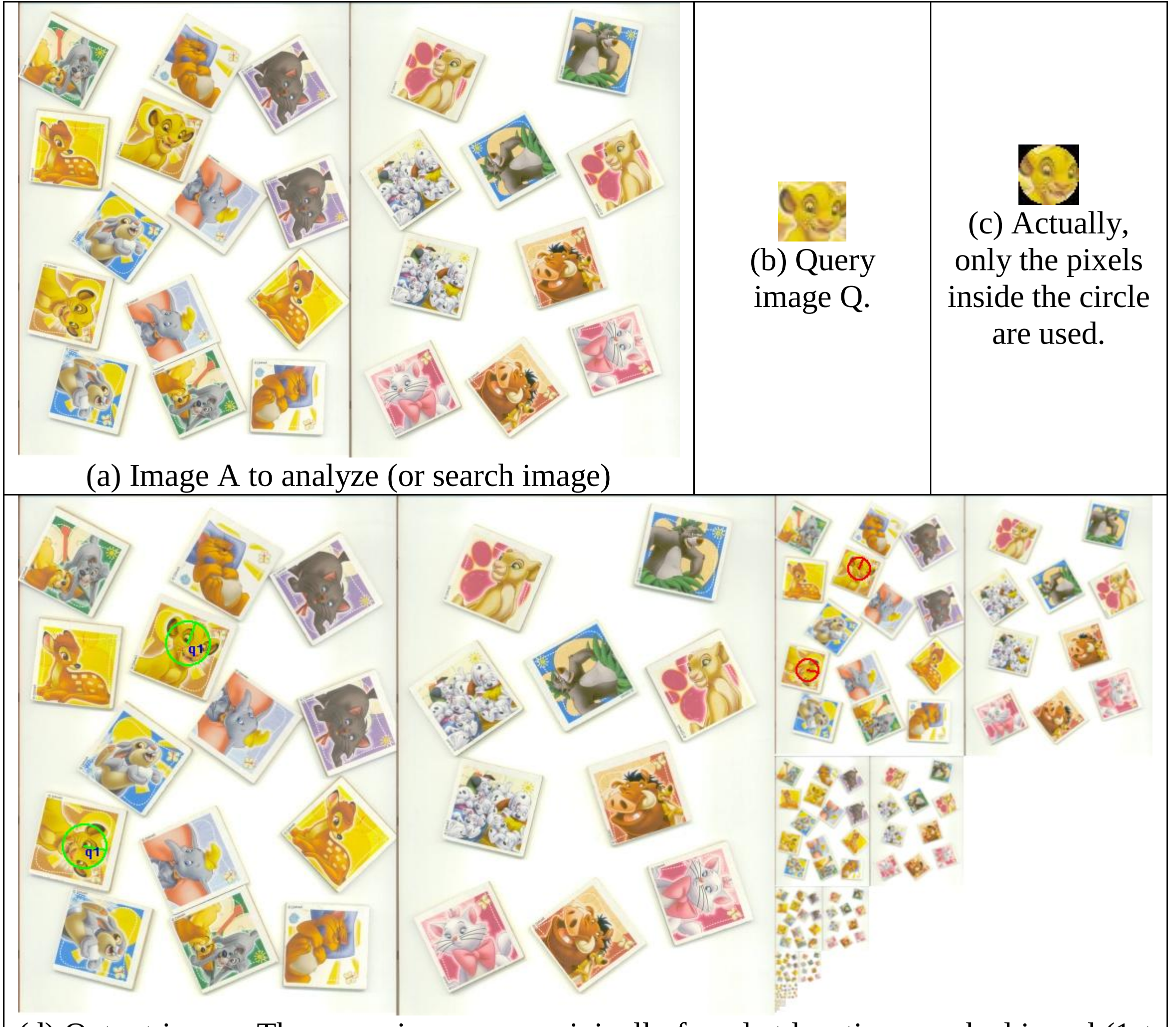
Ciratefi demonstration programs

Last revision: April 2, 2014

1 Introduction

File Ciratefi.zip contains improved pyramidal Ciratefi demonstration programs and example image files. Ciratefi (Circular, Radial and Template-Matching Filter) is a template matching technique invariant to rotation, scale, translation, brightness and contrast. Ciratefi was defined in papers [Ci22, Ri13].

An example of Ciratefi input/output is depicted in Fig. 1.



(a) Image A to analyze (or search image)

(b) Query image Q.

(c) Actually, only the pixels inside the circle are used.

(d) Output image. The query image was originally found at locations marked in red (1st floor of the pyramid). The locations in green are translations of these locations to the 0th floor of the pyramid.

```
row     col  scale angDeg struct  color  simil rowPyr colPyr scaleP floorP
149     171    1.1  340.0  0.957  0.969  0.965     74    846    0.6      1
353      67    1.1  260.0  0.937  0.976  0.957    176    794    0.6      1
```

(e) The output text file with information about the matches.

Figure 1: An example of Ciratefi input-output files.

2 Installation

The programs require Windows XP, Vista or 7. Probably, they will run correctly also in Windows 8.

1) Uncompress ciratefi.zip in a directory, say, **c:\ciratefi**, keeping the subdirectory structure.

2) Edit "path" environment variable to append **c:\ciratefi\bin**. In Windows7, you can do this:
MS-explorer → my computer → properties → advanced → environment variables
where you must edit path to *append* **c:\ciratefi\bin**.

3 Installation test

Open a command prompt. In some directory, write *cirategs* or *ciratecs*:

```
c:\directory>cirategs
```

The following prompt must appear:

```
CirateGS: Piramidal ciratefi for grayscale images v1.05
CirateGS cirategs.cfg [a.pgm q.pgm ci.ppm]
  If [a.pgm q.pgm ci.pgm] are specified, overrides cirategs.cfg
Error: Invalid number of arguments
```

Note: If the program does not run, install the following VC-2005 redistributables:
vcredist_x86.exe
The same program can be downloaded directly from the Microsoft's site:
http://www.microsoft.com/downloads/details.aspx?familyid=200B2FD9-AE1A-4A14-984D-389C36F85647&displaylang=en

4 Running the examples

4.1 Running an example

Go to c:\ciratefi\1instance and run:

```
c:\ciratefi\1instance>ciratecs ciratecs-1instance.cfg a2.jpg q01.ppm c201.ppm
```

This command will search for `q01.ppm` in image `a2.jpg` and write the output in `c201.ppm`.

4.2 Intermediary files

Many intermediary files are generated:

```
ga.tga: Gaussian filtered search image (a2.jpg)
gq.tga: Gaussian filtered query image (q01.ppm)
actualq.tga: Only the pixels inside the circle are used.
a.avi: Circular (ring) projections of ga.tga
cq.tga: Circular (ring) projections of gq.tga
cp.tga: Cifi's output and Rafi's input.
        1st degree candidate pixels in color.
        Embedded scale parameter.
cm.tga: Cifi's output and Rafi's input.
        1st degree candidate pixels in color.
        Embedded Cifi's color similarity measure.
rq.tga: Radial projection of gq.tga.
rp.tga: Rafi's output and Tefi's input.
        2nd degree candidate pixels in color.
        Embedded scale and rotation parameters.
rm.tga: Rafi's output and Tefi's input.
        2nd degree candidate pixels in color.
        Embedded Rafi's color similarity measure.
tp.tga: Tefi's output.
        Matching pixels in color.
        Embedded scale and rotation parameters.
tm.tga: Tefi's output.
        Matching pixels in color.
        Embedded color similarity measure.
c201.ppm: The output image with circle and pointer.
p.txt: The output text file.
```

4.3 Configuration file

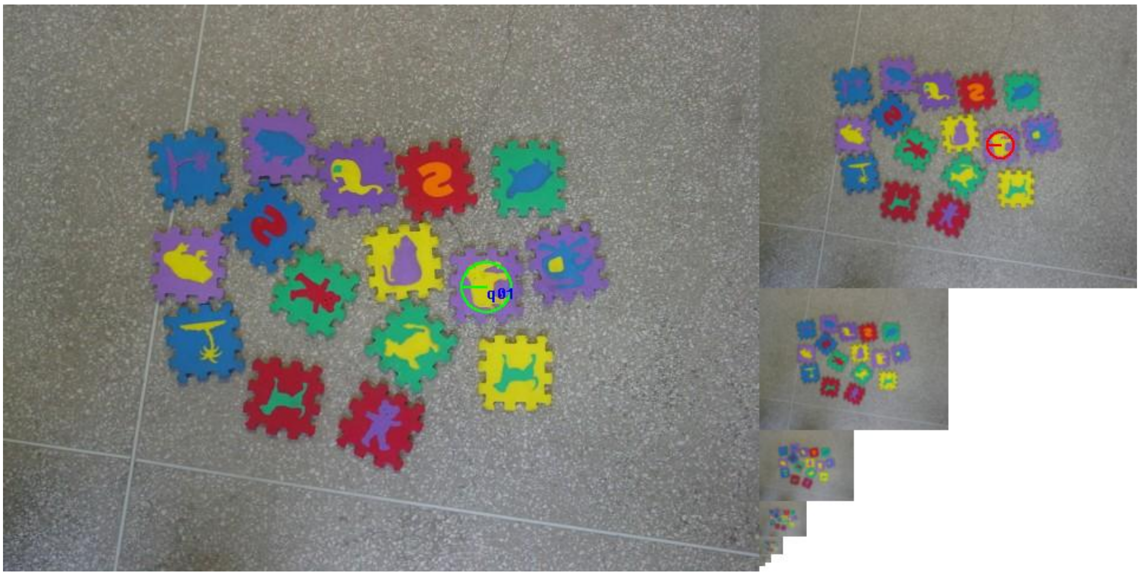
You can change Ciratefi parameters by editing `ciratecs-1instance.cfg`. The parameters the user should know to fine-tune the search are:

```
pct_cand_1f=2.0
// Percentage of first grade candidates in relation to the total number of
pixels of A. If you increase this parameter, the program will run slower, but
the probability of missing the template decreases. If you want to find more
than one instance of the query image, you should increase this parameter.

pct_cand_2f=1.0
// Percentage of second grade candidates in relation to the total number of
pixels of A. If you increase this parameter, the program will run slower, but
the probability of missing the template decreases. If you want to find more
than one instance of the query image, you should increase this parameter.

qtd_cand_3f=1
// The number of objects to be detected. For example, if qtd_cand_3f=4, the
program will return 4 locations in search image A most similar to the query
image Q.

dist_pixel_3f=0
// The minimal distance between two matching pixels. For example, if
dist_pixel_3f=20, there cannot be two matchings separated by less than 20
pixels. For example, if qtd_cand_3f=2 and dist_pixel_3f=0, the program can
return two neighboring pixels as the two matchings.

ssalpha=0.01 // Weight of brightness.
ssbeta=0.01  // Weight of contrast. Do not set to zero.
ssgama=0.49  // Weight of "structure" or correlation
ssdelta=0.49 // Weight of color or chromaticity
```

```
// You can change the weights assigned to brightness, contrast, structure and
color differences.
```

4.4 *Output files*

The program marks in red the location where the template was found in the output image. If the template was not found on the 0-th floor of the scale pyramid, then the program marks in green the corresponding location on the 0-th floor.



The output text file is p.txt (unless you have changed its name in the configuration file):

```
   row    col  scale angDeg struct  color  simil rowPyr colPyr scaleP floorP
   239    409    1.0   90.0  0.961  0.945  0.957    119    844    0.5      1
```

Row, *column*, *scale* and *angle_degrees* returns the localization of the template on the 0-th floor of the scale pyramid. *Angle_degrees* measures the angle starting at 12 hour and rotates counterclockwise.

Struct, *color* and *similarity* returns the similarity measure. They all ranges from 0 to 1. *Struct* is the normalized correlation of the grayscale image. *Color* is the similarity of coloration. *Similarity* is a ponderation between struct, color, brightness and contrast (the last two measures are not written in the output file).

Row_pyramid, *column_pyramid*, *scale_pyramid* and *floor_pyramid* are the location of the matching on the floor>0 of the scale pyramid. This is the original location where the template was found.

4.5 Running the batches

You can run the tests by calling the batch files:
C:\ciratefi\1instance>runc (color tests)
C:\ciratefi\1instance>rung (grayscale tests)
C:\ciratefi\2instances>runc (color tests)
C:\ciratefi\2instances>rung (grayscale tests)

Note: Converting the query image c:\ciratefi\1instance\q14.ppm to grayscale, it becomes an image with almost constante grayscale. So, Cirategs fails to find this query image.

5 Recompiling the source programs

1) Install Proeikon library, dev-cpp compiler and run setproeikon.bat as described in:
http://www.lps.usp.br/hae/software/proeikon.html

2) The following commands should recompile the programs

```
c:\ciratefi\src>cpv cirategs
c:\ciratefi\src>cpv ciratecs
```

Note: The source code of Proeikon library is not available.

6 References

[Ci22] H. Y. Kim and S. A. Araújo, "Grayscale Template-Matching Invariant to Rotation, Scale, Translation, Brightness and Contrast," *IEEE Pacific-Rim Symposium on Image and Video Technology, Lecture Notes in Computer Science*, vol. 4872, pp. 100-113, 2007.

[Ri13] S. A. Araújo and H. Y. Kim, “Ciratefi: An RST-Invariant Template Matching with Extension to Color Images,” *Integrated Computer-Aided Engineering*, vol. 18, no. 1, pp. 75-90, 2011.