SYSTEM AND METHOD FOR SENTENCE DIRECTED VIDEO OBJECT CODETECTION

Codetection

FIG. 2


(51) International Patent Classification:
G06F 17/30 (2006.01) G06K 9/66 (2006.01)
G06K 9/62 (2006.01) G06K 9/72 (2006.01)

(21) International Application Number:
PCT/US2017/036232

(22) International Filing Date:
06 June 2017 (06.06.2017)

(25) Filing Language:
English

(26) Publication Language:
English

(30) Priority Data:
62/346,459 06 June 2016 (06.06.2016) US

(71) Applicant: PURDUE RESEARCH FOUNDATION [US/US]; 1801 Newman Road, West Lafayette, IN 47906 (US).

(72) Inventors: SISKIND, Jeffrey; Mark: 3429 Crawford Street, West Lafayette, IN 47906 (US). YU, Haoman; 1029 Blazingwood Drive, Sunnycovel, CA 94089 (US).

(74) Agent: JALAIE, Bobak, P.; Purdue Research Foundation, 1801 Newman Road, West Lafayette, IN 47906 (US).

(81) Designated States (unless otherwise indicated, for every kind of national protection available): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DJ, DK, DM, DO,
SYSTEM AND METHOD FOR SENTENCE DIRECTED VIDEO OBJECT CODETECTION

CROSS REFERENCE TO RELATED APPLICATIONS
[0001] The present application is related to and claims the priority benefit of U.S. Provisional Patent Application Serial No. 62/346,459, filed June 6, 2016, the contents of which are hereby incorporated by reference in their entirety into the present disclosure.

STATEMENT REGARDING GOVERNMENT FUNDING
[0002] This invention was made with government support under W91 INF- 10-2-0060 awarded by the Army Research Laboratory and under 1522954-IIS awarded by the National Science Foundation. The government has certain rights in the invention.

TECHNICAL FIELD
[0003] The present application relates to video detection systems, and more specifically, to a system for determining the locations and types of objects in a video content

BACKGROUND
[0004] Prior art video codetection systems work by selecting one out of many object proposals per image or frame that maximizes a combination of the confidence scores associated with the selected proposals and the similarity scores between proposal pairs. However, such systems typically require human pose and depth information in order to prune the search space and reduce computer processing time and increase accuracy. Further, codetection methods, whether for images or video, codetect only one common object at a time: different object classes are codetected independently. Therefore, improvements are needed in the field.
SUMMARY

[0005] According to one aspect, a method for determining the locations and types of objects in a plurality of videos is provided, comprising pairing each video with one or more sentences describing the activity or activities in which those objects participate in the associated video, wherein no use is made of a pretrained object detector. The object locations are specified as rectangles, the object types are specified as nouns, and sentences describe the relative positions and motions of the objects in the videos referred to by the nouns in the sentences. The relative positions and motions of the objects in the video are described by a conjunction of predicates constructed to represent the activity described by the sentences associated with the videos. According to certain aspects, the locations and types of the objects in the collection of videos are determined by using one or more object proposal mechanisms to propose locations for possible objects in one or more frames of the videos. In various aspects, the set of proposals is augmented by detections produced by a pretrained object detector.
BRIEF DESCRIPTION OF THE DRAWINGS

[0006] In the following description and drawings, identical reference numerals have been used, where possible, to designate identical features that are common to the drawings.

[0007] FIG. 1 is a diagram showing input video frames according to various aspects.

[0008] FIG. 2 is a diagram illustrating an object codetection process according to various aspects.

[0009] FIG. 3 is a diagram showing output of the codetection process of FIG. 2 according to various aspects.

[0010] FIG. 4 is a diagram showing a system for performing the method of FIGs. 1-3 according to various aspects.

[0011] The attached drawings are for purposes of illustration and are not necessarily to scale.
DETAILED DESCRIPTION

[0012] In the following description, some aspects will be described in terms that would ordinarily be implemented as software programs. Those skilled in the art will readily recognize that the equivalent of such software can also be constructed in hardware, firmware, or micro-code. Because data-manipulation algorithms and systems are well known, the present description will be directed in particular to algorithms and systems forming part of, or cooperating more directly with, systems and methods described herein. Other aspects of such algorithms and systems, and hardware or software for producing and otherwise processing the signals involved therewith, not specifically shown or described herein, are selected from such systems, algorithms, components, and elements known in the art. Given the systems and methods as described herein, software not specifically shown, suggested, or described herein that is useful for implementation of any aspect is conventional and within the ordinary skill in such arts.

[0013] In the system and method of the present disclosure, input video images are processed to achieve object codiscovery, defined herein as naming and localizing novel objects in a set of videos, by placing bounding boxes (rectangles) around those objects, without any pretrained object detectors. Therefore, given a set of videos that contain instances of a common object class, the system locates those instances simultaneously. The method of the present disclosure differs from most prior codetection methods in two crucial ways. First, the presently disclosed method can codetect small or medium sized objects, as well as ones that are occluded for part of the video. Second, it can codetect multiple object instances of different classes both within a single video clip and across a set of video clips.

[0014] The presently disclosed method extracts spatio-temporal constraints from sentences that describe the videos and then impose these constraints on the codiscovery process to find the collections of objects that best satisfy these constraints and that are similar within each object class. Even though the constraints implied by a single sentence are usually weak, when accumulated across a set of videos and sentences, they together
will greatly prune the detection search space. This process is referred to herein as sentence directed video object codiscovery. The process produces instances of multiple object classes at a time by its very nature. The sentence we use to describe a video usually contains multiple nouns referring to multiple object instances of different classes. The sentence semantics captures the spatiotemporal relationships between these objects. As a result, the codiscovery of one object class affects that of the others and vice versa. In contrast, prior art codetection methods, whether for images or video, codetect only one common object class at a time: different object classes are codetected independently. Each time they output a single object detection of the same class for each video clip.

[0015] In general, the presently disclosed method extracts a set of predicates from each sentence and formulate each predicate around a set of primitive functions. The predicates may be verbs (e.g., CARRIED and ROTATED), spatial-relation prepositions (e.g., LEFTOF and ABOVE), motion prepositions (e.g., AWAYFROM and TOWARDS), or adverbs (e.g., QUICKLY and SLOWLY). The sentential predicates are applied to the candidate object proposals as arguments, allowing an overall predicate score to be computed that indicates how well these candidate object proposals satisfy the sentence semantics. The predicate score is added into the codiscovery framework, on top of the original similarity score, to guide the optimization.

[0016] FIGs. 1-3 illustrate a process for sentence directed video object codiscovery according to one embodiment. As shown in FIG. 1 ("Input"), a set of videos, which is previously paired with human-elicited sentences, one sentence per video is received as input. For each sentence, a conjunction of predicates is extracted together with the object instances as the predicate arguments. In the example illustrated in FIG. 1, we have:
The sentences in this example contain six nouns. Thus we extract six object instances: cabbageO, cabbageI, squashO, bowlO, bowlI, and mouthwashO, and produce six tracks, one track per object instance. Two tracks will be produced for each of the three video clips. To accomplish this, a collection of object-candidate generators and video-tracking methods are applied to each video to obtain a pool of object proposals. Any proposal in a video's pool is a possible object instance to assign to a noun in the sentence associated with that video. Given multiple such video-sentence pairs, a graph is formed where object instances serve as vertices and there are two kinds of edges: similarities between object instances and predicates linking object instances in a sentence. Belief Propagation is applied to this graph to jointly infer object codiscoveries by determining an assignment of proposals to each object instance. In the output as shown in FIG. 3, the red track of the first video clip is selected for cabbageO, and the blue track is selected for bowlO. The green track of the second video clip is selected for squashO, and the blue track is selected for bowlI. The red track of the third video clip is selected for cabbageI, and the yellow track is selected for mouthwashO. All six tracks are produced simultaneously in one inference run. Below, we explain the details of each component of this codiscovery framework.

The presently disclosed method exploits sentence semantics to help the codiscovery process. A conjunction of predicates is used to represent (a portion of) the semantics of a sentence. Object instances in a sentence fill the arguments of the predicates in that sentence. An object instance that fills the arguments of multiple predicates is said to be coreferenced. For a coreferenced object instance, only one track is
For example, a sentence like "The person is placing the mouthwash next to the cabbage in the sink" implies the following conjunction of predicates:

\[ \text{OOWN}(\text{mouthwash}) \land \text{NEAR}(\text{mouthwash}, \text{cabbage}) \]

[0019] In this case, mouthwash is coreferenced by the predicates DOWN (fills the sole argument) and NEAR (fills the first argument). Thus only one mouthwash track will be produced, simultaneously constrained by the two predicates (FIG. 3, yellow track). This coreference mechanism plays a crucial role in the codiscovery process. It tells us that there is exactly one mouthwash instance in the above sentence: the mouthwash that is being placed down is identical to the one that is placed near the cabbage. In the absence of such a coreference constraint, the only constraint between these two potentially different instances of the object class mouthwash would be that they are visually similar. Stated informally in English, this would be:

"The cabbage is near a mouthwash that is similar to another mouthwash which is placed down."

Not only does this impose an unnecessarily weaker constraint between cabbage and mouthwash, it also fails to correctly reflect the sentence semantics. To overcome this limitation, the presently disclosed method for extracting predicates from a sentence consists of two steps: parsing and transformation/distillation. The method first uses the Stanford parser (Socher et al 2013) to parse the sentence. Next, the method employs a set of rules to transform the parsed results to ones that are 1) pertinent to visual analysis, 2) related to a prespecified set of object classes, and 3) distilled so that synonyms are mapped to a common word. These rules simply encode the syntactic variability of how objects fill arguments of predicates. They do not encode semantic information that is particular to specific video clips or datasets. For example, in the sentence "A young man put down the cup", the adjective young is not relevant to our purpose of object codiscovery and will be removed. In the sentence "The person is placing the mouthwash in the sink", the object sink is not one of the prespecified object classes. In this case, we
simply ignore the extraneous objects that are out of scope. Thus for the phrase "placing the mouthwash in the sink" in the above sentence, we only extract the predicate \texttt{OOWN(mouthwash)}. Finally, synonyms introduced by different annotators, \textit{e.g., person, man, woman, child, and adult}, are all mapped to a common word \textit{(person)}. This mapping process also applies to other parts of speech, including verbs, prepositions, and adverbs. This transformation/distillation process never yields \textit{stronger} constraint and usually yields \textit{weaker} constraint than that implied by the semantics of the original sentences.

[0020] While the presently disclosed method employs a set of manually designed rules, the whole transformation/distillation process is automatic performed by the processor, which allows the system to handle sentences of similar structure with the same rule(s). To eliminate the manually designed rules, one could train a semantic parser. However, modern semantic parsers are domain specific, and no existing semantic parser has been trained on our domain. Training a new semantic parser usually requires a parallel corpus of sentences paired with intended semantic representations. Semantic parsers are trained with corpora like PropBank (Palmer et al 2005) that have tens of thousands of manually annotated sentences. Gathering such a large training corpus would be overkill for our experiments that involve only a few hundred sentences, especially since such is not our focus or contribution. Thus we employ simpler handwritten rules to automate the semantic parsing process for our corpora in this paper. Nothing, in principle, precludes using a machine-trained semantic parser in its place. However, we leave that to future work.

[0021] The predicates used to represent sentence semantics are formulated around a set of primitive functions on the arguments of the predicate. These produce scores indicating how well the arguments satisfy the constraint intended by the predicate. Table 1 defines 36 predicates used to represent sentence semantics in certain examples. Table 2 defines 12 example primitive functions used to formulate these predicates. In Table 1, the symbol \( p \) denotes an object proposal, \( p^{(t)} \) denotes frame \( t \) of an object proposal, and \( p^{(L)} \) and \( p^{<L>} \) denote averaging the score of a primitive function over the first and last \( L \) frames.
of a proposal respectively. When there is no time superscript on $p$, the score is averaged over all frames (e.g., BEHIND).
While the predicates of the presently disclosed system and method are manually designed, they are straightforward to design and code. The effort to do so (several hundred lines of code) could be even less than that of designing a machine learning model that handles the three datasets in our experiments. The reason why this is the case is that the predicates encode only weak constraints. Each predicate uses at most four primitive functions (most use only two). The primitive functions are simple, e.g., the temporal coherence (tempCoher) of an object proposal, the average flow magnitude (medFlMg) of a proposal, or simple spatial relations like distLessThan/distGreaterThan between proposals. Unlike features used to support activity recognition or video captioning, these primitive functions need not accurately reflect every nuance of motion and changing spatial relations between objects in the video that is implied by the sentence semantics. They need only reflect a weak but sufficient level of the sentence semantics to
help guide the search for a reasonable assignment of proposals to nouns during
codiscovery. Because of this important property, these primitive functions are not as
highly engineered as they might appear to be. The predicates of the presently disclosed
method are general in nature and not specific to specific video samples or datasets.

[0023] To generate object proposals, the system first generates \( N \) object candidates for
each video frame and construct proposals from these candidates. To support codiscovery
of multiple stationary and moving objects, some of which might not be salient and some
of which might be occluded for part of the video, the presently disclosed method for
generating object candidates must be general purpose: it cannot make assumptions about
the video (e.g., simple background) or exhibit bias towards a specific category of objects
(e.g., moving objects). Thus methods that depend on object salience or motion analysis
would not be suitable with the presently disclosed method. The presently disclosed
method uses EdgeBoxes (Zitnick and Dollar 2014) to obtain the \( N/2 \) top-ranking object
candidates and MCG (Arbelaez et al. 2014) to obtain the other half, filtering out
candidates larger than \( 1/20 \) of the videoframe size to focus on small and medium-sized
objects. This yields \( NT \) object candidates for a video with \( T \) frames. The system then
generate \( K \) object proposals from these \( NT \) candidates. To obtain object proposals with
object candidates of consistent appearance and spatial location, one would nominally
require that \( K \ll N \). To circumvent this, the system first randomly samples a frame \( t \)
from the video with probability proportional to the average magnitude of optical flow
(Farneback 2003) within that frame. Then, the system samples an object candidate from
the \( N \) candidates in frame \( t \). To decide whether the object is moving or not, the system
samples from \{MOVING, STATIONARY\} with distribution \{1/3, 2/3\}. The system
samples a MOVING object candidate with probability proportional to the average flow
magnitude within the candidate. Similarly, the system samples a STATIONARY object
candidate with probability inversely proportional to the average flow magnitude within
the candidate. The sampled candidate is then propagated (tracked) bidirectionally to the
start and the end of the video. We use the CamShift algorithm (Bradski 1998) to track
both MOVING and STATIONARY objects, allowing the size of MOVING objects to
change during the process, but requiring the size of STATIONARY objects to remain constant. STATIONARY objects are tracked to account for noise or occlusion that manifests as small motion or change in size. The system tracks STATIONARY objects in RGB color space and MOVING objects in HSV color space. Generally, RGB space is preferable to HSV space because HSV space is noisy for objects with low saturation (e.g., white, gray, or dark) where the hue ceases to differentiate. However, HSV space is used for MOVING objects as it is more robust to motion blur. RGB space is used for STATIONARY objects because motion blur does not arise. The system preferably does not use optical-flow-based tracking methods since these methods suffer from drift when objects move quickly. The system repeats this sampling and propagation process $K$ times to obtain $K$ object proposals $\{p_k\}$ for each video. Examples of the sampled proposals ($K = 240$) are shown as black boxes (rectangles) 110 in FIG. 2.

[0024] To compute the similarity of two object proposals, the system implements the method as follows. The system first uniformly sample $M$ boxes (rectangles) $\{b^m\}$ from each proposal $p$ along its temporal extent. For each sampled box (rectangle), the system extracts PHOW (Bosch et al 2007) and HOG (Dalai and Triggs 2005) features to represent its appearance and shape. The system also does so after rotating this detection by 90 degrees, 180 degrees, and 270 degrees. Then, the system measures the similarity $g$ between a pair of detections $b^m_1$ and $b^m_2$ with:

$$g(b^m_1, b^m_2) = \max_{i,j \in \{0,1,2,3\}} \frac{1}{2} \left( g_{x^2}(\text{rot}_i(b^m_1), \text{rot}_j(b^m_2)) + g_{L^2}(\text{rot}_i(b^m_1), \text{rot}_j(b^m_2)) \right)$$

where $\text{rot}_i$ $i = 0,1,2,3$ represents rotation by 0 degrees 90 degrees 180 degrees and 270 degrees respectively. The system uses $g_{x^2}$ to compute the $x^2$ distance between the PHOW features and $g_{L^2}$ to compute the Euclidean distance between the HOG features, after which the distances are linearly scaled to [0,1] and converted to log similarity scores.

Finally, the similarity between two proposals $p_1$ and $p_2$ is taken to be:
\[ g(p_1, p_2) = \text{median}_m h(v_{p_1}, v_{p_2}) \]

[0025] The system extracts object instances from the sentences and model them as vertices in a graph. Each vertex \( v \) can be assigned one of the \( K \) proposals in the video that is paired with the sentence in which the vertex occurs. The score of assigning a proposal \( k_v \) to a vertex \( v \) is taken to be the unary predicate score \( h_v(k_v) \) computed from the sentence (if such exists, or otherwise 0). The system constructs an edge between every two vertices \( u \) and \( v \) that belong to the same object class. This class membership relation is denoted as \( (u, v) \in \mathcal{E} \). The score of this edge \((w,v)\), when the proposal \( k_u \) is assigned to vertex \( u \) and the proposal \( k_v \) is assigned to vertex \( v \), is taken to be the similarity score \( g_wv(k_u, k_v) \) between the two proposals. Similarly, the system also constructs an edge between two vertices \( u \) and \( v \) that are arguments of the same binary predicate. This predicate membership relation is denoted as \( (u, v) \in \mathcal{P} \). The score of this edge \((u,v)\), when the proposal \( k_u \) is assigned to vertex \( u \) and the proposal \( k_v \) is assigned to vertex \( v \), is taken to be the binary predicate score \( h_{uv}(k_u, k_v) \) between the two proposals. The problem, then, is to select a proposal for each vertex that maximizes the joint score on this graph, i.e., solving the following optimization problem for a CRF:

\[
\max_k \sum_v h_v(k_v) + \sum_{(u,v) \in \mathcal{E}} g_{u,v}(k_u, k_v) + \sum_{(u,v) \in \mathcal{P}} h_{uv}(k_u, k_v)
\]

where \( k \) is the collection of the selected proposals for all the vertices. This discrete inference problem can be solved approximately by Belief Propagation (Pearl 1982).

[0026] Conceptually, this joint inference does not require sentences for every video clip. In such a case where some video clips are not described with sentences, the system output would only have the similarity score \( g \) in Eq. 1 for these clips, and would have both the similarity and predicate scores for the rest. This flexibility allows the presently disclosed method to work with videos that do not exhibit apparent semantics or exhibit semantics that can only be captured by extremely complicated predicates or models. Furthermore, the semantic factors \( h \) may cooperate with other forms of constraint or knowledge, such
as the pose information, by having additional factors in the CRF to encode such constraint or knowledge. This would further boost the performance of object codiscovery implemented by the disclosed system.

[0027] FIG. 4 is a high-level diagram showing the components of an exemplary data-processing system for analyzing data and performing other analyses described herein, and related components. The system includes a processor 186, a peripheral system 120, a user interface system 130, and a data storage system 140. The peripheral system 120, the user interface system 130 and the data storage system 140 are communicatively connected to the processor 186. Processor 186 can be communicatively connected to network 150 (shown in phantom), e.g., the Internet or a leased line, as discussed below. It shall be understood that the system 120 may include multiple processors 186 and other components shown in FIG. 4. The video content data, and other input and output data described herein may be obtained using network 150 (from one or more data sources), peripheral system 120 and/or displayed using display units (included in user interface system 130) which can each include one or more of systems 186, 120, 130, 140, and can each connect to one or more network(s) 150. Processor 186, and other processing devices described herein, can each include one or more microprocessors, microcontrollers, field-programmable gate arrays (FPGAs), application-specific integrated circuits (ASICs), programmable logic devices (PLDs), programmable logic arrays (PLAs), programmable array logic devices (PALs), or digital signal processors (DSPs).

[0028] Processor 186 can implement processes of various aspects described herein. Processor 186 can be or include one or more device(s) for automatically operating on data, e.g., a central processing unit (CPU), microcontroller (MCU), desktop computer, laptop computer, mainframe computer, personal digital assistant, digital camera, cellular phone, smartphone, or any other device for processing data, managing data, or handling data, whether implemented with electrical, magnetic, optical, biological components, or
otherwise. Processor 186 can include Harvard-architecture components, modified-Harvard-architecture components, or Von-Neumann-architecture components.

[0029] The phrase "communicatively connected" includes any type of connection, wired or wireless, for communicating data between devices or processors. These devices or processors can be located in physical proximity or not. For example, subsystems such as peripheral system 120, user interface system 130, and data storage system 140 are shown separately from the data processing system 186 but can be stored completely or partially within the data processing system 186.

[0030] The peripheral system 120 can include one or more devices configured to provide information to the processor 186. For example, the peripheral system 120 can include electronic or biological sensing equipment, such as magnetic resonance imaging (MRI) scanners, computer tomography (CT) scanners, and the like. The processor 186, upon receipt of information from a device in the peripheral system 120, can store such information in the data storage system 140.

[0031] The user interface system 130 can include a mouse, a keyboard, another computer (connected, e.g., via a network or a null-modem cable), or any device or combination of devices from which data is input to the processor 186. The user interface system 130 also can include a display device, a processor-accessible memory, or any device or combination of devices to which data is output by the processor 186. The user interface system 130 and the data storage system 140 can share a processor-accessible memory.

[0032] In various aspects, processor 186 includes or is connected to communication interface 115 that is coupled via network link 116 (shown in phantom) to network 150. For example, communication interface 115 can include an integrated services digital network (ISDN) terminal adapter or a modem to communicate data via a telephone line; a network interface to communicate data via a local-area network (LAN), e.g., an Ethernet LAN, or wide-area network (WAN); or a radio to communicate data via a wireless link, e.g., WiFi or GSM. Communication interface 115 sends and receives electrical,
electromagnetic or optical signals that carry digital or analog data streams representing various types of information across network link 116 to network 150. Network link 116 can be connected to network 150 via a switch, gateway, hub, router, or other networking device.

[0033] Processor 186 can send messages and receive data, including program code, through network 150, network link 116 and communication interface 115. For example, a server can store requested code for an application program (e.g., a JAVA applet) on a tangible non-volatile computer-readable storage medium to which it is connected. The server can retrieve the code from the medium and transmit it through network 150 to communication interface 115. The received code can be executed by processor 186 as it is received, or stored in data storage system 140 for later execution.

[0034] Data storage system 140 can include or be communicatively connected with one or more processor-accessible memories configured to store information. The memories can be, e.g., within a chassis or as parts of a distributed system. The phrase "processor-accessible memory" is intended to include any data storage device to or from which processor 186 can transfer data (using appropriate components of peripheral system 120), whether volatile or nonvolatile; removable or fixed; electronic, magnetic, optical, chemical, mechanical, or otherwise. Exemplary processor-accessible memories include but are not limited to: registers, floppy disks, hard disks, tapes, bar codes, Compact Discs, DVDs, read-only memories (ROM), erasable programmable read-only memories (EPROM, EEPROM, or Flash), and random-access memories (RAMs). One of the processor-accessible memories in the data storage system 140 can be a tangible non-transitory computer-readable storage medium, i.e., a non-transitory device or article of manufacture that participates in storing instructions that can be provided to processor 186 for execution.

[0035] In an example, data storage system 140 includes code memory 141, e.g., a RAM, and disk 143, e.g., a tangible computer-readable rotational storage device such as a hard drive. Computer program instructions are read into code memory 141 from disk 143.
Processor 186 then executes one or more sequences of the computer program instructions loaded into code memory 141, as a result performing process steps described herein. In this way, processor 186 carries out a computer implemented process. For example, steps of methods described herein, blocks of the flowchart illustrations or block diagrams herein, and combinations of those, can be implemented by computer program instructions. Code memory 141 can also store data, or can store only code.

[0036] Various aspects described herein may be embodied as systems or methods. Accordingly, various aspects herein may take the form of an entirely hardware aspect, an entirely software aspect (including firmware, resident software, micro-code, etc.), or an aspect combining software and hardware aspects. These aspects can all generally be referred to herein as a "service," "circuit," "circuitry," "module," or "system."

[0037] Furthermore, various aspects herein may be embodied as computer program products including computer readable program code stored on a tangible non-transitory computer readable medium. Such a medium can be manufactured as is conventional for such articles, e.g., by pressing a CD-ROM. The program code includes computer program instructions that can be loaded into processor 186 (and possibly also other processors), to cause functions, acts, or operational steps of various aspects herein to be performed by the processor 186 (or other processor). Computer program code for carrying out operations for various aspects described herein may be written in any combination of one or more programming language(s), and can be loaded from disk 143 into code memory 141 for execution. The program code may execute, e.g., entirely on processor 186, partly on processor 186 and partly on a remote computer connected to network 150, or entirely on the remote computer.

[0038] The invention is inclusive of combinations of the aspects described herein. References to "a particular aspect" and the like refer to features that are present in at least one aspect of the invention. Separate references to "an aspect" (or "embodiment") or "particular aspects" or the like do not necessarily refer to the same aspect or aspects; however, such aspects are not mutually exclusive, unless so indicated or as are readily
apparent to one of skill in the art. The use of singular or plural in referring to "method" or "methods" and the like is not limiting. The word "or" is used in this disclosure in a non-exclusive sense, unless otherwise explicitly noted.

[0039] The invention has been described in detail with particular reference to certain preferred aspects thereof, but it will be understood that variations, combinations, and modifications can be effected by a person of ordinary skill in the art within the spirit and scope of the invention.
CLAIMS:

1. A method for determining the locations and types of objects in a plurality of videos, comprising:
   using a computer processor, receiving a plurality of videos;
   pairing each of the videos with one or more sentences;
   using the processor, describing one or more activities in which those objects participate in a corresponding video; and
   wherein no use is made of a pretrained object detector.

2. The method of claim 1, wherein locations of the objects are specified by the processor as rectangles in frames of the videos, the object types are specified as nouns, and sentences describe the relative positions and motions of the objects in the videos referred to by the nouns in the sentences.

3. The method of claims lor 2, wherein the relative positions and motions of the objects in the video are described by a conjunction of predicates constructed to represent the activity described by the sentences associated with the videos.

4. The method according to any previous claim, wherein the locations and types of the objects in the plurality of videos are determined by:
   a. using one or more object proposal mechanisms to propose locations for possible objects in one or more frames of the videos;
   b. using one or more object trackers to track the positions of the proposed object locations forward or backward in time;
   c. collecting the tracked proposal positions for each proposal into a tube;
   d. computing features for each tube based on image features for the portion of the images inside the tubes; or
   e. forming a graphical model, wherein:
      i. one or more noun occurrences in sentences associated with a video are associated with vertices in the model;
      ii. the set of potential labels of each vertex is the set of proposal tubes for the associated video;
      iii. pairs of vertices that are associated with occurrences of the same noun in two sentences associated with different videos are attached by a binary
factor computed as a similarity measure between the tubes selected for from the label sets for the two vertices;

iv. collections of vertices that are associated with occurrences of different nouns in the same sentence associated with a video are attached by a factor whose arity is the arity of a predicate in the conjunction of the predicates used to represent the activity described by the sentence where the score of said represents the degree to which the collection of tubes selected for those vertices exhibits the properties of that predicate; or

v. the graphical model is solved by selecting a single proposal tube for each vertex from the set of potential labels for that vertex that collectively maximizes a combination of the similarity measure for all pairs of vertices connected by a similarity factor and the predicate scores of all collections of vertices connected by a predicate factor.

5. The method according to any previous claim, wherein the proposal generation mechanism is MCG.

6. The method according to any previous claim, wherein the proposal generation mechanism is EdgeBoxes.

7. The method according to any previous claim, wherein the proposals are tracked by CamShift.

8. The method according to any previous claim, wherein moving proposals are tracked in HSV color space and allowed to change size.

9. The method according to any previous claim, wherein stationary proposals are tracked in RGB color space and are required to remain of constant size.

10. The method according to any previous claim, wherein PHOW features are used as image/tube features.

11. The method according to any previous claim, wherein HOG features are used as image/tube features.
12. The method according to any previous claim, wherein similarity is measured using a chi-squared distance between image/tube features.

13. The method according to any previous claim, wherein similarity is measured using Euclidean distance between image/tube features.

14. The method according to any previous claim, wherein the set of proposal is augmented with proposals rotated by multiples of 90 degrees.

15. The method according to any previous claim, wherein the similarity measures and predicate scores are combined by summation.

16. The method according to any previous claim, wherein the similarity measures and predicate scores are combined by taking their product.

17. The method according to any previous claim, wherein the graphical model is solved using Belief Propagation.

18. The method according to any previous claim, wherein the set of proposals is augmented by detections produced by a pretrained object detector.

19. The method of claim 18, wherein the method of claim 1 is first applied and then the method of claim 18 is applied in one or more subsequent iterations, each iteration using an object detector trained on the proposals selected in earlier iterations.
Input

The man removed the violet cabbage from the bowl.

The person carried the squash to the left, away from the yellow bowl.

The person is placing the mouthwash next to the cabbage in the sink.

FIG. 1
Codetection

FIG. 2
Output

FIG. 3
FIG. 4
INTERNATIONAL SEARCH REPORT

International application No. PCT/US2017/036232

A. CLASSIFICATION OF SUBJECT MATTER
IPC(8) - G06F 17/30; G06K 9/62; G06K 9/66; G06K 9/72 (2017.01)
CPC - G06F 17/3079; G06K 9/00751; G06K 9/00771; G06K 9/723 (2017.02)

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
See Search History document

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched
USPC - 382/158; 704/9; 706/46 (keyword delimited)

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
See Search History document

C. DOCUMENTS CONSIDERED TO BE RELEVANT

<table>
<thead>
<tr>
<th>Category</th>
<th>Citation of document, with indication, where appropriate, of the relevant passages</th>
<th>Relevant to claim No.</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>US 20110243377 A1 (CASAMONA et al) 06 October 2011 (06.10.2011) entire document</td>
<td>1-3</td>
</tr>
<tr>
<td>A</td>
<td>US 20150294158 A1 (DISNEY ENTERPRISES, INC) 15 October 2015 (15.10.2015) entire document</td>
<td>1-3</td>
</tr>
</tbody>
</table>

Further documents are listed in the continuation of Box C. See patent family annex.

* Special categories of cited documents:
  "A" document defining the general state of the art which is not considered to be of particular relevance
  "E" earlier application or patent but published on or after the international filing date
  "L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
  "O" document referring to an oral disclosure, use, exhibition or other means
  "P" document published prior to the international filing date but later than the priority date claimed
  "T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
  "X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
  "Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
  "&" document member of the same patent family

Date of the actual completion of the international search
20 July 2017

Date of mailing of the international search report
1 AUG 2017

Name and mailing address of the ISA/US
Mail Stop PCT, Attn: ISA/US, Commissioner for Patents
P.O. Box 1450, Alexandria, VA 22313-1450
Facsimile No. 571-273-8300

Authorized officer
Blaine R. Copenheaver
PCT Helpdesk: 571-272-4300
PCT OSP: 571-272-7774

Form PCT/ISA/2 10 (second sheet) (January 2015)
This international search report has not been established in respect of certain claims under Article 17(2)(a) for the following reasons:

1. Claims Nos.:
   because they relate to subject matter not required to be searched by this Authority, namely:

2. Claims Nos.:
   because they relate to parts of the international application that do not comply with the prescribed requirements to such an extent that no meaningful international search can be carried out, specifically:

3. Claims Nos.: 4-19
   because they are dependent claims and are not drafted in accordance with the second and third sentences of Rule 6.4(a).

This International Searching Authority found multiple inventions in this international application, as follows:

1. As all required additional search fees were timely paid by the applicant, this international search report covers all searchable claims.
2. As all searchable claims could be searched without effort justifying additional fees, this Authority did not invite payment of additional fees.
3. As only some of the required additional search fees were timely paid by the applicant, this international search report covers only those claims for which fees were paid, specifically claims Nos.:
4. No required additional search fees were timely paid by the applicant. Consequently, this international search report is restricted to the invention first mentioned in the claims; it is covered by claims Nos.:

**Remark on Protest**
- The additional search fees were accompanied by the applicant's protest and, where applicable, the payment of a protest fee.
- The additional search fees were accompanied by the applicant's protest but the applicable protest fee was not paid within the time limit specified in the invitation.
- No protest accompanied the payment of additional search fees.