Visual Intelligence: Toward Machine Understanding of Video Content

Michael C. Burl, Russell L. Knight, Anthony C. Barrett
Jet Propulsion Laboratory, California Institute of Technology
4800 Oak Grove Drive, Pasadena, CA 91109

Abstract - *This paper describes progress toward developing visual intelligence algorithms (VI) that can produce human-like text descriptions (captions) from video inputs. Video frames are assumed to be generated according to an underlying "script' that specifies a camera model and the content and action in a scene. VI is formulated as the problem of recovering the script (or relevant portions of the script) given a sequence of video frames. Three types of scripts at different levels of abstraction are recovered: C-scripts contain object detections, poses, and descriptive information on a frame-by-frame basis; B-scripts assign persistent IDs to objects across frames and "smooth" frame-by-frame information; A-scripts provide a symbolic representation of video content using a sparse timeline in which Planning Executing Agent (PEA) graphical models (behavior snippets) are associated with agents in the scene. From the script representations, a compact text description (caption) of the action in the scene, as well as an envisionment (3D rendering) showing what the algorithm believes happened, can be generated. Scripts have been derived automatically and evaluated on a set of 240 publicly available video vignettes containing over 100,000 frames.*

Keywords: video understanding, natural language, text description, caption, surveillance, behavior recognition.

1 Introduction

On the TV show *Jeopardy*, IBM's *Watson* provided convincing proof that a machine could answer challenging natural language questions on par with, or even better than, human experts [1]. Emerging consumer products, such as *WolframAlpha* [2] and Apple's *Siri* [3], have also shown significant progress on this aspect of AI. We are interested in the related, but arguably more difficult, problem of visual intelligence (VI). *Can a machine, given only video input, reliably answer complex natural language questions about the content of a video and/or produce human-like text descriptions of what it has seen?* In this paper, we target automatic generation of text captions. Unlike the *Jeopardy* problem, simply looking up and conjoining readily available facts from the Internet is not likely to produce a good caption for a specific video input. *(*However, Barnard and Forsyth [4] did achieve some early success with text and image feature co-learning.)

A robust VI capability will provide the foundation for a number of new applications. For the military, placing VI-enhanced, persistent surveillance on unmanned air and ground vehicles could provide situational awareness without endangering personnel or requiring a large number of human eyes to monitor video feeds [5]. Similar benefits could be expected in law enforcement and homeland security. Other applications include human-robot interaction, video indexing and retrieval, sports analysis, retail intelligence, elder care, video games, and anonymization of video.

2 Approach

Our approach combines a front-end computer vision pipeline that leverages state of the art work in human detection, pose estimation, object recognition, and tracking with a back-end, AI-based plan recognition system that uses Planning Executing Agent (PEA) graphical models to recognize and reason about higher level behaviors.

The basic assumption in our approach is that video frames are generated according to a "script". One can imagine that such a script would specify a camera model and the content and action in the scene. VI is the inverse problem of recovering the script (or relevant portions thereof) given the sequence of video frames. Three types of scripts at different levels of abstraction are employed. At the top (agent) level, "A-scripts" built from Planning Executing Agent (PEA) graphical models are used to represent and reason about agent behaviors and video content symbolically. At the mid (tracking) level, "B-scripts" consisting of object pose trajectories provide a more literal encapsulation of what happened in the scene. At the low (detection) level, "C-scripts" consist of information extracted on a frame-by-frame basis.

Figure 1 illustrates the forward and inverse problems. The top path in the figure flows from right to left, taking a high-level A-script as input and producing video frames as output. One can envision this process occurring in two stages: first, the physical state of the actors and objects in the scene are altered over time (either in physical reality or in a simulation) according to the script; second, the world in its updated state is imaged.

The bottom (VI) path flows from left to right taking as input a sequence of frames and generating an estimate of the original

A-script. Computer vision attempts to invert the imaging process and recover a description of the state of the world versus time; the intermediate C-script and B-script representations describe the state in terms of the physical properties of the agents and objects, e.g., size, position, orientation, as well as any internal pose parameters versus time. Plan recognition techniques are used to recover the A-script that was responsible for this evolving state. The result is a higher-level representation of the video content that associates PEAs embodying specific semantic concepts with agents.

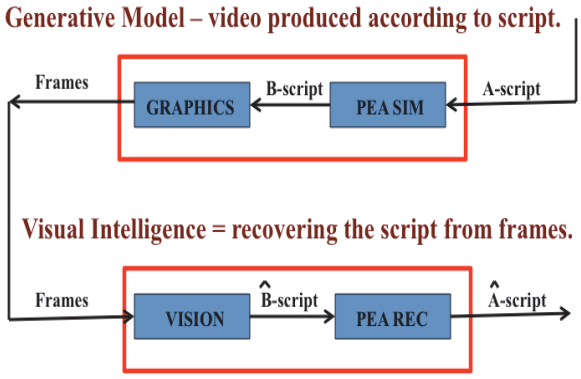

Figure 1. Video frames are produced according to an underlying script. Visual intelligence is formulated as the problem of recovering the script from the frames. Three levels of abstraction are used: A-scripts, B-scripts, and C-scripts. (C-scripts are used internally in the vision block.)

3 Related Work

The survey paper by Aggarwal and Ryoo [6], which served as the basis for a CVPR tutorial [7], provides an extensive discussion and taxonomy of work in activity recognition. Under their taxonomy, our approach would be classified as a hierarchical description-based approach.

The earlier work of Hongeng, Nevatia, and Bremond [8] is similar in some respects to ours. One difference is that we do not rely on temporal change detection to locate people and objects; we directly detect people and objects on a frame-by-frame basis making our approach suitable for a moving platform or a scanning camera. While their activity representation is based on 2D shape and trajectory features, our approach includes detailed human pose information allowing us, for example, to identify and describe colors of individual pieces of clothing. More importantly, our representations are *fully generative*, meaning we preserve enough information to create an envisionment (3D rendering versus time) that captures the essence of the original video. Other approaches from DARPA's Mind's Eye program can be found in [9-13].

A key difference between our work and many others is that others tend to use finite state machines and "Allen" temporal logic relations [14] to model and recognize complex behaviors involving multiple actors or actors and objects. In our approach we use PEAs, which are general graphical models in which the nodes represent states, and arcs represent transitions between states. PEAs subsume state machines and Markov models, but are more powerful because PEAs provide an explicit representation of resources through member variables. Resources are values that change by persisting in, entering, or exiting a state. Resource "gateways" can be established that prevent transition to other states until a resource collection requirement is met. Our system allows for the efficient encoding of such a resource collection state without adding linearly many states per resource, or exponentially many states for combined resources, as would normally be required by a simple finite state automaton. Our pre-compilation of resource goals allows us to plan, in constant time, to achieve required resource levels.

PEAs also provide more flexibility by associating arbitrary predicates with the arcs. For Hidden Markov Models (HMMs), which have been widely used for activity [15,16] and speech recognition [17], the transitions from state-to-state happen randomly according to some fixed probability distribution. In PEAs transitions are triggered by predicates, which can be complex (probabilistic, if desired) rules implemented as *general procedural computer code*, and as noted above may incorporate resource constraints. This flexibility enables PEAs to correctly model multi-agent interactions, e.g., the action of an agent can force another agent to make a state transition.

PEAs are hierarchical allowing complex behaviors to be built up from simpler behaviors. A node in a PEA graph can correspond to another PEA graph that represents a subordinate behavior. Figure 2 shows a PEA model for the verb GIVE, which makes use of subordinate concepts such as HOLD, APPROACH, RUN, and WALK.

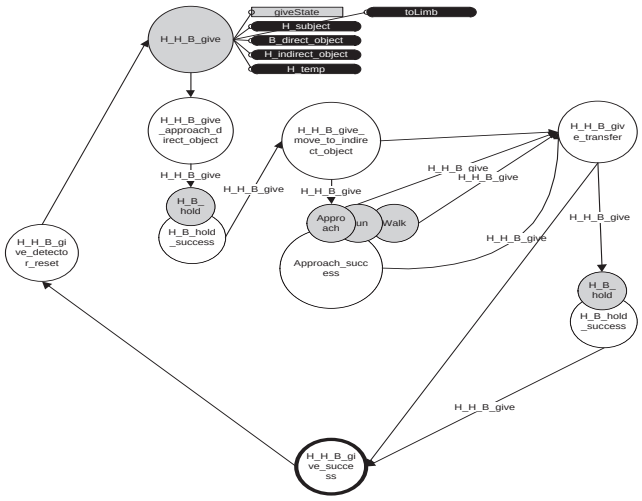

Figure 2. PEA model for GIVE in which subject is a human, direct object is a ball, and the indirect object is a human. The model for this complex verb is hierarchical making use of subordinate verbs such as HOLD, APPROACH, RUN, and WALK.

There are some preliminary VI efforts moving toward commercial applications in "video analytics" [18]. Some companies, such as Brickstream Corp. [19], provide analysis of overhead video taken in retail store scenarios, e.g., to detect shoplifting and to aid in marketing. Sports video analysis, especially for team sports such as football and soccer, has received significant attention in the academic research community. However, it is common in sports domains to use multiple cameras deployed in favorable vantage points. Work on activity recognition with camera systems that provide 3D range measurements, e.g., the KINECT sensor or stereo vision systems, as well as sensor-based activity recognition have shown some success. The annual NIST-sponsored TRECVID competition [20] also supports development of VI capabilities. *Compared to existing work, however, we target a more general-purpose VI capability suitable for a monocular camera in unconstrained environments with a richer range of agent appearances and behaviors.*

4 From Video to B-scripts

Figure 3 shows the vision pipeline that we use to convert raw video into a B-script representation. In addition to the raw video, the pipeline takes as input the known[1] camera model consisting of the camera intrinsic parameters (focal parameters, skew, and principal point) and the camera extrinsic parameters (position and orientation with respect to the world coordinate system).

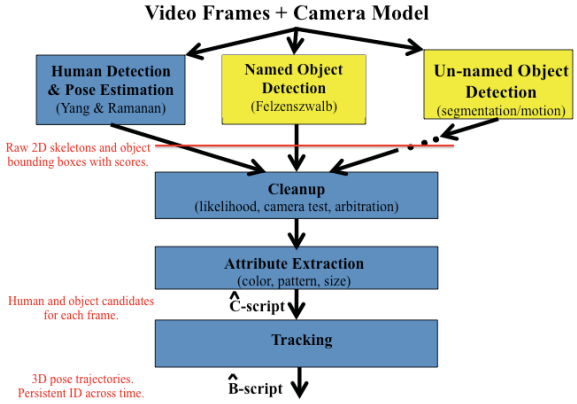

Figure 3. Vision pipeline used to recover the C-script and B-script representations from raw video frames. Note that we assume the camera model is known a priori.

Detection: There are three detection branches or pathways through the pipeline, which we refer to as the human pathway, the named object pathway, and the un-named object pathway. The human pathway is based on the human detection and 2D pose estimation algorithm of Yang and Ramanan [21]. This approach uses a training set to learn part detectors and a tree-structured model of the geometrical layout of the parts. The resulting human detector produces skeleton hypotheses, which delineate believed positions in the image plane of various body parts, along with an overall log-likelihood score that combines the part scores and geometry score. The named object pathway is based on the Deformable Parts Model (DPM) object detection algorithm of Felzenszwalb et al [22]. It is also based on learning part detectors and their geometrical layout from a training set. The output is a set of object hypotheses consisting of 2D bounding boxes and log-likelihood scores. DPM is most effective for objects that have a well-defined characteristic structure such as cars, motorcycles, bicycles, etc. It is less useful or robust for objects that are: rare (for which there may not be a pre-trained model on hand), highly variable in appearance or deformable in structure (e.g., handbags), or relatively small (few pixels) compared to the image resolution. The un-named object pathway is intended to catch these types of objects for which there is not a reliable, pre-trained detector. The un-named object pathway was not used in the experiments of Section 6.

Cleanup: Following the top-level detection blocks, there is a "cleanup" stage in which the candidate humans and objects are pruned. This stage includes thresholding the likelihood scores, applying a "camera test" to any human detections to ensure that the implied physical sizes under the known camera model are reasonable, and arbitration, which is a form of non-maximum suppression to eliminate or reduce multiple overlapping detections of people or objects. Arbitration is applied separately to humans and objects. The highest scoring detection is allowed to claim real estate in the image. Lower scoring detections are rejected if their area of support overlaps too much with an existing higher-scoring detection. This process continues until all detections meeting a minimum score threshold have been considered.

Attribute Extraction: This stage measures various properties of the surviving candidates that will be useful for composing a description and for tracking. This stage also involves situating the candidates in 3D space. For human candidates the Levenberg-Marquardt (L-M) nonlinear optimization algorithm is used to lift the 2D skeleton representation into a full 3D joint angle representation of pose. The optimization adjusts the joint angle parameters of an articulated humanoid model[2] to bring the projected positions of the model joints into agreement with the 2D image plane observations of joint positions (junctions between links in the 2D skeleton). The errors in joint position projections are insufficient to uniquely determine the joint angles. This fact is clearly shown in the work by Taylor [23], which highlights that there is a depth sign ambiguity for each

[1] For arbitrary videos downloaded from websites such as YouTube, the camera model may not be provided; we assume

[2] We use a 17-bone model for the skeleton. Internal pose consists of 36 degrees of freedom with 6 additional degrees of freedom for the overall position and orientation with respect to world frame.

link ('bone") in the humanoid model. We have added several ad hoc constraints (freezing some degrees of freedom such as trunk torsion in the humanoid model, adding penalty terms to keep one foot close to the ground plane, allowing an overall scale factor, etc.), but incorporating a stronger prior model of probable joint angle configurations as in [24] and leveraging pose information from temporally-nearby frames could improve the results. The Levenberg-Marquardt optimization is run with four initial seeds corresponding to different facing directions. The best scoring result yields an estimate of the position and orientation in 3D, as well as an overall skeleton scale factor and the internal pose parameters (joint angles). For monocular localization of the humans in 3D, we found that the L-M whole body procedure was more robust than reverse ray-tracing feet pixels to the ground plane, since foot estimates from the pose estimator tend to be unreliable.

In addition to the joint angles, colors are extracted for each body part of the human detections. The colors are found over a rectangular region centered along the corresponding "bone" in the 2D skeleton estimate.

For objects, the algorithm currently extracts a 3D position and a single average color for the object. We have yet to implement an estimator for object pose or for the physical size of objects. Even estimating the 3D position of objects is non-trivial since it requires reasoning about support relationships. Is the object supported by the ground, by a human, by another object, or is it free flying? Currently, if the bounding box of an object intersects with a dilated version of a detected human, then we assume the object is at the same depth from the camera as the human. We then reverse ray-trace the object's position in the image plane to a vertical plane in the world that is at the same depth from the camera as the human. For objects that are not close enough to be in a support relationship with a human, we ray-trace the bottom of the object's bounding box in the image plane back to the horizontal ground plane to determine its 3D coordinates.

Although this object support logic works well most of the time, there are still some problem cases. One situation is when the detection of a human carrying an object is unreliable so the human detection drops out in some frames. In these cases, the object is not deemed to be in a support relationship and is projected out into the distance onto the ground plane rather than being put closer but off the ground. Another situation occurs if there is an object on the ground in the distance beyond the person. When the two bounding boxes in the image plane get close together, the object is brought forward to the human's depth.

The output from the attribute extraction stage is a C-script, consisting of frame-by-frame detected humans and objects with color and 3D position information, and estimated pose (humans only for now). Each appearance of a human or object in a frame is given a separate label.

Tracking: A tracking algorithm is applied to upgrade C-scripts into full-fledged B-scripts in which the humans and objects maintain a persistent identity (trackID) across time. We currently use a greedy data association strategy based on a combination of distance in world coordinates, distance in pixel coordinates, and distance in color space to match detections from a new frame with previously seen agents/objects (tracks). We plan to incorporate a multiple hypothesis tracker (MHT) [25,26] in future work.

5 From B-scripts to A-scripts

Recovering an "A-script" from a video clip enables the system to not only recognize what happened, but also why it happened, what is likely to happen next, etc. An A-script consists of a sparse timeline in which parameterized behaviors (PEA graphs) are associated with agents.

5.1 Planning Executing Agent (PEA) Models

As shown earlier in Figure 2, PEAs are graphical models consisting of a set of resources, states (nodes), and transitions between states (arcs). PEAs have provided the core agent reasoning capability in several real time games and simulations. PEAs are computationally lightweight enabling simulation of millions of agents in a real-time, distributed, massively multiplayer game; the same PEA models can be used both to simulate and to recognize behaviors.

Transitions in PEA graphs are regulated by a set of predicates that indicate whether particular conditions required for transition are true. For example, if an insurgent is in the **observe** state and is notified that an IED is arriving, he immediately transitions to the **egress** state. Resource requirements may be imposed as part of a predicate, e.g., a shopper is not allowed to transition to the **buyingFromMerchant** state unless *money* > 0. Predicate values can be produced by random number generators or any user-defined procedural computer code.

Goals levied on PEAs come in two basic types: resource goals and state goals. A resource goal is an expression about one of the resources that should be achieved during the execution of the PEA, e.g., money > 100. A state goal is merely a state in the PEA graph, e.g., **playing_tag**. These basic goals can imply subgoals. For example, if we have a goal that *food* > 10, but transitioning into a food-increasing state requires that money > 10, then we have a subgoal that *money* > 10. The process of PEA compilation provides a set of lookup tables that allow us to know what our next step should be, given our current state, resources, and goals. More specifically, PEA compilation provides tables that are indexed in goal-state pairs for each PEA. These tables represent what edge a PEA should prefer or take, if given the opportunity, given the state and goal being achieved. Note that a goal could be another state or a resource request.

The PEA model in Figure 2 represents the complex verb, *give*, which involves a subject, direct object, and indirect object. The hierarchical nature of the PEA framework allows complex concepts to be built up from simpler concepts. Many

subordinate verbs may also be observed during the execution of a complex transaction.

PEA models are hand-constructed as opposed to learned from data. Since the state transitions are governed by arbitrary computer code, there is no compact way to present full details on all the PEAs used in our system (short of listing the code).

5.2 Recognizing Behaviors

To determine which PEA/goal combination best describes a series of observations, we must first map observable actions to states in each PEA. This process is via notification from the vision system that certain predicates, such as movement and proximity to other agents, have occurred. Given this mapping, we can compute the likelihood that one PEA/goal combination is more likely than another by computing the odds of each transition. Gaps in knowledge about an agent are patched by adding a goal at the start of the gap that gets us to the state at the end of the gap. *Transitioning through a PEA graph to the end state of a particular verb means that verb is believed to have occurred.*

PEAs can also be used to project future behavior by simply simulating the PEA in its environment. Multiple runs that include all agents will provide a set of execution traces. The accuracy of conclusions on the likely short-term future behavior depend highly on how probabilistic the environment is and how faithful the model is to reality. Projection has been demonstrated against human opponents in games and is reliable for determining interception paths and likely subgoals the human is trying to achieve.

6 Experiments and Results

We have applied our system to the 240 videos in the mindseye-y1-description-task available from visint.org [27]. No camera models or calibration data, e.g., checkerboard images [28,29], are available for these videos; hence, approximate camera models for each video were derived offline through various methods (horizon line, assumption of standard human height, etc.). A Matlab mex implementation of the Yang-Ramanan pose estimation algorithm was applied to every frame of every video with the output results (2D skeletons) saved in files. The part detection models were trained on a completely different video corpus. A third party [30] kindly provided object detection results from the Felzenszwalb DPM algorithm for every frame of every video. The DPM models were trained on a development corpus that was similar in style and content to the evaluation corpus.

One quirk with the object detection results, however, was that only the DPM-based detectors for objects known to occur in a particular video were applied. For example, if a video contained two people and a bicycle, only the DPM detector for "bicycle" was applied. (DPM-based human detections were not used; only Yang-Ramanan results were used for human detection.)

6.1 Envisionments and Text Descriptions

An advantage of our approach is that the scripts we derive are fully generative; they can be reformed into a graphical display (synthesized movie) called an *envisionment* or *playback.* Figure 4 (top) shows a frame of an envisionment constructed from an automatically extracted B-script.

Figure 4. (Top) A single frame in a B-script reconstruction of a video. The background is a textured canvas, while the humans and bicycle are full-fledged 3D objects. (Bottom) Original frame.

Another result is shown in Figure 5, which is a hybrid between a B-script and A-script: the detailed pose trajectories from the B-script and the text from the A-script are combined. The text annotations say: "Addison stopped", "Bailey walking", "Bailey held the dark olive green skateboard", "Bailey carried the dark olive green skateboard" and the skateboard itself is labeled with "the dark olive green skateboard". The names of the people are arbitrarily assigned (first person is assigned a name that starts with "A"; there is no gender recognition or face recognition at present). Further information readily available in the script indicate that Bailey is the person in the blue shirt. We have, in separate work, demonstrated noun-adjective queries over the video corpus such as "find bicycle and red shirt".

The PEAs currently implemented do not include a concept of "ride", so the system (somewhat incorrectly) concludes that Bailey is walking, based on his rate of travel, and that he is holding/carrying the skateboard since it travels with him. The people and the skateboard are full-fledged 3D mesh objects that are rendered according to the camera model. Only the position and color of the skateboard were derived from the image data, however; the size, orientation and CAD model are based on default values. Pose estimation for inanimate objects is intended for future work.

Figure 5. (Top) Hybrid AB-Reconstruction. (Bottom) Original Frame. The text annotations say: "Addison stopped", "Bailey walking", "Bailey held the dark olive green skateboard", "Bailey carried the dark olive green skateboard" and the skateboard itself is labeled as "the dark olive green skateboard". Bailey is the person in the blue shirt.

6.2 Quantitative Performance Metrics

While envisionments are useful for qualitative performance assessment, it is desirable to have quantitative metrics as well. Currently, we have only systematically assessed the person detection performance, which combines the Yang-Ramanan detector with the "cleanup" described in Section 4. Head positions reported in the C-scripts were projected back to image coordinates and compared to manually clicked ground truth locations. If the C-script head circle[3] encompassed the ground truth location, it was counted as a correct detection.

Table 1 shows the head-detection performance over the 113,268 frames of the corpus. Ignoring the *FAs* column for the moment, the entry in (row=*i*, column=*j*) is the *number of frames* for which there were *i* people present in the ground truth and the algorithm correctly detected *j* of them. The (0,0) entry is simply the *number of frames* in which there were no people present. The FAs column is the *number of false alarms* given that the true person count of the frame was *i*. Converting into normalized values we see that the conditional probability of correctly detecting one person when only one person is present is about 64.4%. The conditional probability of correctly detecting two people when two people are present is only 30.41%. The probability of detecting a person (without conditioning on the number of people present) is 60.4%. The average number of false positives per frame over all situations is about 0.3751. Videos where the person detection rate is below 20% typically have the person in an elevated position relative to the ground (e.g., on a ladder, atop stairs, on a fence) or sitting on the ground. These comprise about 12% of the videos in the corpus.

[3] Because the manual clicking was only done every 10th frame with interpolation between, we expanded the head radius by a factor of 1.5 for scoring.

	0	1	2	3	4	FAs
0	19105	-	-	-	-	4900
1	26507	47993	-	-	-	28876
2	4537	8335	5625	-	-	8159
3	52	381	694	8	-	498
4	3	11	8	8	1	55

Table 1. Human (head) detection performance from C-scripts evaluated over 113,268 frames of the corpus. The entry in row *i* and column j is the *number of frames* in which j persons were correctly detected given that i persons were present. The *FAs* column is simply the total number of false alarms (not the number of frames) that occurred in situations where *i* people were present in the frame. For example, from the row with *i*=4, one can conclude that only 31 frames had 4 people present and in only one of those frames did the algorithm correctly detect and localize all 4 people. In those 31 frames, a total of 55 false alarms occurred.

7 Conclusion

We have developed a visual intelligence system for automatically processing HD video into a highly compressed[4] script that identifies the actors, objects, and actions in the original video. The script is fully generative and can be used to produce graphical renderings ("envisionments") of the action and/or natural language text descriptions. The system combines state of the art vision components for human pose estimation and object detection with extremely powerful backend graphical models that allow video content to be represented and reasoned about at a symbolic level.

The performance of the overall system is currently limited by the human detection and pose estimation step. We have avoided the use of background subtraction so that the method can be applied from a moving platform or in situations where there is significant background motion. However, the state-of-the-art in person detection from static frames does not appear to be sufficiently robust to achieve desired performance levels. To be sure, in some cases, the person detection and pose estimation work well yielding convincing envisionments and text descriptions that match the action in the scene. But in many cases, the human detection is not reliable enough, particularly for non-standing poses. Making human detection and pose estimation robust, is a much-needed improvement.

Another interesting direction is to automatically determine which adjectives most clearly separate and uniquely identify the individuals involved in an action. Is it more informative to say "the person in the blue jeans" or "the taller person"? From descriptions provided by human annotators, it appears that gender designation ("the man approached the woman") is very common. Adding a gender classification component could provide a more human-like character to the descriptions.

[4] Five orders of magnitude more compact than the original raw video.

8 Acknowledgments

This research was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration. The authors thank James Donlon and Pietro Michelucci for organizing the Mind's Eye program and providing support, Deva Ramanan of UCI for making his pose estimation code available, Jeff Siskind of Purdue University for sharing the Felzenszwalb object detection results he generated over this dataset, and Allan Runkle of JPL for developing the *DisplayServer* tool used to show the envisionments.

9 References

[1] Ferrucci, D., Brown, E., Chu-Caroll, J., Fan, J., Gondek, D., Kalyanpur, A.A., Lally, A., Murdock, J.W., Nyberg, E., Prager, J., Schlaefer, N., Welty, C., "Building Watson: An Overview of the DeepQA Project", *Assoc. for Advancement of AI*, pp. 59-79, (2010).

[2] Wolfram, Stephen, "WolframAlpha: Computational Knowledge Engine", URL: http://www.wolfram.com/

[3] *Siri*, URL: http://www.apple.com/ios/siri/

[4] Barnard, K. and Forsyth, D., "Learning the Semantics of Words and Pictures", *Int. Conf. on Computer Vision (ICCV)*, vol. 2, pp. 408-415, (2001).

[5] Ackerman, S., "Beyond Surveillance: DARPA wants a thinking camera", *Wired*, (2011/01/05).

[6] Aggarwal, J.K., Ryoo, M.S., "Human Activity Analysis: A Review", *ACM Computing Surveys*, 43(3), (2011/04/XX).

[7] Aggarwal, J.K., Ryoo, M.S., Kitani, K. "Tutorial on Human Activity Recognition – Frontiers of Human Activity Analysis", *IEEE Conf. on Computer Vision and Pattern Recognition, (CVPR)*, (2011).

[8] Hongeng, S., R. Nevatia, F. Bremond, "Video-based event recognition: activity representation and probabilistic recognition methods", *Computer Vision and Image Understanding (CVIU)*, 96, pp. 129-162, (2004).

[9] Van den Broek, S., ten Hove, J.-M., den Hollander, R., Burghouts, G., "Automated recognition of human activities in video streams in real time", *SPIE Newsroom*, (2014).

[10] Das, P., Xu, C., Doell, R.F., and Corso, J.J, "A thousand frames in just a few words: Lingual description of videos through latent topics and sparse object stitching." *CVPR*, (2013).

[11] S. O'Hara and B. Draper, "Using a Product Manifold Distance for Unsupervised Action Recognition", Image and Vision Computing, 30(3):206-216, (2012).

[12] Siddharth, N., Barbu, A., and Siskind, J.M., "Seeing What You're Told: Sentence-Guided Activity Recognition In Video", *CVPR*, pp. 732-739, Columbus, OH, (2014/06/XX).

[13] Guadarrama, S., Krishnmoorthy, N., Malkarnenkar, G., Venugopalan, S., Mooney, R., Darrell, T., Saenko, K., "YouTube2Text: Recognizing and Describing Arbitrary Activities Using Semantic Hierarchies and Zero-Shot Recognition", *ICCV*, (2013).

[14] Allen, J.F., "Maintaining Knowledge about Temporal Intervals", *Comm. of the ACM*, pp. 3-843, (1983/11/26).

[15] Brand, M., Oliver, N., Pentland, A., "Coupled Hidden Markov Models for Complex Action Recogntion", *CVPR*, pp. 994-999, (1997).

[16] Oliver, N., Horvitz, E., Ashutosh, G., "Layered Representations for Human Activity Recognition", *Fourth IEEE Int. Conf. on Multimodal Interfaces*, (2002).

[17] Rabiner, L.R., "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition", *Proc. of IEEE*, vol. 77, no. 2, (1989/02/XX).

[18] IP Video Market, URL: http://ipvideomarket.info/comapnies/videoanalytics/

[19] Brickstream, Inc., URL: http://www.brickstream.com/

[20] Smeaton, A.F., Over, P. , Kraaij, W., "Evaluation Campaigns and TRECVid", *Eigth ACM Int. Workshop on Multimedia Information Retrieval*, Santa Barbara, CA, pp. 321—330, (2006).

[21] Yang, Y., and Ramanan, D., "Articulated pose estimation with flexible mixtures-of-parts", *CVPR, (2011)*.

[22] Felzenszwalb, P., Girshick, D. McAllester, D. Ramanan, "Object Detection with Discriminatively Trained Part Based Models", *IEEE Trans. on PAMI (TPAMI)*, vol. 32, no. 9, (2010/09/XX).

[23] Taylor, C., "Reconstruction of articulated objects from point correspondences in a single uncalibrated image", *CVIU*, 81(3):269-284, (2001/03/XX).

[24] Ramakrishna, V., Kanade, T., Sheikh, Y., "Reconstructing 3D Human Pose from 2D Image Landmarks", *Euro. Conf. on Comp. Vision (ECCV)*, pp. 573-586, (2012).

[25] Cox, I.J., and Hingorani, S.L., "An Efficient Implementation of Reid's Multiple Hypothesis Tracking Algorithm for the Purpose of Visual Tracking", *TPAMI*, vol. 18, no. 2, pp. 138-150, (1996/02/XX).

[26] Blackman, S.S., "Multiple Hypothesis Tracking for Multiple Target Tracking", *IEEE Trans. on Aerospace and Electronics Systems (AES)*, vol. 19, no. 1, pp. 5-18, (2004/01/XX).

[27] Visint.org, "Resources for visual intelligence research", URL: http://www.visint.org/index.html

[28] Bouguet J.-Y., "Camera Calibration Toolbox for Matlab", URL: http://www.vision.caltech.edu/bouguetj/calib_doc/

[29] Zhang, Z., "Flexible Camera Calibration by Viewing a Plane from Unknown Orientation", *ICCV*, (1999).

[30] Siskind, J., Purdue University, private communication via e-mail. (2011/07/19)