videos, and we adapt it to our multi-shot setting with an additional sequential discriminator. Within one shot, Pano2Vid samples candidate keyframes with different azimuthal and polar angles every five seconds and leverages a pretrained discriminator to score these keyframes. It then constructs a trajectory over the keyframes by procedurally generating the current frame to maximize the scores based on its previous frame. Due to this reason, it is hard to produce one shot with a smooth movement trajectory in a long distance. Default uses the viewpoints provided by the 360 footage recorder and the video is shortened in the same way as our method. As the viewpoint is determined, there is no additional cinematic change in the output, which is impossible to track a moving object or show a huge building in the full figure.

Table 6: User study results of the percentage that ours are better than comparing methods among pro and non-pro.

Ours vs.	Q_1^P	Q_2^P	Q_3^P	Q_1^N
Pano2Vid	72%	96%	91%	84%
Default	82%	92%	82%	81%

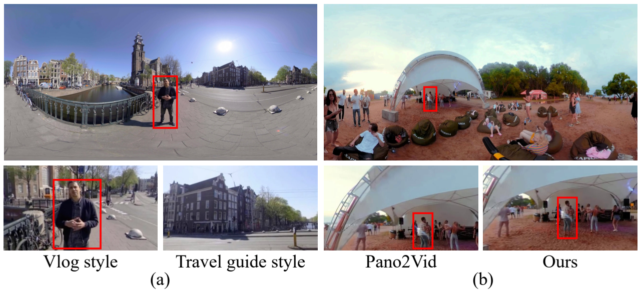

Figure 4: Qualitative comparison of (a) different exemplar videos tested on *Amsterdam* and (b) different methods on *Part* scenes.

4.5.2 Results. It is found that the professionals clearly tell the superiority of ours to Pano2Vid, and default viewpoints from three professionals' perspectives, especially in cinematic styles Q_2^P and overall pace of the generated multi-shot video Q_3^P. This verify the strength of our cinematic control and data-driven local-to-global video ensemble strategy. Similar observation can also be found from the non-professionals, where ours are better than baselines in over 80% cases.

4.6 Extension to Generalized Scenarios

In this section, we briefly discuss how our system generalizes to other scenarios and present the pairwise quality comparison between ours and Pano2Vid among professional judge panel (the same as Sec. 4.2/4.3/4.5) at the end of each case.

4.6.1 Different exemplar videos. Though the exemplar videos used above mostly are travel guide videos, we can also change them to another type to generate videos in different styles. We demonstrate with vlog style videos to show its generalization ability. For a fair comparison, the source 360 footage is kept the same. We crawl exemplar videos from YouTube using the keyword search in a "city name plus vlog" way. The qualitative results are shown in Fig. 4 (a). It is found that, the tool trained on vlog style videos tend to choose people-centric shot compared to the original one trained on travel guide style exemplar videos.

4.6.2 Various themes. Based on the above, we could further change the source footage and exemplar videos to other themes, *e.g.*, party and city aerial views, where we focus on people and building respectively. As shown in Fig. 4 (b), the qualitative results on party scenes prove that our method can better maintain the framing quality.

4.6.3 Moving cameras. Although the videos tested in the paper mostly come from stationary 360 cameras, *Shoot360* also works with relatively steady moving 360 cameras. Under this case, the movement of frames presented in the final videos comes from the superposition of the relative motion produced by *Shoot360* and the absolute motion of the camera itself. Still, benefiting from our discriminator, it can produce good views.

5 DISCUSSION AND CONCLUSION

In this work, we present *Shoot360* that takes panorama videos as input and outputs user desired normal view videos. Given a set of user-specified constraints, the system operates on the shot level and focuses on the semantic meaningful units to output desired contents, with the joint efforts of instruction guidance (director), candidate generation (videographer) and video ensemble (editor). Our user interface provides user-friendly control with an accompanying preview panel, which allows users to review the automatically generated results and refine the video. The comprehensive subjective and objective evaluations among professionals, non-professionals and machine statistics prove the effectiveness and usability of our *Shoot360.* And it also holds the following limitations and opportunities to be studied in the future.

Generalization. The footage theme tested in the paper is city travel, where people and buildings commonly appear and the recorded cameras are mostly stationary. The reason to choose it as the study target comes from the easy accessibility of data and mature detection algorithms for people and buildings. While our method achieves pretty nice results, we admit that these semantic element sets are not rich enough to support more diverse and fancy creations. The future work could generalize to broader coverage of themes and dynamic cameras and support more cinematic styles, excluding those videos that only have overly simple contents lacking of semantics, or are in bad qualities, *e.g.*, totally dark/blurred or too shaky. Specifically, individual modules in our workflow can be adapted accordingly, via 1) replacing the exemplar videos and learning different discriminators, as partially shown in our vlog style extension, 2) using continuously developed content detectors to acquire more interesting semantics. 3) stabilizing videos with SOTA methods [Guilluy et al. 2021]. 4) incorporating expert filming knowledge to enable more cinematic styles such as tense or musical.

User interaction. Additionally, the user interaction can be adapted to support specifying a person/event with bounding box selection to achieve instance-level control. This is a trade-off between human force and automation. Though the non-professional users who participate in our studies are satisfied with the designed semantic and camera style adjustable bars, some of them also agree it would be a nice optional function to support bounding box specification.

REFERENCES

Ido Arev, Hyun Soo Park, Yaser Sheikh, Jessica Hodgins, and Ariel Shamir. 2014. Automatic Editing of Footage from Multiple Social Cameras. *ACM Transactions on Graphics (TOG)* 33, 4 (2014), 1–11.

Iro Armeni, Zhi-Yang He, JunYoung Gwak, Amir R Zamir, Martin Fischer, Jitendra Malik, and Silvio Savarese. 2019. 3D Scene Graph: a Structure for Unified Semantics, 3D Space, and Camera. In *Proceedings of the IEEE/CVF International Conference on Computer Vision.* 5664–5673.

Alexandre Bruckert, Marc Christie, and Olivier Le Meur. 2021. Where to Look at the Movies: Analyzing Visual Attention to Understand Movie Editing. *arXiv preprint arXiv:2102.13378* (2021).

Yuan-Yang Chang and Hwann-Tzong Chen. 2009. Finding Good Composition in Panoramic Scenes. In *IEEE International Conference on Computer Vision.* IEEE, 2225–2231.

Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. 2018. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European Conference on Computer Vision.* Springer, 801–818.

Peggy Chi, Zheng Sun, Katrina Panovich, and Irfan Essa. 2020. Automatic Video Creation From a Web Page. In *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology.* 279–292.

Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. 2016. The Cityscapes Dataset For Semantic Urban Scene Understanding. In *IEEE Conference on Computer Vision and Pattern Recognition.* 3213–3223.

Louis D Giannetti and Jim Leach. 1999. *Understanding Movies.* Vol. 1. Prentice Hall Englewood Cliffs, NJ.

Wilko Guilluy, Laurent Oudre, and Azeddine Beghdadi. 2021. Video stabilization: Overview, challenges and perspectives. *Signal Processing: Image Communication* 90 (2021), 116015.

Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. 2017. Mask R-CNN. In *Proceedings of the IEEE International Conference on Computer Vision.* 2961–2969.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep Residual Learning For Image Recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.* 770–778.

Li-wei He, Michael F Cohen, and David H Salesin. 1996. The Virtual Cinematographer: a Paradigm for Automatic Real-time Camera Control and Directing. In *Proceedings of the Annual Conference on Computer Graphics and Interactive Techniques.* 217–224.

Hou-Ning Hu, Yen-Chen Lin, Ming-Yu Liu, Hsien-Tzu Cheng, Yung-Ju Chang, and Min Sun. 2017. Deep 360 Pilot: Learning a Deep Agent for Piloting Through 360 Sports Videos. In *IEEE Conference on Computer Vision and Pattern Recognition.* 1396–1405.

Chong Huang, Chuan-En Lin, Zhenyu Yang, Yan Kong, Peng Chen, Xin Yang, and Kwang-Ting Cheng. 2019. Learning to Film from Professional Human Motion Videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.* 4244–4253.

Hao-Juan Huang, I-Chao Shen, and Liwei Chan. 2020. Director-360: Introducing camera handling to 360 cameras. In *22nd International Conference on Human-Computer Interaction with Mobile Devices and Services.* 1–11.

Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional LSTM-CRF Models for Sequence Tagging. *arXiv preprint arXiv:1508.01991* (2015).

Hongda Jiang, Bin Wang, Xi Wang, Marc Christie, and Baoquan Chen. 2020. Example-driven virtual cinematography by learning camera behaviors. *ACM Transactions on Graphics (TOG)* 39, 4 (2020), 45–1.

Kyoungkook Kang and Sunghyun Cho. 2019. Interactive and Automatic Navigation for 360 Video Playback. *ACM Transactions on Graphics (TOG)* 38, 4 (2019), 1–11.

Wei-Sheng Lai, Yujia Huang, Neel Joshi, Christopher Buehler, Ming-Hsuan Yang, and Sing Bing Kang. 2017. Semantic-driven Generation Of Hyperlapse From 360 Degree Video. *IEEE Transactions On Visualization And Computer Graphics* 24, 9 (2017), 2610–2621.

Mackenzie Leake, Abe Davis, Anh Truong, and Maneesh Agrawala. 2017. Computational video editing for dialogue-driven scenes. *ACM Transactions on Graphics (TOG)* 36, 4 (2017), 130–1.

Debang Li, Junge Zhang, Kaiqi Huang, and Ming-Hsuan Yang. 2020. Composing Good Shots by Exploiting Mutual Relations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.* 4213–4222.

Junhua Liao, Haihan Duan, Xin Li, Haoran Xu, Yanbing Yang, Wei Cai, Yanru Chen, and Liangyin Chen. 2020. Occlusion Detection for Automatic Video Editing. In *Proceedings of the 28th ACM International Conference on Multimedia.* 2255–2263.

Christophe Lino and Marc Christie. 2015. Intuitive and Efficient Camera Control with the Toric Space. *ACM Transactions on Graphics (TOG)* 34, 4 (2015), 1–12.

Sean J Liu, Maneesh Agrawala, Stephen DiVerdi, and Aaron Hertzmann. 2019. View-dependent Video Textures for 360° Video. In *Proceedings of the Annual ACM Symposium on User Interface Software and Technology.* 249–262.

Zhiyu Pan, Zhiguo Cao, Kewei Wang, Hao Lu, and Weicai Zhong. 2021. TransView: Inside, Outside, and Across the Cropping View Boundaries. In *Proceedings of the IEEE/CVF International Conference on Computer Vision.* 4218–4227.

Amy Pavel, Björn Hartmann, and Maneesh Agrawala. 2017. Shot Orientation Controls for Interactive Cinematography with 360 Video. In *Proceedings of the Annual ACM Symposium on User Interface Software and Technology.* 289–297.

Anyi Rao, Jiaze Wang, Linning Xu, Xuekun Jiang, Qingqiu Huang, Bolei Zhou, and Dahua Lin. 2020. A Unified Framework for Shot Type Classification Based on Subject Centric Lens. In *Proceedings of the European Conference on Computer Vision.* Springer, 17–34.

D. Raj Reddy. 1977. Speech Understanding Systems: A Summary of Results of the Five-Year Research Effort. *Interim Report Carnegie-Mellon Univ., Pittsburgh, PA. Dept. of Computer Science.* (1977).

Hijung Valentina Shin, Wilmot Li, and Frédo Durand. 2016. Dynamic Authoring of Audio with Linked Scripts. In *Proceedings of the Annual Symposium on User Interface Software and Technology*. 509–516.

Vincent Sitzmann, Ana Serrano, Amy Pavel, Maneesh Agrawala, Diego Gutierrez, Belen Masia, and Gordon Wetzstein. 2017. How do people explore virtual environments? *IEEE Transactions on Visualization and Computer Graphics* (2017).

John Parr Snyder. 1987. *Map Projections: A Working Manual*. Vol. 1395. US Government Printing Office.

Yu-Chuan Su and Kristen Grauman. 2017. Making 360 Video Watchable in 2d: Learning Videography for Click Free Viewing. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 1368–1376.

Yu-Chuan Su, Dinesh Jayaraman, and Kristen Grauman. 2016. Pano2Vid: Automatic Cinematography for Watching 360 degree Videos. In *Proceedings of the Asian Conference on Computer Vision*.

Anh Truong and Maneesh Agrawala. 2019. A Tool for Navigating and Editing 360 Video of Social Conversations into Shareable Highlights.. In *Graphics Interface*. 14–1.

Anh Truong, Floraine Berthouzoz, Wilmot Li, and Maneesh Agrawala. 2016. Quickcut: An interactive tool for editing narrated video. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*. 497–507.

Anh Truong, Sara Chen, Ersin Yumer, David Salesin, and Wilmot Li. 2018. Extracting regular fov shots from 360 event footage. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*. 1–11.

Yi Tu, Li Niu, Weijie Zhao, Dawei Cheng, and Liqing Zhang. 2020. Image Cropping with Composition and Saliency Aware Aesthetic Score Map. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 12104–12111.

Miao Wang, Yi-Jun Li, Wen-Xuan Zhang, Christian Richardt, and Shi-Min Hu. 2020. Transitioning360: Content-aware NFoV Virtual Camera Paths for 360° Video Playback. In *IEEE International Symposium on Mixed and Augmented Reality*. IEEE, 185–194.

Miao Wang, Guo-Wei Yang, Shi-Min Hu, Shing-Tung Yau, and Ariel Shamir. 2019. Write-a-video: Computational Video Montage from Themed Text. *ACM Transactions on Graphics (TOG)* 38, 6 (2019), 177–1.

Hui-Yin Wu, Francesca Palù, Roberto Ranon, and Marc Christie. 2018. Thinking like a director: Film editing patterns for virtual cinematographic storytelling. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)* 14, 4 (2018), 1–22.

Ke Xie, Hao Yang, Shengqiu Huang, Dani Lischinski, Marc Christie, Kai Xu, Minglun Gong, Daniel Cohen-Or, and Hui Huang. 2018. Creating and Chaining Camera Moves for Qadrotor Videography. *ACM Transactions on Graphics (TOG)* 37, 4 (2018), 1–13.

Hao Yang, Ke Xie, Shengqiu Huang, and Hui Huang. 2018. Uncut aerial video via a single sketch. In *Computer Graphics Forum*, Vol. 37. Wiley Online Library, 191–199.

Bolei Zhou, Hang Zhao, Xavier Puig, Tete Xiao, Sanja Fidler, Adela Barriuso, and Antonio Torralba. 2019. Semantic Understanding of Scenes Through the Ade20k Dataset. *International Journal of Computer Vision* 127, 3 (2019), 302–321.