

Package ‘VideoComparison’

July 9, 2014

Version 0.11

Date 2014-07-02

Title Video comparison tool

Author Silvia Espinosa, Joaquin Ordieres, Antonio Bello, Jose Maria Perez

Maintainer Joaquin Ordieres <j.ordieres@upm.es>

Depends R (>= 2.15.2), RJSONIO, RCurl, zoo, stats, pracma, Rcpp (>= 0.10.3)

LinkingTo Rcpp

Suggests MASS

Description It will take the vectors of motion for two videos (coming from a variant of shotdetect code allowing to store detailed motion vectors in json format, for instance) and it will look for comparing taking out the common chunk. Then, provided you have some image's hashes it will compare their signature in order to make up the decision about chunk similarity of two video files.

License GPL (>= 2)

URL <http://www.r-project.org>

NeedsCompilation yes

Repository CRAN

Date/Publication 2014-07-09 11:14:17

R topics documented:

VideoComparison-package	2
Compare2Videos	3
ExtractImgHash	4
ExtractImgPos	5
ExtractMotion	6

h2	7
hh	7
imagehash1	8
imagehash2	8
imgs	8
mm	9
VideoComparison	9
VideoDistance	11
VideoMatch	12
videomotion1	13
videomotion2	14
VideoSearch	14

Index **16**

VideoComparison-package
Comparison between videos.

Description

Comparison between videos by their descriptors (motion curves and image hashes for transitions)

Details

Package:	VideoCompare
Type:	Package
Version:	0.10
Date:	2014-07-02
License:	Under GPL 2.0+
Description:	Functions to compare video motion vector and image hashes

Author(s)

Silvia Espinosa, <silviaespinosag@gmail.com>, Jose Maria Perez, <josem.perez.ramos@gmail.com>, Joaquin Ordieres, <j.ordieres@upm.es>, Antonio Bello, <abello@uniovi.es>.

Maintainer: Joaquin Ordieres <j.ordieres@upm.es>

References

Espinosa-Gutierrez, S., Ordieres-Mere, J., Bello-Garcia, A.: Large scale part-to-part video matching by a likelihood function using featured based video representation. TAMIDA 2013 - Taller de Minería de Datos dentro del IV congreso Español de Informática. 254-257 (2013). http://bioinspired.dacya.ucm.es/maeb2013/images/ActasCAEPIA_final.pdf

Cui, M., Femiani, J., Hu, J., Wonka, P., Razdan, A.: Curve matching for open 2D curves. Pattern Recogn. Lett. 30, 1-10 (2009)

Compare2Videos

Compare two videos starting from their names.

Description

Compare two videos from their names by retrieving their data from the database. A likelihood estimation will be returned, as well as basic parameters for motion matching (correlation, pos on each video and length of the matching).

Usage

```
Compare2Videos(nv1,nv2,stp=10,fsc=0)
```

Arguments

nv1	String with the identification of the first name.
nv2	String with the identification of the second name.
stp	Unit of motion curvature to be considered.
fsc	Way to determine the correlation between videos. When 0 it will adopted the two times of score derived from the motion video analysis. Otherwise the specified factor will be used.

Author(s)

Joaquin Ordieres, <j.ordieres@upm.es>, Antonio Bello, <abello@uniovi.es>.

References

Espinosa-Gutierrez, S., Ordieres-Mere, J., Bello-Garcia, A.: Large scale part-to-part video matching by a likelihood function using featured based video representation. TAMIDA 2013 - Taller de Minería de Datos dentro del IV congreso Español de Informática. 254-257 (2013). http://bioinspired.dacya.ucm.es/maeb2013/images/ActasCAEPIA_final.pdf

Cui, M., Femiani, J., Hu, J., Wonka, P., Razdan, A.: Curve matching for open 2D curves. Pattern Recogn. Lett. 30, 1-10 (2009)

Examples

```
## Requires specific server architecture
# lv<-VideoSearch()
# Compare2Videos(lv[1],lv[2])
```

ExtractImgHash	<i>NoSQL JSON oriented database video frame hash extraction function</i>
----------------	--

Description

A R function allowing to ask to a REST/Json server for the video frame hashes stored there. It will return it a list with the hashes related to the requested image inside the video for those frames.

Following packages are required: RJSON, RCurl

Usage

```
ExtractImgHash(pos, father, url="http://localhost:9200/selected_db/selected_db/_search")
```

Arguments

pos	String describing the ID of the frame we are interested into its hashes
father	String describing the ID key allowing to access the data into the database.
url	String describing the URL which will answer queries in JSON oriented mode.

Author(s)

Joaquin Ordieres, <j.ordieres@upm.es>, Antonio Bello, <abello@uniovi.es>.

References

Espinosa-Gutiez, S., Ordieres-Mere, J., Bello-Garcia, A.: Large scale part-to-part video matching by a likelihood function using featured based video representation. TAMIDA 2013 - Taller de Minería de Datos dentro del IV congreso Espannol de Informatica. 254-257 (2013). http://bioinspired.dacya.ucm.es/maeb2013/images/ActasCAEPIA_final.pdf

Cui, M., Femiani, J., Hu, J., Wonka, P., Razdan, A.: Curve matching for open 2D curves. Pattern Recogn. Lett. 30, 1-10 (2009)

Examples

```
## Requires specific server architecture  
# mm<-ExtractImgHash("000060", "C0031D0")
```

ExtractImgPos	<i>NoSQL JSON oriented database video frame extraction function</i>
---------------	---

Description

A R function allowing to ask to a REST/Json server for the video frame descriptions stored there. It will return it as a vector of strings making clear the positions inside the video for those frames.

Following packages are required: RJSON, RCurl

Usage

```
ExtractImgPos(father, url="http://localhost:9200/selected_db/selected_db/_search")
```

Arguments

father	String describing the ID key allowing to access the data into the database.
url	String describing the URL which will answer queries in JSON oriented mode.

Author(s)

Joaquin Ordieres, <j.ordieres@upm.es>, Antonio Bello, <abello@uniovi.es>.

References

Espinosa-Gutiez, S., Ordieres-Mere, J., Bello-Garcia, A.: Large scale part-to-part video matching by a likelihood function using featured based video representation. TAMIDA 2013 - Taller de Minería de Datos dentro del IV congreso Espannol de Informatica. 254-257 (2013). http://bioinspired.dacya.ucm.es/maeb2013/images/ActasCAEPIA_final.pdf

Cui, M., Femiani, J., Hu, J., Wonka, P., Razdan, A.: Curve matching for open 2D curves. Pattern Recogn. Lett. 30, 1-10 (2009)

Examples

```
## Requires specific server architecture  
# mm<-ExtractImgPos("C0031D0")
```

ExtractMotion

NoSQL JSON oriented database motion vector extraction function

Description

A R function allowing to ask to a REST/Json server for the motion vector of a video stored there. It will return it as a vector of integers.

Following packages are required: RJSON, RCurl

Usage

```
ExtractMotion(father, pos, url="http://localhost:9200/selected_db/selected_db/_search")
```

Arguments

father	String describing the ID key allowing to access the data into the database.
pos	Number of component retrieved. By default 1=> Motion. Alternatives are 2:4 (RGB color component or 5 timeframe)
url	String describing the URL which will answer queries in JSON oriented mode.

Author(s)

Joaquin Ordieres, <j.ordieres@upm.es>, Antonio Bello, <abello@uniovi.es>.

References

Espinosa-Gutiez, S., Ordieres-Mere, J., Bello-Garcia, A.: Large scale part-to-part video matching by a likelihood function using featured based video representation. TAMIDA 2013 - Taller de Minería de Datos dentro del IV congreso Espannol de Informatica. 254-257 (2013). http://bioinspired.dacya.ucm.es/maeb2013/images/ActasCAEPIA_final.pdf

Cui, M., Femiani, J., Hu, J., Wonka, P., Razdan, A.: Curve matching for open 2D curves. Pattern Recogn. Lett. 30, 1-10 (2009)

Examples

```
## Requires specific server architecture  
# mm<-ExtractMotion("C0031D0")
```

h2

Example of Hashes for Image Sample number 2

Description

Example of image hashes as recovered by ExtractionImgHash

Format

A list with four components (dct, str, mw, rd)

Source

TAMIDA 2013

References

TAMIDA 2013

hh

Example of Hashes for Image Sample number 1

Description

Example of image hashes as recovered by ExtractionImgHash

Format

A list with four components (dct, str, mw, rd)

Source

TAMIDA 2013

References

TAMIDA 2013

imagehash1

Image Hash JSON Example

Description

Example of JSON file with image hashes retrieved by ExtractImgHash

Format

JSON

imagehash2

Image Hash JSON Example

Description

Example of JSON file with image hashes retrieved by ExtractImgHash

Format

JSON

imgs

Video Image Frame list as retrieved by ExtractionImgPos

Description

Example of vector of image frames available

Format

A vector with 30 entries, each a string with the position of the available frame

Source

TAMIDA 2013

References

TAMIDA 2013

mm	<i>Video Motion Vector as example</i>
----	---------------------------------------

Description

Example of motion vector for a video

Format

A vector with 249 motion values

Source

TAMIDA 2013

References

TAMIDA 2013

VideoComparison	<i>Calculate the matching segment between two video motion vectors according the the minimum length required.</i>
-----------------	---

Description

For two video segments represented by their own video motion vector it is interesting to look for their matching in order to verify potential coherence between them happens.

By using an improved algorithm based on the Cui et al paper it is implemented the identification of which segment bigger than the requested length, is common to both videos, in terms of motion structure.

Usage

```
VideoComparison(mm,m2,stp,nprocesses=0)
```

Arguments

mm	Motion vector for video 1. It is supposed to be contained in the second one so, it will be shorter than the second vector
m2	Motion vector for video 2. It is supposed to be the container
stp	Unit of motion curvature to be considered as relevant for our analysis.
nprocesses	Number of processes that should be spawned. If 0, it's left to the function's discretion. Requires package parallel to spawn more than 1 process.

Details

This function maximizes the score (Score of matching = length of matching * degree of matching), but returns it separated by its components.

If the package parallel is available, this function will run across several processes.

Value

The output provided is a list with following components:

\$sc	Degree of matching
\$pos1	Frame where matching starts in video1
\$pos2	Frame where matching starts in video2
\$length	Length of matching segment found

Author(s)

Silvia Esinosa, <silviaespinosag@gmail.com>, Jose Maria Perez, <josem.perez.ramos@gmail.com>, Joaquin Ordieres, <j.ordieres@upm.es>, Antonio Bello, <abello@uniovi.es>.

References

Espinosa-Gutierrez, S., Ordieres-Mere, J., Bello-Garcia, A.: Large scale part-to-part video matching by a likelihood function using featured based video representation. TAMIDA 2013 - Taller de Minería de Datos dentro del IV congreso Espannol de Informatica. 254-257 (2013). http://bioinspired.dacya.ucm.es/maeb2013/images/ActasCAEPIA_final.pdf
 Cui, M., Femiani, J., Hu, J., Wonka, P., Razdan, A.: Curve matching for open 2D curves. Pattern Recogn. Lett. 30, 1-10 (2009)

Examples

```
#load example json from data folder
fileName1 = strcat(path.package(package="VideoComparison"),"/data/videomotion1.txt.gz")
json1 = readLines(gzfile(fileName1))
fileName2 = strcat(path.package(package="VideoComparison"),"/data/videomotion2.txt.gz")
json2 = readLines(gzfile(fileName2))

#Extract data from json
hh<- as.numeric(unlist(lapply(fromJSON(json1)$hits$hits[[1]]$`_source`$video_hash$frames,head,1)))
h2<- as.numeric(unlist(lapply(fromJSON(json2)$hits$hits[[1]]$`_source`$video_hash$frames,head,1)))

## Requires specific server architecture
# hh<-ExtractMotion("C0031D0")
# h2<-ExtractMotion("C0035D0")

mm<-VideoComparison(hh,h2,12,nprocesses=1);
```

VideoDistance	<i>Calculate the distances between the two images provided in accordance to its hashes.</i>
---------------	---

Description

Each image es represented by a list of hashes (in current implementation DCT, MW, RD, STRADA) and appropriate metrics is measured, per hash, to both images.

A list of distances is returned

Usage

```
VideoDistance(hh,h2)
```

Arguments

hh List having hash descriptions for the image. Expected components are .dct .str .mw .rd

h2 List having hash descriptions for the image. Expected components are .dct .str .mw .rd

The output is a list with distance components .dct, .str, .mw amd .rd as well as two additional components errlevel and errtext. The first one collects, when different from zero, that some unexpected thing has happened and the explanation will be stored in the second element errtext.

Author(s)

Silvia Esinosa, <silviaespinosag@gmail.com>, Joaquin Ordieres, <j.ordieres@upm.es>, Antonio Bello, <abello@uniovi.es>.

References

Espinosa-Gutiez, S., Ordieres-Mere, J., Bello-Garcia, A.: Large scale part-to-part video matching by a likelihood function using featured based video representation. TAMIDA 2013 - Taller de Minería de Datos dentro del IV congreso Espanol de Informatica. 254-257 (2013). http://bioinspired.dacya.ucm.es/maeb2013/images/ActasCAEPIA_final.pdf

Cui, M., Femiani, J., Hu, J., Wonka, P., Razdan, A.: Curve matching for open 2D curves. Pattern Recogn. Lett. 30, 1-10 (2009)

Examples

```
#Load json data from txt
fileName1 = strcat(path.package(package="VideoComparison"),"/data/imagehash1.txt.gz")
json1 = readLines(gzfile(fileName1))
fileName2 = strcat(path.package(package="VideoComparison"),"/data/imagehash2.txt.gz")
json2 = readLines(gzfile(fileName2))
```

```

# Extract data from json
out1<-fromJSON(json1)$hits$hits[[1]]$`_source`$Hash;
dct1<-as.character(out1[1])
hstrada1<-as.numeric(unlist(strsplit(out1[2],",")))
mw1<-as.numeric(unlist(strsplit(out1[3],",")))
rd1<-as.numeric(unlist(strsplit(out1[4],",")))
img1<-list(dct=dct1,hstrada=hstrada1,mw=mw1,rd=rd1)

out2<-fromJSON(json2)$hits$hits[[1]]$`_source`$Hash;
dct2<-as.character(out2[1])
hstrada2 <-as.numeric(unlist(strsplit(out2[2],",")))
mw2<-as.numeric(unlist(strsplit(out2[3],",")))
rd2<-as.numeric(unlist(strsplit(out2[4],",")))
img2<-list(dct=dct2,hstrada=hstrada2,mw=mw2,rd=rd2)

## Requires specific server architecture
# img1<-ExtractImgHash("000060","C0031D0")
# img2<-ExtractImgHash("000009","C0035D0")

mm<-VideoDistance(img1,img2);

```

VideoMatch

Starting from two list of image hashes and the score in motion matching factor.

Description

VideoMatch looks for the likelihood of coherence between the sequence of pertinent image hashes obtained after preparing the comparison between video Motion vector.

Likelihood estimation relates the coherence (distance: norlized average of four normalized distances) between image sequence from the two videos in the region where their motion curve matches. It can be affected by the factor of coherence of this matching.

Usage

```
VideoMatch(lh1,lh2,sc)
```

Arguments

lh1	List with the ordered sequence on hashes from images belonging to the common motion chunk. It refers to the video1.
lh2	List with the ordered sequence on hashes from images belonging to the common motion chunk. It refers to the video2.
sc	This is the motion scoring factor. It can be derived from the VideoComparison function (first argument of the answer).

Author(s)

Joaquin Ordieres, <j.ordieres@upm.es>, Antonio Bello, <abello@uniovi.es>.

References

Espinosa-Gutierrez, S., Ordieres-Mere, J., Bello-Garcia, A.: Large scale part-to-part video matching by a likelihood function using featured based video representation. TAMIDA 2013 - Taller de Minería de Datos dentro del IV congreso Español de Informática. 254-257 (2013). http://bioinspired.dacya.ucm.es/maeb2013/images/ActasCAEPIA_final.pdf

Cui, M., Femiani, J., Hu, J., Wonka, P., Razdan, A.: Curve matching for open 2D curves. Pattern Recogn. Lett. 30, 1-10 (2009)

Examples

```
#Load json data from txt
fileName1 = strcat(path.package(package="VideoComparison"),"/data/imagehash1.txt.gz")
json1 = readLines(gzfile(fileName1))
fileName2 = strcat(path.package(package="VideoComparison"),"/data/imagehash2.txt.gz")
json2 = readLines(gzfile(fileName2))

# Extract data from json
out1<-fromJSON(json1)$hits$hits[[1]]$`_source`$Hash;
dct1<-as.character(out1[1])
hstrada1<-as.numeric(unlist(strsplit(out1[2],",")))
mw1<-as.numeric(unlist(strsplit(out1[3],",")))
rd1<-as.numeric(unlist(strsplit(out1[4],",")))
img1<-list(dct=dct1,hstrada=hstrada1,mw=mw1,rd=rd1)

out2<-fromJSON(json2)$hits$hits[[1]]$`_source`$Hash;
dct2<-as.character(out2[1])
hstrada2 <-as.numeric(unlist(strsplit(out2[2],",")))
mw2<-as.numeric(unlist(strsplit(out2[3],",")))
rd2<-as.numeric(unlist(strsplit(out2[4],",")))
img2<-list(dct=dct2,hstrada=hstrada2,mw=mw2,rd=rd2)

## Requires specific server architecture
# img1<-ExtractImgHash("000060","C0031D0")
# img2<-ExtractImgHash("000009","C0035D0")

VideoMatch(img1,img2)
```

videomotion1

Video Motion JSON Example

Description

Example of JSON file with a video motion vector retrieved by ExtractMotion

Format

JSON

videomotion2

*Video Motion JSON Example***Description**

Example of JSON file with a video motion vector retrieved by ExtractMotion

Format

JSON

VideoSearch

*Obtain the ID for main video entries into the database.***Description**

Video hashes have been stored in differnt chunks. Main chunk includes the container data. Then, a second chunk named _video.json stores the motion vector and later on, under the ev_father entry different chunks have been stored with key image hashes. Those hashes implements different pHash hashes (DCT, MW, RD, STRADA).

A vector of string is returned

Usage

```
VideoSearch(url)
```

Arguments

url URL where the NonSQL database answer queries.

Author(s)

Silvia Esinosa, <silviaespinosag@gmail.com>, Joaquin Ordieres, <j.ordieres@upm.es>, Antonio Bello, <abello@uniovi.es>.

References

Espinosa-Gutiez, S., Ordieres-Mere, J., Bello-Garcia, A.: Large scale part-to-part video matching by a likelihood function using featured based video representation. TAMIDA 2013 - Taller de Minería de Datos dentro del IV congreso Espannol de Informatica. 254-257 (2013). http://bioinspired.dacya.ucm.es/maeb2013/images/ActasCAEPIA_final.pdf

Cui, M., Femiani, J., Hu, J., Wonka, P., Razdan, A.: Curve matching for open 2D curves. Pattern Recogn. Lett. 30, 1-10 (2009)

Examples

```
## Requires specific server architecture  
# videos<-VideoSearch()
```

Index

- *Topic **Image Hash**
 - [ExtractImgHash](#), 4
 - [VideoDistance](#), 11
 - *Topic **JSON**
 - [ExtractImgHash](#), 4
 - [ExtractImgPos](#), 5
 - [ExtractMotion](#), 6
 - *Topic **NoSQL**
 - [Compare2Videos](#), 3
 - [ExtractImgHash](#), 4
 - [ExtractImgPos](#), 5
 - [ExtractMotion](#), 6
 - [VideoMatch](#), 12
 - *Topic **Video Likelihood**
 - [Compare2Videos](#), 3
 - *Topic **Video List**
 - [VideoSearch](#), 14
 - *Topic **Video Motion Curve**
 - [ExtractImgPos](#), 5
 - [ExtractMotion](#), 6
 - [VideoMatch](#), 12
 - *Topic **Video Motion**
 - [VideoComparison](#), 9
 - *Topic **datasets**
 - [h2](#), 7
 - [hh](#), 7
 - [imagehash1](#), 8
 - [imagehash2](#), 8
 - [imgs](#), 8
 - [mm](#), 9
 - [videomotion1](#), 13
 - [videomotion2](#), 14
 - *Topic **package**
 - [VideoComparison-package](#), 2
- [Compare2Videos](#), 3
- [ExtractImgHash](#), 4
- [ExtractImgPos](#), 5
- [ExtractMotion](#), 6
- [h2](#), 7
- [hh](#), 7
- [imagehash1](#), 8
- [imagehash2](#), 8
- [imgs](#), 8
- [mm](#), 9
- [VideoCompare-package](#)
([VideoComparison-package](#)), 2
- [VideoComparison](#), 9
- [VideoComparison-package](#), 2
- [VideoDistance](#), 11
- [VideoMatch](#), 12
- [videomotion1](#), 13
- [videomotion2](#), 14
- [VideoSearch](#), 14