begindocument/before

Xylem properties and the dynamics of embolism propagation in plants

Florencia Falkinhoff
LPENS and
IFPEN[*]

Alexandre Ponomarenko, Mica, and Romain
LPENS

Jean-Lou Pierson and Lionel Gamet
IFPEN[*]
(Dated: August 14, 2021, version 01)

I. DETAILS FOR THE EXPERIEMENT SHOWN IN THE ABSTRACT

frames per second: 780fps.
exposure time: 1282 μs

II. INTRODUCTION

introduction introduction introduction

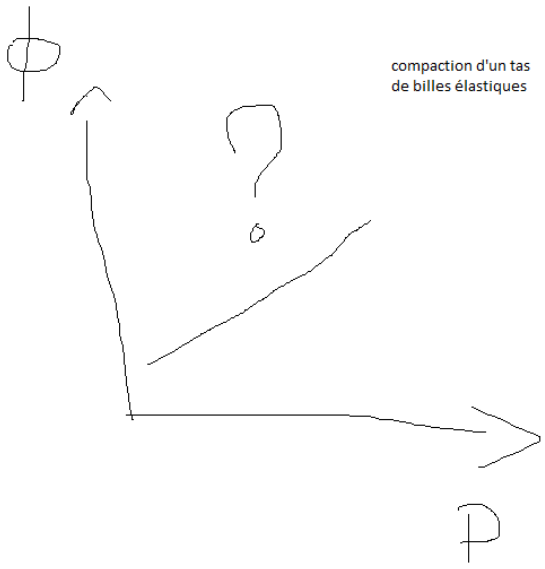

FIG. 1: Figure idea about the volumic fraction of an elastic stack of beads.

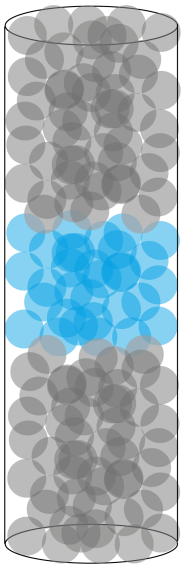

FIG. 2: beads

A. Discussion Vincent

fraction volumique tas de billes molles.?

B. Notes pour discussion avec Romain Volk

- achat billes pour remplir la colonne.
- Analyse 3D.
- Analyse PTV.
- Qualité caméra VS qualité téléphone portable?

$$p^* \approx \frac{2\gamma}{r^*}. \tag{1}$$

*Electronic address: florencia.falkinhoff@ens-lyon.fr

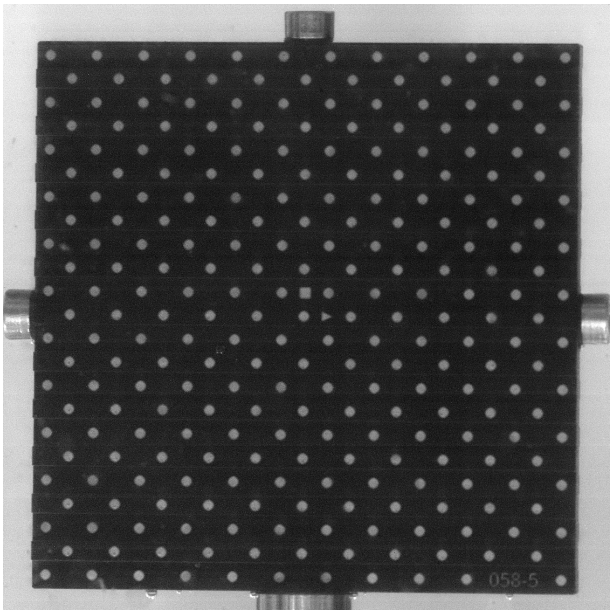

FIG. 3: calibration target

III. TODO LIST

- gcc compilateur c++ - mpi for parallelisation

A. Calibration with two level target

Adapt David Dumond 4DPTV program to the new scale target.

Remarks: for the calculation of the spatial transformation from image points (pimg) to 2D postition in real space (pos2D), the code uses the function $T3rw2px = fitgeotrans(pimg, pos2D,' polynomial', 3);$, if there is not enough points it reports : " Error using images.geotrans.PolynomialTransformation2D (line 162) At least 10 non-collinear points needed to infer polynomial transform. " . To deal with that, a possibility is to set the polynomial degree to 2 instead of 3.

Depending on the face you look at, the square is on a up line or on a down line. On figure (**??**), the square is on an UP line.

The calibration target has 23 lines. The DOWN lines have 12 circles. The UP lines have 11 circles. There are two aditional elements: a square on line 11 and a triangle on line 12. There is a total of 143 circles on DOWN lines and 121 circles on UP lines plus a square on line 11 and a triangle on line 12.

Worflow for doing the calibration with this special target: 20°C or 44.9°C 44.9 °C

IV. MATERIAL AND METHODS

A. 4D PTV

The code is on github 4D-ptv on git and the documentation on read the docs.

I install VisualStudioCode, and add C++ tools following this tutorial Visual Studio Code C++. I can compile C++ code.

Installing gcc. I used tips from here: link but I downloaded MinGCW from download MinGW as indicated on VSCode infos: VSCode MinGW

Change drive in command prompt. To go to D: cd /d D: Instead of 'make', use 'mingw32-make'

Install hf5++ , good guide here and maybe also here and here

1. PSMN

./STM -i "./test/rays.dat" -o "./test/" -f 10 -c 2 -d 0.2 -s 1 -m 2 -x 400 -y 400 -z 400 -b -3 3 -3 3 1 5 –hdf5 ¿¿ rays.log

B. Sketch plugging the two cameras together

Shut off all Windows fire walls
terminal:
ping 100.100.111.52
ping 100.100.118.227

C. index matching beads

1. Hydrogel beads

There are three hydrogel beads names 1, 2 and 3. Order of preference: 1, 3, 2.

About hydrogel beads 2. They come in pinky pockets named Jelly-Beads, containing 5.1g of beads that means $\sim$ 300 beads

D. Flow Meter

E. solenoid valve: Burkert

F. tunings for electrovannes

note from 2021 01 15
electrovanne
ancien réglage (DARCY 01):
low 564
high 665

G. 3D printer

H. HDR image from bracketing exposure time image sequence

https://docs.opencv.org/master/d3/db7/tutorial$_h dr_i maging.html$
https://towardsdatascience.com/hdr-imaging-what-is-an-hdr-image-anyway-bdf05985492c
https://www.dpreview.com/articles/9828658229/computational-photography-part-i-what-is-computational-photography/2

I. bracketing time lapse

List of links to took pictures with the camera: Digicam control command lines: http://digicamcontrol.com/doc/userguide/cmd Info taken from here https://stackoverflow.com/questions/43358257/using-digicamcontrol-to-control-nikon-camera-using-python and here http://www.pauldebevec.com/Research/HDR/ and here https://fr.mathworks.com/matlabcentral/fileexchange/57196-cameracontroller?s$_t id = FX_r c3_b ehav$

Some information for processing .NEF pictures: A comibnation of matrawread (from https://github.com/QiuJueqin/MatRaw) and dcraw.exe from https://www.dechifro.org/dcraw/ & https://www.fastpictureviewer.com/downloads/links
Use of dcraw: https://www.programmersought.com/article/46784093292/
https://www.cnba.it/contenuti/uploads/2016/03/Processing-RAW-Images-in-MATLAB-Sumner.pdf
finally, dcraw.exe got here: https://fr.osdn.net/projects/sfnet$_d crawnet/downloads/dcraw.exe/$
From image analysis boss from matlab: https://blogs.mathworks.com/steve/2011/03/08/tips-for-reading-a-camera-raw-file-into-matlab/

1. HDR from serie of .NEF

link 01
link 02
link 03
link 04

J. mirrorless ?

https://www.dxomark.com/things-are-heating-up-in-the-full-frame-mirrorless-camera-market/
https://www.jmpeltier.com/disadvantages-of-mirrorless-cameras/
not happy with a7: https://fstoppers.com/originals/i-wish-id-known-i-moved-sony-366521

K. negative shutter time on cameras ?

https://photodoto.com/here-is-why-mirrorless-cameras-have-shutters/

V. LIST OF THE EXPERIMENTS

A. experiment 2021 05 28

We make a calibration in air.
We record an image sequence at 50Hz of a black point drawn on a white paper which is moved in 3D. The dot position in 2D on each camera is tracked with FIJI (smooth → threshold → analyse particles). The IMAGEJ points are saved in Matlab variables 'Camera0_FIJI' and 'Camera1_FIJI'.
Then rays are crossed and it gives points in 3D. As seen on the figures.

VI. PRELIMINARY RESULTS

VII. FIGURES SECTION FROM KAARE