***Pascal Matsakis* Dec 11, 2009**

University of Guelph

pmatsaki@uoguelph.ca

***JingBo Ni***

University of Guelph

jni@uoguelph.ca

***Let us know if you run into any bugs. Please do not pass this software package to others outside your research team: ask them to contact us and we will be happy to send them the latest version of the package.***

The concept of the force histogram was introduced by Matsakis in [1]. This package contains four different implementations of the force histogram in the case of 2D raster data:

***(i)*** The original implementation by Wendling and Matsakis. It is based on the partitioning of the image into parallel raster lines. Force histograms are computed in O(KN√N) time for concave object and in O(KN) time for convex objects, where K is the number of reference directions and N is the number of pixels in the image [2]. ***Please cite [1] and [2] in any work using this implementation.***

***(ii)*** and ***(iii)*** The implementations by Ni of the two algorithms developed by Wang et al. [3]. Both algorithms are dedicated to the computation of constant force histograms. One is in O(KN) time, the other is in O(NlogN) time. ***Please cite [1] and [3] in any work using these implementations.***

***(iv)*** The implementation by Ni of the algorithm developed by Ni and Matsakis [4]. It is based on the computation, in O(NlogN) time, of a mapping defined over the 2D discrete vector space. A force histogram can be derived from this mapping in O(K√N) time. ***Please cite [1] and [4] in any work using this implementation.***

**References**

[1] P. Matsakis, *Relations spatiales structurelles et interprétation d’images*,

PhD. Thesis, Institut de Recherche en Informatique de Toulouse, France, 1998.

[2] P. Matsakis, L. Wendling, "A new way to represent the relative position of areal objects",

*IEEE Trans. on Pattern Analysis and Machine Intelligence*, 21(7):634-643, 1999.

[3] Y. Wang, F. Makedon, R.L. Drysdale, "Fast algorithms to compute the force histogram",

2004, unpublished.

[4] J. Ni, P. Matsakis, "An equivalent definition of the histogram of forces:

theoretical and algorithmic implications", *Pattern Recognition*, in press.

**Readme File**

Written by JingBo Ni

**1. Files**

*util.h and util.c:*

define and implement a number of utility functions.

*imageio.h and imageio.c:*

define the data structure for representing a PGM image; implement the imageio functions; and other functions related to processing PGM images.

*objfunc.h and objfunc.c:*

define the data structure for representing a raster object; and a number of functions for processing a raster object including resize, extension, creating, and reflection etc.

*fourier.h and fourier.c:*

implement the 2D fast fourier transformation and the 2D digital convolution.

Although there are many existing implementations of FFT, we implemented it

to include some special optimizations.

*forcehistogram.h and forcehistogram.c:*

implement the original algorithm [1] of force histogram computation using both the double scheme [5] and the single sum scheme [6].

*framework.h and framework.c:*

implement the algorithm according to the new framework [3][4]. It also provides an implementation of the angle histogram [6][7] using the new framework.

*forcehistogram\_w.h and forcehistogram\_w.c:*

implement the two algorithms proposed by Wang [2]. These algorithms are only applicable to constant force histograms.

*A\_1.pgm and B\_1.pgm:*

a pair of intersecting crisp objects with complex shapes.

*A\_2.pgm and B\_2.pgm:*

a pair of disjoint crisp objects with simple shapes.

*makefile:*

for compiling under linux.

**2. How to compile**

In Linux, simply type "make". It compiles into three executable programs:

fh\_std: the original algorithm.

fh\_new: the new framework.

fh\_wang: the wang's algorithms.

**3. How to run**

***For the original algorithm, you can run it by:***

"./fh\_std object\_A object\_B directions r double\_or\_single histogram\_file"

for example:

"./fh\_std A\_1.pgm B\_1.pgm 360 0.5 0 his.txt"

object\_A and object\_B: are respectively the PGM images containing object A (reference object) and object B (argument object).

Here, in a PGM image, if the object is a crisp object, then pixels belonging to the object are in white and others are in black.

If the object is a fuzzy object, the grey level of a pixel indicates the membership degree that the pixel belongs to the object. Brighter colors indicate higher membership degrees.

The range of the grey levels of a PGM image is from 0 to 255.

directions (d): is an integer indicating the number of reference directions considered.

r: real value indicating the type of forces considered, r=0 is the constant forces and r=2 is the gravitational forces.

double\_or\_single: 0 indicates computing force histogram using the double sum scheme, 1 indicates using the single sum scheme. For crisp objects, you will get exactly the same histogram no matter you use the double or the single sum scheme.

histogram\_file: the txt file stores the generated histogram. The first line in the txt file gives the number of reference directions considered. The follows are the corresponding force values from direction 0 to direction 2pi.

***For the algorithm using the new framework, you can run it by:***

"./fh\_new object\_A object\_B directions r histogram\_file"

for example:

"./fh\_std A\_1.pgm B\_1.pgm 360 0.5 his.txt"

The parameters are the same as those mentioned above.

***For the Wang's algorithms, you can run them by:***

"./fh\_new object\_A object\_B directions first\_or\_second histogram\_file"

for example

"./fh\_std A\_1.pgm B\_1.pgm 360 1 his.txt"

(using the first algorithm with complexity O(KN));

or "./fh\_std A\_1.pgm B\_1.pgm 360 2 his.txt"

(using the second algorithm with complexity O(NlogN)).

Remember Wang's algorithms only compute constant (r=0) force histograms, so here users are not allowed to select the types of forces.

first\_or\_second: 1: the first algorithm; 2: the second algorithm.

The other parameters are the same as those mentioned above.

**References**

[1] P. Matsakis, L. Wendling. "A new way to represent the relative position of areal objects",

*IEEE Trans. on Pattern Analysis and Machine Intelligence*, 21(7):634-643, 1999.

[2] Y. Wang, F. Makedon, R.L. Drysdale, "Fast algorithms to compute the force histogram",

2004, unpublished.

[3] J. Ni, P. Matsakis, "Force histograms computed in O(NlogN)", *Proceedings of the 19th IAPR Int. Conf. on Pattern Recognition*, 1-4, 2008.

[4] J. Ni, P. Matsakis, "An equivalent definition of the histogram of forces:

theoretical and algorithmic implications", *Pattern Recognition*, in press.

[5] D. Dubois, M.C. Jaulent, "A general approach to parameter evaluation in fuzzy digital pictures", *Pattern Recognition Letters*, 6(4):251-259, 1987.

[6] R. Krishnapuram, J.M. Keller, Y. Ma, "Quantitative analysis of properties and spatial relations of fuzzy image regions", *IEEE Trans. on Fuzzy Systems*, 1(3):222-233, 1993.

[7] K. Miyajima, A. Ralescu, "Spatial organization in 2D segmented images: representation and recognition of primitive spatial relations", *Fuzzy Sets and Systems*, 65(2-3):225-236, 1994.