

[USPTO PATENT FULL-TEXT AND IMAGE DATABASE](#)[Home](#)[Quick](#)[Advanced](#)[Pat Num](#)[Help](#)[Bottom](#)[View Cart](#)[Add to Cart](#)[Images](#)

( 1 of 1 )

**United States Patent  
Maslan****9,124,783  
September 1, 2015**

Method and system for automated labeling at scale of motion-detected events in video surveillance

**Abstract**

A method and system are provided for automated labeling at scale of motion-detected events in video surveillance imagery that may be private to and viewable only by the owner of that imagery. Further provided are processing, organization, and archival aspects related to such automated labeling at scale.

**Inventors:** Maslan; Jonathan Carter (Burlingame, CA)**Applicant:**           Name                   City    State Country Type

CAMIOLOG, INC. Burlingame CA    US

**Assignee:** CAMIOLOG, INC. (Burlingame, CA)**Family ID:** 47992228**Appl. No.:** 13/629,048**Filed:**       September 27, 2012**Prior Publication Data****Document Identifier**

US 20130083198 A1

**Publication Date**

Apr 4, 2013

**Related U.S. Patent Documents****Application Number**

61542062

**Filing Date**

Sep 30, 2011

**Patent Number****Issue Date****Current U.S. Class:****1/1****Current CPC Class:**

H04N 7/188 (20130101)

**Current International Class:**

H04N 7/18 (20060101)

**Field of Search:**

;348/155

**References Cited [\[Referenced By\]](#)**



images from a plurality of source cameras, said plurality comprising said source camera, said processing by: receiving and decrypting an encrypted token that uniquely identifies said end user entity and said source camera; upon authorizing said decrypted token, receiving at least a first incoming image frame, and at least a second incoming image frame from said source camera; comparing one or more objects on said at least second incoming image frame with one or more objects on said at least first incoming image frame and identifying significant motion based on said comparison, wherein significant motion is identified in part by computing a delta of change between any of said objects on said at least second incoming image frame and said objects on said at least first incoming image frame; for each identified significant motion, determining a motion zone, wherein said motion zone is the zone of said at least second incoming image frame on which said significant motion overlays; for each motion zone: computing an amount of an overlap with a candidate labeled zone from said stored correlated labeled zones; and measuring said overlap among said motion zone and said labeled zone, determining therefrom whether further granular analyses are required to be performed, and performing said further granular analyses when required; generating and indexing a label hit for each matching motion zone with candidate labeled zone; and generating tasks for each generated label hit; wherein at least one of the steps is performed by a processor.

2. The computer-implemented method of claim 1, wherein said correlating said at least one label name with said at least one zone for said image source comprises translating said at least one labeled zone into percentage offsets from the edges of said source image.

3. The computer-implemented method of claim 1, wherein performing further granular analyses comprises: performing another iteration by returning to said step of comparing objects wherein said comparing uses any of: pixel resolution in analysis of specific bounding boxes, image filtering parameters, and degree of computational effort in searching for contiguous zones.

4. The computer-implemented method of claim 1, wherein a task of said generated tasks is sending a notification to the end user entity.

5. The computer-implemented method of claim 1, wherein label hits are correlated each with any one of source images sourced from a particular camera of a particular user and wherein the user is able to perform any of: view, review, and rate the accuracy of said label hit.

6. An apparatus for smart labeling at scale of motion detected events in video surveillance, comprising: a server adapted to label at least one zone of a source image, said source image from a source camera of a plurality of source cameras, and to store said label zone by comprising: a receiver adapted to receive said source image with a corresponding upload token, wherein said upload token contains data identifying said source camera and an end user entity; an authorizing processor adapted to authorize said token; a presenter adapted to present, using said token, said source image to said end user entity for labeling; a receiver adapted to receive at least one label name of said at least one zone of said source image, to correlate said at least one label name with said at least one zone for said image source, and store said correlated labeled zone; and a server adapted to process, in real time, label hits for incoming images from a plurality of source cameras, said plurality comprising said source camera, by comprising: a decryptor adapted to receive and decrypt an encrypted token that uniquely identifies said end user entity and said source camera; a receiver adapted to receive, upon authorization of said decrypted token, at least a first incoming image frame and at least a second incoming image frame from said source camera; a comparing component adapted to compare one or more objects on said at least second incoming image frame with one or more objects on said at least first incoming image frame and identifying significant motion based on said comparison, wherein significant motion is identified in part by computing a delta of change between any of said objects on said at least second incoming image frame and said objects on said at least first incoming image frame; a determiner adapted to determine, for each identified significant motion, a motion zone, wherein said motion zone is the zone of said at least second incoming image frame on which said significant motion overlays; for each motion zone: a computing component adapted to compute an amount of an overlap with a candidate labeled zone from said stored correlated labeled zones, and measure said overlap among said motion zone and said labeled zone, determine therefrom whether further granular analyses are required to be performed, and perform said further granular analyses when required; and a generator adapted to generate and index a label hit for each matching motion zone with candidate labeled zone, and generate tasks for each generated label hit.

7. The apparatus of claim 6, wherein said correlate said at least one label name with said at least one zone for said image source comprises to translate said at least one labeled zone into percentage offsets from the edges of said source image.

8. The apparatus of claim 6, wherein to perform further granular analyses comprises to perform another iteration by returning to said step of comparing objects wherein said comparing uses any of: pixel resolution in analysis of specific bounding boxes, image filtering parameters, and degree of computational effort in searching for contiguous zones.

9. The apparatus of claim 6, wherein a task of said generated tasks is sending a notification to the end user entity.

10. The apparatus of claim 6, wherein label hits are correlated each with any one of source images sourced from a particular camera of a particular user and wherein the user is able to perform any of: view, review, and rate the accuracy of said label hit.

11. A non-transitory computer readable storage medium having stored thereon a computer program for smart labeling at scale of motion detected events in video surveillance, said computer program comprising a program code which, when executed by a processor, performs the steps of: at a server, labeling at least one zone of a source image, said source image from a source camera of a plurality of source cameras, and storing said label zone by: receiving said source image with a corresponding upload token, wherein said upload token contains data identifying said source camera and an end user entity; upon authorizing and using said upload token, presenting said source image to said end user entity for labeling; and receiving at least one label name of said at least one zone of said source image, correlating said at least one label name with said at least one zone for said source image, and storing said correlated labeled zone; and at a server in real time, processing label hits for incoming images from a plurality of source cameras, said plurality comprising said source camera, said processing by: receiving and decrypting an encrypted token that uniquely identifies said end user entity and said source camera; upon authorizing said decrypted token, receiving at least a first incoming image frame, and at least a second incoming image frame from said source camera; comparing one or more objects on said at least second incoming image frame with one or more objects on said at least first incoming image frame and identifying significant motion based on said comparison, wherein significant motion is identified in part by computing a delta of change between any of said objects on said at least second incoming image frame and said objects on said at least first incoming image frame; for each identified significant motion, determining a motion zone, wherein said motion zone is the zone of said at least second incoming image frame on which said significant motion overlays; for each motion zone: computing an amount of an overlap with a candidate labeled zone from said stored correlated labeled zones; and measuring said overlap among said motion zone and said labeled zone, determining therefrom whether further granular analyses are required to be performed, and performing said further granular analyses when required; generating and indexing a label hit for each matching motion zone with candidate labeled zone; and generating tasks for each generated label hit.

12. The non-transitory computer readable storage medium of claim 11, wherein said program code further comprises correlating said at least one label name with said at least one zone for said image source comprising translating said at least one labeled zone into percentage offsets from the edges of said source image.

13. The non-transitory computer readable storage medium of claim 11, wherein said program code further comprises performing further granular analyses by performing another iteration by returning to said step of comparing objects wherein said comparing uses any of: pixel resolution in analysis of specific bounding boxes, image filtering parameters, and degree of computational effort in searching for contiguous zones.

14. The non-transitory computer readable storage medium of claim 11, wherein a task of said generated tasks is sending a notification to the end user entity.

15. The non-transitory computer readable storage medium of claim 11, wherein label hits are correlated each with any one of source images sourced from a particular camera of a particular user and wherein the user is able to perform any of: view, review, and rate the accuracy of said label hit.





limited to being due to invalid authentication credentials or unauthorized state of the account, e.g. of the user.

#### Step 4. Store Content, Populate Cache, Queue Analysis as New Task

In an embodiment, uploaded content may be stored and indexed as exemplified in steps 5, 6 and 7 hereinbelow which use storage mechanisms for the image, difference task queue, and memcache image sequence, respectively. A cache of the most recent incoming frame sequences for each camera may be provided and may be populated in sequential order such that subsequent analyses have fast, efficient access to such content. In an embodiment, analysis of new content may be queued as a new task. In an embodiment, when content first comes arrives, the system stores such content and subsequently schedules a new task, which is to analyze what was just stored.

#### Step 5. Store Raw Content

In an embodiment, raw content including but not limited to video, audio, images, and meta-data like timestamps may be stored in a particular storage, e.g. high replication storage, accessible to many server instances. For example, images and video may be stored with Google's High Replication Datastore to ensure that content is accessible for analysis among tasks running in parallel across one or more datacenters.

#### Step 6. Generate and Populate the Image Difference Task Queue

In an embodiment, an image difference task is queued for execution and populated with a definition of the task that can be executed with idempotence in order to tolerate failures and re-tries across distributed computers. Such image difference task queue may be defined as an analysis work unit to be completed when incoming frames are compared to previous frames. The analysis work unit can include pair-wise image differences, statistical patterns across multiple images, and recursive analysis of particular regions of the scene over time. The number of frames under consideration may vary, and the analysis work unit is complete after having received and considered the input images required to judge the changes in the particular frame. Once that frame's motion has been analyzed, the results of that analysis are stored in the HRD datastore to be used as input for subsequent analyses of motion, matching of labels, and triggering of user notifications. The task queue itself is a Google App Engine data structure backed by both cached memory and High Replication Datastore with task scheduling and re-try-upon-failure rules that ensure each analysis work unit is completed while executing the tasks across multiple dynamically provisioned servers.

In an embodiment, such task includes comparing at least two frames, could be more than two frames, to determine what has changed in the scene. For example, according to an embodiment, a sequence of comparisons of images is processed. It should be appreciated that such sequence of comparisons may include a comparison of one frame and the one immediately prior or may be a sensitive analysis to a sequence in the last few minutes. Regardless of the characterization of the type of comparisons, embodiments herein determine what has changed over the last few frames of motion.

An embodiment can be understood with reference to FIG. 7. For example, in a sequence in which frequent motion from blowing trees 701 is mixed with the motion of a postman delivering mail 702, the analysis of motion differences looks across the prior 10 minutes of motion patterns to isolate the postman as the new motion of interest instead of including the motion of the recurring moving branches and shadows. By comparing the motion difference identified in the postman frame with the pattern of motion differences in the prior frames, the postman frame is more accurately labeled as motion warranting the user's interest and worthy of being labeled as interesting motion.

#### Step 7. Populate Frame Sequence Cache

In an embodiment, a frame sequence cache may be provided and populated where such cache maintains for each camera source the most recently received image frames in sequential order. For example, a user's cameras named "front" and "back" might each maintain a cache of the last 30 motion images received from the camera in time-sequential order. The rate at which motion images are cached varies by camera, because the scene observed by each camera varies in the amount of motion occurring in that scene. Though stored permanently in Google App Engine (GAE) High Replication Datastore, each camera's motion sequence is also cached in memory using Google Memcache that is accessible to multiple servers such that motion

analyses have fast in-memory access to the entire frame sequence under consideration.

#### Step 8. Label Scene Regions

In an embodiment, this step begins a process from an end user set-up viewpoint. The end user may define label names s/he wants associated with particular regions of an observed scene. In an embodiment, from a browser, the user may define one or more contiguous or non-contiguous regions by labeling such regions. For example, in an implementation, such regions may be labeled by any of click, drag, and extended-select of html/css overlays on the scene. Examples of such labels may include but are not limited to "mailbox", "entrance", "expensive supplies." It should be appreciated that examples herein currently allow labeling in any browser as an innovative way to enable users to designate zones of interest without having to learn camera-specific tools or use camera-specific installed software. However, embodiments herein allow labeling in smartphone applications or in automated suggestions and so forth.

#### Step 9. Correlate Labels with Image Source and Translate into Percentage Offsets

In an embodiment, the system uses the decoded upload token to identify the user and camera image to present to the user for labeling. It should be appreciated that Step 9 reflects that the user may need to see a sample image on which to draw those labeled zones. Thus, this step may be the step that identifies a recently uploaded image to present to the user in performing step 8. The upload token used to store a recent image is used to identify the image as belonging to a particular user's camera in order to present a sample image corresponding to the stored recent image for the user's labeling of regions of the scene on the image. In an embodiment, the labeled regions are translated into percentage offsets from the edges of the image such that such labeled regions may be processed at any image resolution.

For example, when a user draws a 40.times.20 (width.times.height) pixel rectangular region in the center of a sample image with pixel dimensions 640.times.480, that region is translated into a percentage offset calculated as: top=230/480=47.92% left=300/640=46.88% width=40/640=6.25% height=20/480=4.17%

By recording the user's labeled regions as percentage offsets, the motion analysis work unit is free to choose variably sized input images to optimize cost and performance.

It should be appreciated that line from Step 4 to Step 9 in FIG. 1 may represent when one of the recently stored images is used as the canvas for which the regions of the scene may be labeled. In an embodiment, the system stores the labeled regions as meta-data about the content source. Thus, those labeled regions may be part of the logical definition of the camera, however actually may be stored independent of the camera such that they may vary over time, e.g. the same camera may have labeled "tent" in the summer but removed that label once the tent was removed.

#### Step 10. Store Labeled Regions

In an embodiment, labeled regions may be stored as associated with a particular camera source and user. That is, the camera source is identified uniquely by the combination of its owning user account and its name. For example, the camera source named "front yard" that is owned by user id "1180036789" uniquely identifies that camera source.

Thus, in an embodiment, regions that apply for a particular camera source are stored. For example, in an embodiment, an end user has defined boxes or polygonal types of areas. Such defined areas are saved as part of the definition of the particular camera.

#### Step 11. Compare Images and Generate Difference Images Attributes, Queue Label Analysis

An embodiment compares incoming images with those proceeding to identify significant motion. In an embodiment, motion areas may be weighted and indexed, e.g. in a Quad Tree data structure, for subsequent label hit-testing. For purposes of discussion herein, `label-hit testing` may be defined as the process by which interesting motion identified in a particular sequence of image frames is compared for intersection with the user's labeled regions of interest. For example, when the region containing the approaching postman is detected as the region of interesting motion, that region is intersected with the regions defined by the user as

"walkway" and "front lawn". If the postman's motion region overlaps with both the "walkway" and "front lawn" regions of a particular frame, then the postman's motion may be considered a "label-hit" on "walkway" and/or "front lawn".

An embodiment may additionally generate label-hit analysis tasks. For purposes of discussion herein, `a label-hit analysis task` may be defined as the analysis work unit required to determine how the interesting motion that overlaps with labeled regions of interest is classified. For example, when there are candidate matches identified by the overlap among regions with interesting motion and regions with labels defined by the user, a label-hit analysis task may be generated.

#### Step 12. Store Image Difference Analyses

In an embodiment, output of each image difference is stored. In an example implementation, such output may be stored in Quad Tree form in Google's High Replication Datastore.

#### Step 13. Store Label Hit Analyses

In an embodiment, a label hit analysis task queue is populated which defines the work unit(s) to be completed in comparing the image difference output with the stored user-labeled regions. In an embodiment a task is a unit of work scheduled for execution. The work performed by that work unit varies based on the actual computations needing to be done. In an embodiment, the task is stored using a GAE Task Queue that schedules execution with inputs referencing the datastore key of the image stored in Step 5. That unique datastore key is used to formulate the keys and retrieve the image differences stored in Step 12 and the labeled regions stored in Step 10.

#### Step 14. Analyze Label Hits, Queue Detailed Analysis

In an embodiment, for motion regions, e.g. as identified in Step 11 and stored in Step 12, with candidate label matches, as defined by Step 9 and stored in Step 10, judge the overlap among motion regions and labeled regions to determine whether more granular and possibly more CPU-intensive analyses are required to be performed. Thus, for example, the motion stored in the image difference Quad Tree may be searched for hits with each labeled region. In an embodiment, such Quad Tree hits each may be scored with weightings based on a plurality of factors. Examples of such factors may include but are not limited to the size of the matching region, the horizontal and vertical proportions of contiguous matches, ratings of previous analyses submitted by users, and patterns observed over time. For example, the output of a label-hit analysis may be represented with an overall motion size score and individually scored label hits that calculated the size of the motion region (`ms`) and the size of the labeled region (`ls`) represented in JSON as:

```
TABLE-US-00001 Motion: 2.343750, {u`front lawn`: {`ls`: 11.848958333333334, `ms`: 0.0}, u`sidewalk`: {`ls`: 2.9947916666666665, `ms`: 0.0}, u`street`: {`ls`: 6.380208333333333, `ms`: 2.2135416666666665}, u`driveway`: {`ls`: 17.057291666666668, `ms`: 0.0}, u`walkway`: {`ls`: 11.979166666666666, `ms`: 0.0}}
```

In this example, the labeled regions `front lawn`, `driveway`, `sidewalk`, and `walkway` had no motion (0.0) and the labeled regions `street` contained nearly the entire are of motion (2.2135416666666665 of 2.343750)

#### Step 15. Determine Whether More Details Analysis is Required, Tier n+1?

In an embodiment, when the scoring requires more detailed analysis, then perform another iteration of the image difference task, i.e. return to Step 6, with work unit parameters that describe, for example, that the image diff be performed at higher resolution of 640.times.480 pixels rather than the previous 160.times.120 resolution. The work unit parameters are supplied as inputs to the task scheduled for execution using GAE Task Queue and may vary factors such as but not limited to pixel resolution used in analysis of specific bounding boxes, image filtering parameters, and the degree of effort in searching for contiguous regions. For example, if the motion overlapped small and closely aligned labeled regions, the new task request another pass at higher resolution by adding the task parameter &size=640.times.480. Or, if only a portion of the image needs higher resolution analysis, the parameters may be:

TABLE-US-00002 &size=640x480@ions=[{"top": "47.92%", "left": "46.88%", "height": "4.17%", "width": "6.25%"}, {"top": "7.02%", "left": "0%", "height": "10.27%", "width": "3.45%"}]

The degree of effort varies by factors such as the resolution of the image because higher resolutions require more CPU processing to compare large matrices, the timespan of the analysis, because considering a greater number of frames may require more computation, and the depth of the search for contiguous regions, as color comparisons and recursive search consume more CPU processing time. In an embodiment, measurement is performed in CPU, memory, and disk writes. One large factor in image difference analysis is CPU. As more CPU is used, more servers may be required to keep up with incoming stream of images. Thus, the embodiment tries to optimize the image analysis for accuracy without creating huge CPU/server costs.

When no further analysis is warranted, then the label hits may be processed as shown in Step 20 hereinbelow.

#### Step 16. Label Hit Analyses

In an embodiment, output of the last iteration of label matches are stored. For example, this represents in JSON a single label-hit on the labeled region "street":

TABLE-US-00003 {"motion size": 2.34375, "alerts": [ ], "label\_hit\_info": {"front lawn": {"ls": 11.848958333333334, "ms": 0.0}, "sidewalk": {"ls": 2.9947916666666665, "ms": 0.0}, "street": {"ls": 6.380208333333333, "ms": 2.2135416666666665}, "driveway": {"ls": 17.057291666666668, "ms": 0.0}, "walkway": {"ls": 11.979166666666666, "ms": 0.0}}}

#### Step 17. Correlate Label Hits with Image Source

In an embodiment, label hits are correlated each with any one of the images sourced from a particular camera of a particular user. Such hits are correlated such that the user may view, review, and rate the resulting label-hit of the analyses and also the motion analysis prior to the label-hit analysis as described in Step 18 hereinbelow.

#### Step 18. Rate Analyses

In an embodiment, a user may select and rate particular regions of the motion labeling analyses. Such rating may provide training data that improves the accuracy of subsequent analyses. Because particular imagery may be private and viewable only by a particular user owning the surveillance imagery, it should be appreciated that such human feedback provides judgment from perhaps the sole person able to judge the quality of the automated calculations.

#### Step 19. Store User-Rated Analyses

In an embodiment, such ratings submitted by the user may be stored, e.g. for use in subsequent analyses.

#### Step 20. Process Label Hits

In an embodiment, resulting labeled motion analyses are indexed for the organization and presentation of labeled motion. Also, create and queue tasks triggered by label hits including but not limited to sending notifications to users.

#### Step 21. Populate Notification Task Queue Storage

In an embodiment, such notification task queue stores the work units to be completed by notifying users of the label matches via various communication channels. Examples of such various communication channels may include but are not limited to instant message, email, and placing in a particular inbox, each according to preferences of each user.

#### Step 22. Send Notifications, Offer Option to Rate

In an embodiment, notifications may be sent via preferred channels of users. In an embodiment, such

notifications may include links to the indexed labeled motion in HTML or include data encoded and delivered over any other protocol like UDP or XMPP. In an embodiment, from any frame viewed, the user may rate the accuracy and relevance of the labeled motion either explicitly with a rating action, or implicitly with actions such as but not limited to clicking, playing, hiding, deleting, or saving a particular frame.

FIG. 2 and FIG. 3 each represent an example frame of two sequential frames. Together, such frames illustrate significant motion labeled in the two sequential frames from one camera. For example, in FIG. 2 a person is detected on the sidewalk 201 and a dog is detected on a different part of the sidewalk 202 as well. In FIG. 3 the person has moved since the frame of FIG. 2 and is now detected further along the sidewalk 301 and the dog is detected on the lawn 302.

In an embodiment, by subtracting transformations of the previous frame, e.g. as detected at 201 and 202, from the current frame, e.g. as detected at 301 and 302, and applying filters to reduce unwanted motion from, for example, blowing leaves and moving branches, the dog may be labeled as motion on the "front lawn" and the person may be labeled as motion on the "walkway".

Thus, FIG. 2 and FIG. 3 show the difference between the prior frame and the subsequent frame. The yellow highlights are an example of what changed, such as if an individual was viewing the dog and the person.

FIG. 4 illustrates a user-labeled camera scene. In an embodiment, a user may select any contiguous or non-contiguous regions of a scene and assign a label. Typically, such labels may be used for characterizing interesting motion in the scene, specifying preferred notification rules, and for reducing the computational costs of image analysis. For example, in the screen shot of FIG. 4, a user provided the following labels: street 401, swing 402, sidewalk 403, front lawn 404, walkway 405, and driveway 406. It should be appreciated that such labels are way by example and are not meant to be limiting.

In an embodiment, regions may also be specified or defined. Examples of specified regions may be understood with reference to FIG. 4. A street region is specified by a step-like boundary 407. As another example, a driveway region 408 is also shown as bounded by a step-like boundary. Again, such examples are for illustrative purposes and are not meant to be limiting.

Rather than requiring any specialized client applications, an embodiment enables a user to use any web browser to draw such labeled regions. For example, in an embodiment, regions and labels may be drawn as HTML or Cascading Style Sheets (CSS) overlays on top of the image. For purposes of discussion herein, an overlay may be defined as percentage offsets from the edges of an image such that label-hit analysis may be performed at any pixel resolution, e.g. for computational efficiency.

FIG. 5 illustrates a mechanism by which a user may provide an explicit rating of labeled motion in a frame, according to an embodiment. In this illustration, which is not meant to be limiting, a particular frame 500 is provided. Also provided in frame 500 are three buttons that may be selected to indicate whether such frame depicts an accurate triggering of motion. In frame 500, a GOOD button 501 is provided such that when it is selected, a learning process provided in an embodiment, may use such selection as input for improving accuracy of detecting significant motion. As well, an embodiment may provide a BAD button 502, which when selected indicates that the frame does not accurately indicate significant motion. As well, an embodiment may provide an UNSURE button 503, which when selected indicates that it is not clear whether the frame accurately indicates significant motion. It should be readily apparent that the particular details for indicating accuracy of motion are by way of example and that a wide variety of implementations are contemplated.

Further according to FIG. 5, a user may select portions of the motion identified as significant. For example, as shown at the bottom of frame 500, a user may "rubber band" select a collection of frames, for example, by using arrow keys in conjunction with pressing and holding the left mouse button. In the embodiment, when the user characterizes relevance and accuracy of such portion of the labeled motion, such characterizations may be as an input training signal for future analyses. Though there also may be implicit training signals, e.g. clicks to play labeled sequences, drill-downs on specific labels and times, etc., such explicit feedback also may be important because a particular user, e.g. the owner of the camera, may be the only human or entity authorized to view or be presented with the analysis results and judge or determine their quality.

## Large-Scale Labeling

An embodiment provides a large-scale labeling process of labeling camera images, where large-scale labeling of hundreds of thousands of input or arriving images is performed. In an embodiment, such large-scale labeling may be achieved by the herein described steps around performing the differences and labeling motion and images. It should be appreciated that such large-scale labeling process may be performed in real time with cloud-based processing at an economical price by the way that the problem is segmented in accordance with an embodiment. Real-time labeling of significant motion in frames is provided in such a way that may scale to hundreds of thousands of cameras, which is not provided presently by the prior art.

Thus, in accordance with an embodiment, any digital camera may be used. That is, such camera used in embodiments herein are not required to be loaded with particular intelligence around recognizing objects, such as a person or a car. Such cameras with such built-in intelligence require particular processing, which creates more expensive cameras that may not adapt and learn over time. Thus, embodiments herein allow using "dumb" or "stupid" cameras, e.g. without particular intelligence such as recognition intelligence. Thus, with such cameras that do not require such built-in intelligence, images are transmitted to a server, such as on the cloud. Such images are labeled with significant motion in the cloud, economically and at scale.

## Camera

Embodiments herein use cameras that are configured to trigger an event. Examples of such types of events that get triggered may include blowing leaves, shadow changes, bugs, all kinds of night spiders, etc. Thus, in an embodiment, cameras are allowed to send forms of events that matter to the camera or, put another way, per the configuration of the individual camera.

## Token

In an embodiment, a camera is provided with a token that permits the camera to upload only to the user's account without also granting access to the user's account. Put another way, an embodiment provides a token that enables contribution of content, e.g. but not limited to video, images, and audio, without giving the camera the ability to sign-in to the user's account, which may be a security risk. Such token may be considered to provide a type of security mechanism and may serve to provide such consumer-device cameras with a security mechanism that is otherwise nonexistent. Thus, in an embodiment, a camera uploads one or more images to a particular account without having to have access to the account, however upload using the token. In an embodiment, the camera receives such token by way of a configuration of the camera that just indicates to upload to a directory. For example, an Upload Token may include encrypted elements like the user id, date created and secret used to encrypt a token that may look as follows:

AQCH1tA8Jf6yldAnzb6JdyzEj1Xt6TcoeUB76haDaXDzl3F

or a shortened form like:

ic9t69

For example, instead of a camera having to know conventional user names and passwords, such cameras have this token that is used to upload. It should be appreciated that in an embodiment, if someone were to steal the token, a new token may be reissued. For example, the user generates a new Upload Token by entering a new Upload Code such as "my secret" that is used as one of the inputs to create the new encrypted Upload token. An embodiment may be understood with reference to FIG. 15 showing an application window which a user may use to generate a new token.

Thus, using the provided token separates the permission to upload from the permission to access and view content. Thus, such token provides the consumer electronics with more security because the device is not given the permission to actually access the particular account.

## Cache

In an embodiment, caching may be part of contributing to the scalability aspect of the video surveillance

scheme provided herein because such caching allows for receiving tens of thousands of images when it is not reasonable to assume that such images may be read and written to disk, or repeatedly transformed in various matrix operations, much of the time. Thus, embodiments herein facilitate keeping a buffer of incoming images in a fast cache that makes it more cost effective and fast. It should be appreciated that another part of the cache advantage is that intermediate analysis products, such as but not limited to the matrix representation of an image split into RGB color space, may not have to be re-created again for the comparison with subsequent frames.

## Segmenting

An embodiment provides a pipeline process that takes what is usually a very expensive computation and segments such computation into smaller tasks such that such computation and other related processes may scale across a large number of cameras. Thus, embodiments herein allow performing video surveillance which had typically been performed on either a dedicated processor on a camera or a dedicated processor in some kind of data center and, instead, segmenting the whole process into smaller tasks that can be performed efficiently at scale.

## Image Difference Task Queue

In an embodiment, an image difference task queue contains a sequence of images to be compared. A processor determines a delta between the two images with the objective of identifying one or more significant objects that may have changed. An embodiment may be configured to determine a significant change versus an artifact such as blowing leaves or shadow changes, etc. For example, when blowing leaves on a sunny day cast many small shadows on the lawn, a simple image difference between two images would produce lots of "false-positive" motion since the light and dark spots on the lawn may change each second. An embodiment may instead be smarter about masking irrelevant motion by noticing patterns in motion across the prior frame sequences, analyzing hue variations, and learning patterns specific to but not limited to the camera source, time of day, and location.

## Time Sequencing

An embodiment provides an efficiency of the cache and normalizing of time stamps across the incoming images. Such efficiency and normalization are important because cameras may have varying time stamps and the cameras may be in communication with different servers. An embodiment takes such disparate data corresponding to the images and places such images based on the data onto a common timeline, thereby establishing a time sequence. For example, a camera whose clock is two minutes fast, is not adjusted to daylight savings time, and is not set to the right time zone may be corrected by but not limited by the following: noting the server timestamp at the time the image transfer began, the timestamp of the image storage, and the deviation among those timestamps and the timestamp supplied in the filename of the image uploaded by the camera source. Typically, the server timestamps alone may correctly sequence the images. However, when many images arrive across many servers with varying processing loads, connection bandwidths, and disk write latencies, an embodiment corrects sequencing errors by observing the discrepancies between the known-consistent server timestamps, e.g. in Greenwich Mean Time (GMT), and those reported by the camera source itself. Thus for example, when the camera's filename is /front/MA.sub.-2012-09-24.sub.--16-01-17.sub.--178.jpg and the upload began at GMT timestamp 2012-09-24T23:04:23.454684 and the server stored the image at 24/Sep/2012:16:04:28-0700, the system may recognize that the camera source is consistently three minutes and five seconds behind.

Thus, it should be appreciated that such frame sequence cache is important for the efficiency as well as for the time sequencing of images from the same cameras. Time sequencing of images from the same cameras is important because a plurality of servers may be fielding the incoming images, which may be processed at different points in time.

In an embodiment, even when frames samples may be only a second apart, part of the role of the cache sequence is to keep the frames in order such that they may be compared, e.g. from T0 to T1 to T2 in terms of time.

Thus, an embodiment keeps an efficiently-sequenced set of images ready to be compared.

## Regions/Zones and Labels

In an embodiment, before receiving images for monitoring a scene, a user may designate, e.g. from a browser or from another application window, certain regions of the camera's view thereby creating different zones or regions of a scene. Examples of such zones may include but are not limited to walkway, doorway, front yard, and so forth. Importantly, embodiments herein enable labeling such regions in the browser or other client window. Examples of screenshots showing how zones may be designated and labeled are shown in FIG. 8 and FIG. 9. In these examples, FIG. 8 is a sample screen print from a first camera of a house showing zone definitions, according to an embodiment. FIG. 9 is a sample screen print from a second camera at the front of the house in FIG. 8 showing zone overlap, according to an embodiment.

Following are examples showing both a sample html markup and a reference to the JQuery and CSS used to style the overlay on top of the image.

For example, a recently received camera image is presented to the user in an HTML page that includes an overlay grid. The overlay grid is a series of <div> tags distributed on top of the image and selectable by mouse or touch via JQuery operations on the CSS-styled <div> tags. For example the definition of the "walkway" zone might look like:

```
TABLE-US-00004 <div class="m1 blockSAVED s-walkway"
style="left:0%;top:66%;width:2%;height:2%;"></div> <div class="m1 blockSAVED s-walkway"
style="left:0%;top:68%;width:2%;height:2%;"></div> <div class="m1 blockSAVED s-walkway"
style="left:0%;top:70%;width:2%;height:2%;"></div> etc...
```

When the user saves the defined zone, the browser sends the selected <div> tags to the server for interpretation and storage as a labeled zone. Thus, an embodiment allows a user from any camera to view a sample image of a scene and define which parts of the image to be labeled. An example may be understood with reference to FIG. 4, which presents a sidewalk, front lawn, swing, and walkway, which are labeled regions of the scene.

## Correlation of Labeled Regions with Motion

An embodiment receives incoming image and may determine from which user and camera the image arrived. For example, an uploaded image may arrive with the server path such as but not limited to:

```
TABLE-US-00005 /upload/AQD5IAMiXDMjfmWHD0yvJkIa0YPoAqJ2jhbYfXEILLTw/front/ MA_2012-
09-24_16-01-17_178.jpg
```

Such path is translated into a unique camera source owned by a distinct user by:

- a) deciding the encrypted upload token to verify the user id; and
- b) noting the camera source name in the path (e.g. "front").

Subsequently, the system may correlate labeled regions corresponding to the incoming image with motion observed from the camera, based on the incoming images. For example, an embodiment may detect that there is somebody walking on a walkway. In an embodiment, boundaries of the defined regions are detected and are intersected with the motion that happened for that camera image.

In an embodiment, images arrive with an encrypted token and the system reads the token to determine the original user and camera. Having determined the original user and camera, the system may retrieve the labeled regions that the user had previously defined. For example, after decoding the Upload Token to determine the user id and concatenating that user id with the camera source name, such unique combination is used as the datastore key to retrieve any labeled regions defined for that camera source.

## Perform Difference







An embodiment enables labeling the results of different smart analyses. For example, a user may want a labeled notification confirming that his grandmother is living alone independently and safely by defining a label as a combination of factors such as but not limited to: a zone hit at the "coffee maker" between the hours of "8 am-10 am" of a "human sized object" for duration of "more than 2 seconds". An embodiment not only reduces noise levels, but allows intelligent labeling decisions while looking at or analyzing the motion.

### Exemplary Use Cases

Some example implementations of embodiments may include but are not limited to the following.

#### Construction Site Monitoring

In an embodiment, contractors often must leave valuable tools, materials, and vehicles on job sites unattended and at risk of theft. An embodiment enables contractors to place an inexpensive camera on site to receive instant notifications of, for example, after-hours movement of such tools, materials, or vehicles, which may include: a tractor, toolbox, or stack of copper pipes. Such contractors may be able to jump with one-click to a specific video segment that captured that a particular labeled motion corresponding to the after-hours movement of such tools, materials, or vehicles.

#### Senior Independent Living

Older people often prefer to live independently though their family's concerns over health and safety of undetected accidents or health incidents may push them towards assisted living. An embodiment enables an inexpensive camera to be placed in a routinely trafficked area of the home such that the camera may operate as a monitor for confirming that normal routines of the person living independently continue. For example, the refrigerator or coffee maker may be labeled in the image in such a way such that family may be notified when "grandma has not opened the refrigerator/coffee maker this morning." Thus, such labeling may provide an early warning or added assurance.

#### Vacation Home Monitoring

It should be appreciated that unattended vacations homes prompt concerns for owners. For example, an owner of a vacation home or any home may worry and ask himself or herself "did the pipes burst with the recent cold spell?", "are there unexpected intruders at our place?", "who has entered the house in the last 3 days?" etc. Embodiments herein may enable home owners or even renters to browse quickly through significant motion in particular specific areas of interest without having to go through hours of video.

#### Neighborhood Watch

In many communities reduced budgets may mean police must prioritize investigations. For example, police may prioritize by the severity of a crime and by likelihood of sufficient evidence that may lead to conviction of criminals. Embodiments provided herein may be used to assist security surveillance recording by allowing such recording to be accessible to many more entities, e.g. individuals, organizations, communities, etc., than at present. For example, neighbors may collaborate quickly to produce time-sequenced stitched video of event including but not limited to door-to-door solicitors for example that had been suspected of planning burglaries. In an implementation, such event or events may have been labeled with "walkway" motion.

#### An Example Machine Overview

FIG. 6 is a block schematic diagram of a system in the exemplary form of a computer system 600 within which a set of instructions for causing the system to perform any one of the foregoing methodologies may be executed. In alternative embodiments, the system may comprise a network router, a network switch, a network bridge, personal digital assistant (PDA), a cellular telephone, a Web appliance or any system capable of executing a sequence of instructions that specify actions to be taken by that system.

The computer system 600 includes a processor 602, a main memory 604 and a static memory 606, which communicate with each other via a bus 608. The computer system 600 may further include a display unit

610, for example, a liquid crystal display (LCD) or a cathode ray tube (CRT). The computer system 600 also includes an alphanumeric input device 612, for example, a keyboard; a cursor control device 614, for example, a mouse; a disk drive unit 616, a signal generation device 618, for example, a speaker, and a network interface device 620.

The disk drive unit 616 includes a machine-readable medium 624 on which is stored a set of executable instructions, i.e. software, 626 embodying any one, or all, of the methodologies described herein below. The software 626 is also shown to reside, completely or at least partially, within the main memory 604 and/or within the processor 602. The software 626 may further be transmitted or received over a network 628, 630 by means of a network interface device 620.

In contrast to the system 600 discussed above, a different embodiment uses logic circuitry instead of computer-executed instructions to implement processing entities. Depending upon the particular requirements of the application in the areas of speed, expense, tooling costs, and the like, this logic may be implemented by constructing an application-specific integrated circuit (ASIC) having thousands of tiny integrated transistors. Such an ASIC may be implemented with CMOS (complementary metal oxide semiconductor), TTL (transistor-transistor logic), VLSI (very large systems integration), or another suitable construction. Other alternatives include a digital signal processing chip (DSP), discrete circuitry (such as resistors, capacitors, diodes, inductors, and transistors), field programmable gate array (FPGA), programmable logic array (PLA), programmable logic device (PLD), and the like.

It is to be understood that embodiments may be used as or to support software programs or software modules executed upon some form of processing core (such as the CPU of a computer) or otherwise implemented or realized upon or within a system or computer readable medium. A machine-readable medium includes any mechanism for storing or transmitting information in a form readable by a machine, e.g. a computer. For example, a machine readable medium includes read-only memory (ROM); random access memory (RAM); magnetic disk storage media; optical storage media; flash memory devices; electrical, optical, acoustical or other form of propagated signals, for example, carrier waves, infrared signals, digital signals, etc.; or any other type of media suitable for storing or transmitting information.

Further, it is to be understood that embodiments may include performing operations and using storage with cloud computing. For the purposes of discussion herein, cloud computing may mean executing algorithms on any network that is accessible by internet-enabled or network-enabled devices, servers, or clients and that do not require complex hardware configurations, e.g. requiring cables and complex software configurations, e.g. requiring a consultant to install. For example, embodiments may provide one or more cloud computing solutions that enable users, e.g. users on the go, to use an automated labeling of motion-detected event events at scale in video surveillance system on such internet-enabled or other network-enabled devices, servers, or clients. It further should be appreciated that one or more cloud computing embodiments include all or part of such automated labeling of motion-detected event events at scale in video surveillance system using mobile devices, tablets, and the like, as such devices are becoming standard consumer devices.

Although the invention is described herein with reference to the preferred embodiment, one skilled in the art will readily appreciate that other applications may be substituted for those set forth herein without departing from the spirit and scope of the present invention. Accordingly, the invention should only be limited by the Claims included below.

\* \* \* \* \*

