Intelligent Cinematography and Editing: Papers from the AAAI-14 Workshop

Autonomous Camera Systems: A Survey

Jianhui Chen, Peter Carr
Disney Research Pittsburgh, PA, USA
{jianhui.chen, peter.carr}@disneyresearch.com

Abstract

Autonomous cameras allow live events, such as lectures and sports matches, to be broadcast to larger audiences. In this work, we review autonomous camera systems developed over the past twenty years. Quite often, these systems were demonstrated on scripted stage productions (typically cooking shows), lectures, or team sports. We organize the discussion in terms of three core tasks: (1) *planning* where the cameras should look, (2) *controlling* the cameras as they transition from one parameter configuration to another, and (3) *selecting* which camera to put "on-air" in multi-camera systems. We conclude by discussing a trend towards more data-driven approaches fueled by continuous improvements in underlying sensing and signal processing technology.

Introduction

Autonomous camera systems are able to generate content of specific events that would otherwise not see the light of day because the immediate audience is not large enough to justify the cost of a human production crew. Using sensory data, the systems must comprehend what is taking place within the environment, and then configure one or more cameras to capture the actions from best possible vantage points (Pinhanez and Pentland 1995). Generally, these systems must autonomously solve three simultaneous problems:

1. **Planning**: Where should cameras look?
2. **Controlling**: How should cameras move?
3. **Selecting**: Which camera should be "on air"?

In this survey, we have highlighted key works in autonomous camera systems over the last twenty years. For clarity, we have organized the discussion along these three key tasks.

In general, the term "camera" may refer to a real camera (that captures light using an image sensor) or a virtual camera (that renders images using geometric and photometric information). Similarly, an environment may be real or virtual. Real cameras are restricted to real environments, but virtual cameras may be used in both virtual and real environments. This survey covers camera systems in real environments. However, before proceeding, we briefly describe similarities with respect to virtual environments.

Operating virtual cameras within virtual environments falls under the domain of computer graphics, and we refer reader to the excellent survey by Christie *et al.* (2008). In virtual environments, the planning, controlling and selecting algorithms typically have access to reliable high-level semantic information (compared to noisy sensor data generated in real environments). As a result, most algorithms devised for virtual environments are not directly applicable to real environments. However, there is still significant commonality. For example, virtual production systems do not have to contend with real world complications like latency and inertia, but must typically simulate these effects in order to synthesize aesthetic results. As a result, characterizing how cameras move in the real world is critical for generating video of virtual environments.

In real environments, virtual cameras synthesize new images by resampling data captured from real cameras. This "re-cinematography" technique (Gleicher and Masanz 2000; Gleicher and Liu 2007) can be used for a variety of purposes, such as format conversion and video stabilization. In this survey, we have covered both real and virtual cameras when applied to real environments. Although the planning and selection problems do not depend on the type of the camera, controlling real and virtual cameras is quite different: real cameras require feedback loops, while virtual cameras often employ filtering to approximate the intrinsic smoothing of inertia. As a result, we discuss 'planning' and 'selecting' in terms of generic cameras, and split the discussion of 'controlling' into separate sections for real and virtual cameras.

1. Planning

Planning addresses the problem of "where should cameras look?", and in real environments is solved by analyzing sensor data. Most real cameras are stationary robotic pan-tilt-zoom cameras, so the planning algorithm must output a desired pan angle, tilt angle, and zoom factor. Virtual cameras, on the other hand, are not constrained to stationary positions, and are instead often parametrized by a subregion of a real video frame to resample.

One of the earliest autonomous camera systems (Pinhanez and Pentland 1995) was demonstrated for a scripted cooking show. The system used the contextual information of the script to select the necessary computer vision algorithms to search for the expected events unfolding in the scene. Human-defined rules were then used to generate the appropriate virtual camera subregion based on the position of detected objects. More recently, Bayer *et al.* (2004) developed a robot photographer to take photographs at social events, such as wedding and conference receptions. The system used face detection to locate potential subjects in the scene. The robot then navigated based on a variety of factors such as the distance to the the subject, occlusion, and reachability. Finally, the system determined the optimal framing using established composition techniques, such as the rule of thirds.

Several autonomous video production systems have been deployed for recording lectures (Yokoi and Fujiyoshi 2005; Mavlankar et al. 2010; Pang et al. 2010) . The majority of these systems used a fixed camera to detect and track the lecturer, and employed raw tracking data to plan where the broadcast camera should look. Yokoi and Fujiyoshi (2005) employed a virtual camera to generate the final output by cropping the appropriate subregion of the fixed camera. Mavlankar *et al.* (2010) tracked the position of the lecturer in a fixed camera, and used a bimodal planning algorithm that could switch between preset camera configurations, and a dynamic one which followed the lecturer. Pang *et al.* (2010) trained an SVM classifier using features such as the estimated center-of-attention and distance between the center-of-attention and the center of the nearest chalkboard to predict when a camera should pan.

Camera planning has also been applied to more complex scenarios such as team sports. Ariki *et al.* (2006) tracked the locations of soccer players and the ball using fixed cameras. Additionally, they defined rule-based classifiers to recognize game situations, such as ‘penalty kick’ and ‘free kick’ based on the movement of the ball over a temporal window. These events were used to determine the zoom setting of the virtual camera, such as using a wide shot for a ‘free kick’. Chen *et al.* (2010) tracked basketball players and the ball using fixed cameras. The subregion for synthesizing a virtual camera was determined by a user-defined weighted sum of attentional interests (such as following a ‘star’ player). They also used rules such as the proximity of a visible salient object to the image center. Carr *et al.* (2013) tracked basketball players using fixed cameras, and determined the pan angle for a robotic camera by computing the centroid of the players' locations.

Most planning algorithms are based on tracked salient objects, such as faces or people. Occasionally, additional features such as audience gaze direction (Daigo and Ozawa 2004) and visual saliency (Pang et al. 2010) have been incorporated. In almost all cases, the planning algorithm follows the tracked object; often in conjunction with a set of hand crafted heuristics (Lino 2013). However, when multiple objects are within the scene, more complex planning may be required. Kim *et al.* (2012) tracked individual sports players in a calibrated broadcast view using particle filters, and extrapolated the player tracking data to a global motion vector field on the ground plane using Gaussian process regression. The authors showed how regions of convergence in the vector field correlated with actual broadcast camera movements. Alternatively, data-driven methods, such as SVM (Pang et al. 2010), neural networks (Okuda, Inoue, and Fujii 2009) and k-nearest neighbors (Dearden, Demiris, and Grau 2007); have been investigated to produced more complex camera planning without explicitly modeling the underlying process.

2. Controlling

Camera controlling pertains to transitioning a camera from its current parameter state to the desired configuration generated by the planning algorithm. In almost all applications, this task is a regulating process: the camera must move smoothly between fixation points in order to output aesthetic video, but also move fast enough to follow its planned state space sequence.

Kato *et al.* (1997) analyzed how cameramen operated their cameras in cooking shows and sports programs in order to figure out the exact characteristics of smooth camera motion — i.e. determining the speed at which panning is no longer aesthetic. The results showed an asymmetry in panning speed limits: acceleration can be higher when easing into a motion versus deceleration when easing out. In a subsequent study (Kato, Katsuura, and Koyama 2000), they found these characteristics only worked well in simple scenes: smoothly following a target with erratic movements was a substantially more difficult task.

Now that we have addressed how aesthetic camera motion has been characterized through the actions of real camera operators, we turn our attention to how these models are applied to virtual and real cameras.

Virtual Cameras Because virtual cameras resample recorded video, control algorithms for virtual cameras can be devised in an offline fashion — i.e. determining a smooth approximation to the planned signal can use information about both previous and future planned states. When Yokoi *et al.* (2005) used a virtual camera to follow a lecturer, the control algorithm smoothed the trajectory of the tracked subregion using temporal differencing and bilateral filtering. When the camera was undergoing a panning motion, they applied the learned parameters of (Kato et al. 1997) to regulate the position of the subregion. Chen *et al.* (2010) used a Gaussian Markov Random Field (MRF) to generate a smooth state-space trajectory of each virtual camera. Nieuwenhuisen *et al.* (2004) used a probabilistic roadmap to generate an initial estimate of linear segments which linked the current camera state to the desired future camera state. The path was refined by fitting circular arcs between segments and computing a smooth velocity plan which depended on path curvature limits. Grundmann *et al.* (2011) tracked KLT interest points within a recorded video to estimate the state space trajectory of the real camera. They then refined the estimated trajectory using a linear program to generate an equivalent smooth trajectory preferring constant position or constant velocity segments when prudent.

Real Cameras The task of moving a physical camera to keep an object of interest within the field of view is referred as *visual servoing* in robotics literature. Typically, this process involves a feedback loop examining the difference between where the camera was instructed to look, and where it actually looked. Stanciu *et al.* (2002) employed a proportion-only feedback control algorithm to adjust the pan-tilt angle of a camera mounted on the end of a human operated boom to keep a target object in the center of the camera image. Farag *et al.* (2005) used a proportion-only controller to position the centroid of detected image features near the center of the images of a stereo camera pair. Gans *et al.* (2009) used a task-priority kinematic controller to keep a set of interest points within the camera field of view. They showed how the mean and variance of the point cloud are independent objectives: pan-tilt values are modified to keep the mean near the center of the image, and zoom is regulated to keep the standard deviation within the image boundary. Carr *et al.* (2013) used a proportion-only controller to drive a robotic camera, but included delayed feedback from a virtual camera which resampled the raw video to generate a more stable output.

3. Selecting

If multiple cameras are deployed, the final step of an automated production system is to decide which camera should be "on air" at any given moment. Switching between cameras allows the system to output the best view for conveying a particular desired interpretation by the viewer, and also makes the output video more compelling. A shot must be maintained for a minimum duration, so all selection algorithms include some form of hysteresis filtering.

The majority of existing camera systems use a set of human-defined rules based on low-level tracking data to compute the shot quality of each vantage point. For example, Liu *et al.* (2001) interviewed five professional video producers and used their knowledge to create rules such as (a) do not make jump cuts, and (b) each shot should be shorter than a maximum duration. To implement these rules, they used a finite state machine to switch between three cameras (speaker's view, audience's view and overview). Doubek *et al.* (2004) used a network of fixed cameras to observe a subject moving through an office environment. A set of rules based on low-level tracking data were derived from cinematography conventions, such as using a 'long' shot to follow a moving target, and switching to a 'medium' shot when the subject came to rest. A viewpoint score was computed for each camera at each frame, and a 'resistance' factor was used to ensure a cut to any new vantage point only happened when there was a significant change in viewpoint score relative to the current "on-air" camera. Alternatively, Wang *et al.* (2008) used a Hidden Markov Model (HMM) based on camera motion features to choose the best virtual camera to show in a soccer video production system. Chen *et al.* (2010) also used an HMM to select the best virtual camera for basketball production, but incorporated the size and visibility of user selected 'salient objects' in the decision making process as well. Daniyal *et al.* (2011) proposed a multi-camera scheduling method based on view quality: the number of objects weighted by size and location. Larger sizes and locations within the area of interest had larger weights. They used a Partially Observable Markov Devision Process (POMDP) to minimize the number of inter-camera switches.

Recently, Chen *et al.* (2013) explored a purely data-driven approach by training a random forest classifier on field hockey tracking data to successfully recommend the best view to a human director. They used low-level features such as ball visibility, player locations and camera pan-tilt-zoom settings. Previous broadcasts were analyzed to extract sufficient training and testing data. In addition, the authors were able to show how the random forest could be trained for different directorial styles.

4. Summary

With the proliferation of affordable cameras, automated production is quickly becoming a viable method for recording events. Prototypes have been demonstrated with lectures and team sports[1]. The majority of these systems use computer vision algorithms to sense the environment, although other sensing modalities have been used in conjunction with vision-based techniques. All systems employ some form of object detection and tracking to plan camera movements for following salient objects. Occasionally, additional features such as saliency are used as well.

Systems employing virtual cameras are quite popular because, in this paradigm, none of the planning, controlling or selecting algorithms need to work within a realtime constraint. Furthermore, the algorithms can also operate in an offline fashion and incorporate information not only from current and previous events, but also future events. However, offline approaches are not viable for all applications.

In semi-controlled environments like cooking shows and lectures, the sensory data is usually less noisy and less complex compared to team sports (e.g. it is much easier to track a single lecturer in an indoor environment than multiple sports players in an outdoor setting). As a result, human-crafted rule-based methods for planning, controlling and selecting cameras have been quite effective in these settings.

In addition to low-level tracking data, most rule-based methods incorporate mid-level details as well, such as action recognition or game state information. In early work (Pinhanez and Pentland 1995), this context information was reliable human annotation. However, as technology has improved, machine learning techniques such as support vector machines, random forests and k-nearest-neighbour classifiers have been used to generate mid-level features from computer vision data. With state-of-the-art computer vision algorithms now able to estimate gaze and pose, new sets of mid-level features may soon be available for planning and selecting algorithm. Furthermore, data-driven algorithms may continue to replace rule-based methods. Using a data-driven approach, Chen *et al.* (2013) were able to mimic the cutting styles of different directors. Because the underlying process is not explicitly modeled, purely data-driven approaches have tremendous application flexibility. However, when prudent, it is often more efficient to code rule-based decisions manually, instead of learning them from large amounts of data.

[1] http://www.keemotion.com

Finally, as sensing technology improves, such as the introduction of 4K cameras, in tandem with more powerful signal processing algorithms and computing resources, autonomous systems will have more thorough and more reliable understandings of the world. As a result, future generation autonomous camera systems may have more algorithmic similarity in how they tackle the planning, controlling and selecting problems when compared to their virtual environment counterparts. In fact, as data-driven methods become more prevalent in real world autonomous camera systems, the characterization of these tasks may transfer to algorithms for virtual environments.

References

Ariki, Y.; Kubota, S.; and Kumano, M. 2006. Automatic production system of soccer sports video by digital camera work based on situation recognition. In *Multimedia, 2006. ISM'06. Eighth IEEE International Symposium on*, 851–860. IEEE.

Byers, Z.; Dixon, M.; Smart, W. D.; and Grimm, C. M. 2004. Say cheese! experiences with a robot photographer. *AI magazine* 25(3):37.

Carr, P.; Mistry, M.; and Matthews, I. 2013. Hybrid robotic/virtual pan-tilt-zom cameras for autonomous event recording. In *Proceedings of the 21st ACM international conference on Multimedia*, 193–202. ACM.

Chen, F., and De Vleeschouwer, C. 2010. Personalized production of basketball videos from multi-sensored data under limited display resolution. *Computer Vision and Image Understanding* 114(6):667–680.

Chen, C.; Wang, O.; Heinzle, S.; Carr, P.; Smolic, A.; and Gross, M. 2013. Computational sports broadcasting: Automated director assistance for live sports. In *Multimedia and Expo (ICME), 2013 IEEE International Conference on*, 1–6. IEEE.

Christie, M.; Olivier, P.; and Normand, J.-M. 2008. Camera control in computer graphics. *Computer Graphics Forum* 27(8):2197–2218.

Daigo, S., and Ozawa, S. 2004. Automatic pan control system for broadcasting ball games based on audience's face direction. In *Proceedings of the 12th annual ACM international conference on Multimedia*, 444–447. ACM.

Daniyal, F., and Cavallaro, A. 2011. Multi-camera scheduling for video production. In *Visual Media Production (CVMP), 2011 Conference for*, 11–20. IEEE.

Dearden, A.; Demiris, Y.; and Grau, O. 2007. Learning models of camera control for imitation in football matches. In *4th International Symposium on Imitation in Animals and Artifacts*.

Doubek, P.; Geys, I.; Svoboda, T.; and Gool, L. V. 2004. Cinematographic rules applied to a camera network. In *Proc. of Omnivis, The fifth Workshop on Omnidirectional Vision, Camera Networks and Non-classical Cameras*, 17–30.

Farag, A. A., and Abdel-Hakim, A. E. 2005. Virtual forces for camera planning in smart vision systems. In *Application of Computer Vision, 2005. WACV/MOTIONS'05 Volume 1. Seventh IEEE Workshops on*, volume 1, 269–274. IEEE.

Gans, N.; Hu, G.; and Dixon, W. 2009. Keeping multiple objects in the field of view of a single ptz camera. In *American Control Conference, 2009. ACC'09.*, 5259–5264. IEEE.

Gleicher, M. L., and Liu, F. 2007. Re-cinematography: improving the camera dynamics of casual video. In *Proceedings of the 15th international conference on Multimedia*, 27–36. ACM.

Gleicher, M., and Masanz, J. 2000. Towards virtual videography (poster session). In *Proceedings of the eighth ACM international conference on Multimedia*, 375–378. ACM.

Grundmann, M.; Kwatra, V.; and Essa, I. 2011. Auto-directed video stabilization with robust l1 optimal camera paths. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, 225–232. IEEE.

Kato, D.; Yamada, M.; Abe, K.; Ishikawa, A.; Ishiyama, K.; and Obata, M. 1997. Analysis of the camerawork of broadcasting cameramen. *SMPTE journal* 106(2):108–116.

Kato, D.; Katsuura, T.; and Koyama, H. 2000. Automatic control of a robot camera for broadcasting based on cameramen's techniques and subjective evaluation and analysis of reproduced images. *Journal of physiological anthropology and applied human science* 19(2):61–71.

Kim, K.; Lee, D.; and Essa, I. 2012. Detecting regions of interest in dynamic scenes with camera motions. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, 1258–1265. IEEE.

Lino, C. 2013. *Virtual camera control using dynamic spatial partitions*. Ph.D. Dissertation, Université Rennes 1.

Liu, Q.; Rui, Y.; Gupta, A.; and Cadiz, J. J. 2001. Automating camera management for lecture room environments. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, 442–449. ACM.

Mavlankar, A.; Agrawal, P.; Pang, D.; Halawa, S.; Cheung, N.-M.; and Girod, B. 2010. An interactive region-of-interest video streaming system for online lecture viewing. In *Packet Video Workshop (PV), 2010 18th International*, 64–71. IEEE.

Nieuwenhuisen, D., and Overmars, M. H. 2004. Motion planning for camera movements. In *Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on*, volume 4, 3870–3876. IEEE.

Okuda, M.; Inoue, S.; and Fujii, M. 2009. Machine learning of shooting technique for controlling a robot camera. In *Robotics and Biomimetics (ROBIO), 2009 IEEE International Conference on*, 1111–1116. IEEE.

Pang, D.; Madan, S.; Kosaraju, S.; and Singh, T. V. 2010. Automatic virtual camera view generation for lecture videos. Technical report, Stanford.

Pinhanez, C. S., and Pentland, A. P. 1995. Intelligent studios: Using computer vision to control tv cameras. In *IJCAI Workshop on Entertainment and AI/Alife*.

Stanciu, R., and Oh, P. Y. 2002. Designing visually servoed tracking to augment camera teleoperators. In *Intelligent Robots and Systems, 2002. IEEE/RSJ International Conference on*, volume 1, 342–347. IEEE.

Wang, J.; Xu, C.; Chng, E.; Lu, H.; and Tian, Q. 2008. Automatic composition of broadcast sports video. *Multimedia Systems* 14(4):179–193.

Yokoi, T., and Fujiyoshi, H. 2005. Virtual camerawork for generating lecture video from high resolution images. In *Multimedia and Expo, 2005. ICME 2005. IEEE International Conference on*, 4–pp. IEEE.