

# Introduction

**Two-dimensional Digital Image Correlation (2D-DIC)** is an accurate and high-efficiency optical method for motion and deformation measurement of planar sample subjected to in-plane deformation.

This is a MATLAB® Toolbox for the **inverse compositional Levenberg–Marquardt ( IC-LM )** algorithm combining reliability-guided displacement tracking strategy. As a comparison, the inverse compositional Gaussian-Newton ( IC-GN ) algorithm is also involved. You can choose different shape function, including first-order and second-order shape function. The **major advantage** of the IC-LM algorithm is that it has **obviously larger converge radius** while holds **almost the same accuracy and efficiency** compared with the classical IC-GN algorithm.

This MATLAB® toolbox of IC-LM/IC-LM2 algorithm involves some of the advanced features of DIC algorithm. We have tried to make the codes modularized and readable. Hope it could reveal the detail inside DIC for you and contribute your project.

## Usage

1. Download the project and unzip it into a folder;
2. Before running, you need to set some **key parameters in paramset.m** , such as **subset size**, **step size**, **iteration method**, whether **normalization** is used, **threshold of iteration number**, etc
3. Run `demo.m` . In the first pop-up window select one reference image, and in the second pop-up window select a deformed image or a set of deformed images.
4. The bicubic B-spline interpolation C++ MEX file has already been created. However, if you have changed the source code, please create it again.

### On MacOS platform

- Install Xcode on your mac;
- Type the following command in MATLAB command window;  
`mex -v -setup C; mex -v -setup C++;`
- Edit and save your C++ code in file `BicubicBsplineInterp.cpp` ;
- Copy the former file to your work directory, and then type the following command in the command window:  
`mex BicubicBsplineInterp.cpp`
- You will get a file `BicubicBsplineInterp.mexmaci64` in your work directory. Finally you can use it like regular MATLAB function.

### On Windows platform

- Install C and C++ compiler. Here the free MinGW-w64 C/C++ compiler is recommended. Please check the compatible version from [MATLAB support](#). We have confirmed that MATLAB R2019a is compatible with [MinGW-w64 V6.3.0](#) on windows 10.
  - Type the following command in MATLAB command window;  
`mex -v -setup C; mex -v -setup C++;`
  - Edit and save your C++ code in file `BicubicBsplineInterp.cpp` ;
  - Copy the former file to your work directory, and then type the following command in the command window:  
`mex BicubicBsplineInterp.cpp`
  - You will get a file `BicubicBsplineInterp.mexmaci64` in your work directory. Finally you can use it like regular MATLAB function.
5. On the reference image, draw a rectangle region of interest (ROI) by specifying the upper left corner and lower right corner in sequence.
  6. Select a seed point on the reference image, then just wait.
  7. The results are saved in the same folder with the same file name but format of `.mat` ;
  8. Note that you can easily parallelize the algorithm by changing the `for...end` to `parfor...end` loop in the `demo.m` file. Each thread will match one deformed image.
  9. If you need to evaluate the performance of the method, you can use some [open-source code](#) to simulate some desired deformation images.
  10. Enjoy.

## Key features

- **Gaussian pre-filter** is added to reduce sub-pixel registration bias error of DIC;
- **Coarse-to-fine**, **FFT** and **Manually selection** are optional ways for initial displacement guess of a seed point;
- **Tricubic B-spline interpolation** scheme is used to get the gray intensity on the deformed image at sub-pixel location. MATLAB® and C++ implementations of the interpolation are optional in the source code. The default is the C++ version, and you can choose also MATLAB® version.
- **Reliability-guided displacement tracking strategy** is used for fast, robust and accurate point registration. Of course you are not necessarily uses it, but use that without reliability-guided displacement tracking strategy;
- **Inverse compositional Levenberg–Marquart algorithm** is adopted for larger converge radius while keeping the same accuracy and efficiency.
- **First-order** and **Second-order** shape functions are used to describe the deformation of each subset, and you can choose from them accordingly.
- **Normalization** is used to initialize a good damping parameter  $\lambda$ ;
- **Pointwise least-squares** is used to estimate strain distribution.

## Explanation of functions

Functions	Usage
demo.m	The entrance of the whole program. Start from here.
paramset.m	Set all parameters required for this project.
DIC_main.m	Most of the DIC operation are called in this function, including the B-spline filter of the images, the initial guess of the seed point and the matching of the two images.
DICmatch.m	Match the point grid sequentially.
DICmatch_RG.m	Match the point grid with reliability-guided displacement tracking strategy.
gradImg.m	Estimate the gradient of an image with covolution.
plotOnImg.m	Plot the contour figure of an interested result on reference image. Select the last input among {u, v, exx, eyy} to plot the needed result.
Warp.m	Transfer a parameter vector <b>p</b> or parameter increment vector $\Delta \mathbf{p}$ to a warp matrix.
findInitP.m	Estimate the initial displacement guess of the seed point.
corrIter.m	Choose a required iteration method, including IC-GN , IC-GN-RG , IC-LM , IC-LM-RG , IC-GN2 , IC-GN2-RG , IC-LM2 and IC-LM2-RG .
iterICGN.m	The inverse-compositional Gaussian-Newton algorithm ( IC-GN ) algorithm.
iterICLM.m	The inverse-compositional Levenberg-Marquardt algorithm ( IC-LM ) algorithm.
iterICGN2.m	The inverse-compositional Gaussian-Newton algorithm ( IC-GN2 ) algorithm with second-order shape function.
iterICLM2.m	The inverse-compositional Levenberg-Marquardt algorithm ( IC-LM2 ) algorithm with second-order shape function.
calHessian.m	Calculate the Hessian matrix ( $6 \times 6$ ) for first order shape function.
calHessian2.m	Calculate the Hessian matrix ( $12 \times 12$ ) for second order shape function.
BsplineFilter.m	Filter implemented prior to interpolation for bicubic B-spline image interpolation.
bicubicBsplineInterp.m	Compute the gray intensity of the points at sub-pixel locations in the target subset. It is implemented with MATLAB® .
BicubicBsplineInterp.mexmaci64	Compute the gray intensity of the points at sub-pixel locations in the target subset. It is implemented with C++, so it is faster.
calcuPt.m	Select a grid of calculation points.

## Explanation of key parameters

Parameters	Explanation
disp	displacement field with size of $N \times 2$ .
iterNum	the iteration number of each point, a $N$ -dimensional vector
strain	the strain fields with size of $N \times 3$ , and the three columns correpond to exx , eyy and exy , respectively.
ZNCC	the ZNCC coefficient, a $N$ -dimensional vector
Params	A structure comprising some important parameters used in DIC algorithm.
lambda	$\lambda$ , the damping parameter used in Levenberg-Marquardt algorithm, see Eq. (6).
half_subset	Half subset size, which is equal to $K$ .
subset	The subset size, which is equal to $2K+1$ .

Parameters	Explanation
Step	The grid step for selecting the calculation point grid.
strainWin	The window size for calculating strain using pointwise least-squares, and it must be an odd number
IterMethod	Gives the iteration method, and its value could be IC-GN , IC-GN-RG , IC-LM , IC-LM-RG , IC-GN2 , IC-GN2-RG , IC-LM2 or IC-LM2-RG .
Normalization	Its value could either be 1 or 0. If 1, normalization is used, vice versa.
maxIter	The maximum iteration number for matching each point, and its default value is 15. If the iteration number exceeds this value, the point is considered to be wrongly matched.
thre	The threhold for displacement increment, and its default value is 0.001. If the displacement increment at a given iteration is lower than this value, this point is matched accomponied by the finish of this iteration.
fixed_seedPts	Its value is selected between 1 and 0. If 0, you will select region of interest and seed point manually. Otherwise, the algorithm will do it automatically, and it could be useful for algorithm development.
InitMethod	Method for the initial guess of the displacement at seed point, it could be chosen among 0, 1, 2, 3. 0 means directly give intial displacement of (0,0); 1 denotes corse-to-fine algorithm; 2 denotes Fast Fourier Transformation; 3 is manually selection
localSub	The local coordinate of the points in reference subset, with size of $(2K+1)(2K+1) \times 2$
localSubHom	The homogeneous local coordinate of the points in reference subset, with size of $3 \times (2K+1)(2K+1)$ .
InitDispP	The initial guess of the displacement at the seed point.
gradxImR	The intensity gradient of reference image along x direction
gradyImR	The intensity gradient of reference image along y direction
H	Hessian matrix.
invH	The inverse of Hessian matrix.
Jacob	Jacobian = $\nabla f \frac{\partial W}{\partial p}$
deltafVec	$f - \bar{f}$
deltaf	$\Delta f = \sqrt{(f - \bar{f})^2}$
deltagVec	$g - \bar{g}$
deltag	$\Delta g = \sqrt{(g - \bar{g})^2}$
gIntep	The sub-pixel displacement of the correponding point in a target subset
PcoordInt	The sub-pixel coordinate (global coordinate) of the correponding point in a target subset.
comptPoints	The coordinates of the selected calcualtion points on reference image.
defIntp	The interpolated gray intensity on deformed image at sub-pixel locations, and it is a $(2K+1)(2K+1) \times 1$ vector
p	The parameter vector.
deltap	The incremental parameter vector.
M	The diagonal matrix for coordinate normalization.
exx, eyy, exy	exx and eyy are strain along x, y direction, respectively, while exy is the shear strain. Note that their unit are $\mu\epsilon$ .

## Questions & Suggestions

If you wish to contribute code/algorithms to this project, or have any question or suggestion, please contact Bin Chen via [binchen@kth.se](mailto:binchen@kth.se).

## Citation

**Anyone who use the code please cite:**

"Bin Chen., Erik Jungstedt.,

Fast, robust, accurate and large-converge-radius inverse compositional Levenberg-Marquardt algorithm for digital image correlation: preprint  
.(under review)"