SIGGRAPH THINK BEYOND
2020 S2020.SIGGRAPH.ORG

CODE REPLICABILITY IN COMPUTER GRAPHICS

Nicolas Bonneel[1], David Coeurjolly[1],
Julie Digne[1], Nicolas Mellado[2]

[1]Univ. Lyon./CNRS [2]Univ. Toulouse./CNRS

MOTIVATION

- *How to compare our method to others if no code available ?*

- *How to improve other methods if no code available ?*

- *How can I trust others' results if no code available ?*
  - If my reimplementation produces different results, do I have a bug ?
  - Examples:
    - Excel date conversion in genomic (e.g., Septin 2 -> SEPT2 -> 09/02) [Ziemann et al. 2016]. Now SEPTIN2!
    - Bug in fMRI software increased false-positive rates [Eklund et al. 2016]

GOALS

- Evaluate replicability of codes in computer graphics
  - ACM definition: Replicability = using the author’s codes / Reproducibility = re-coding.
    - Definition highly debated ! Opposite definition for National Academies of Sciences, Engineering and Medicine.
  - Siggraph 2014, 2016, 2018 conferences as proxy with high quality standards
  - Spirit: as if I asked my Ph.D. students to compare her approach to a given Siggraph paper
    - Not just “I tried to compile 10min, it didn’t work”

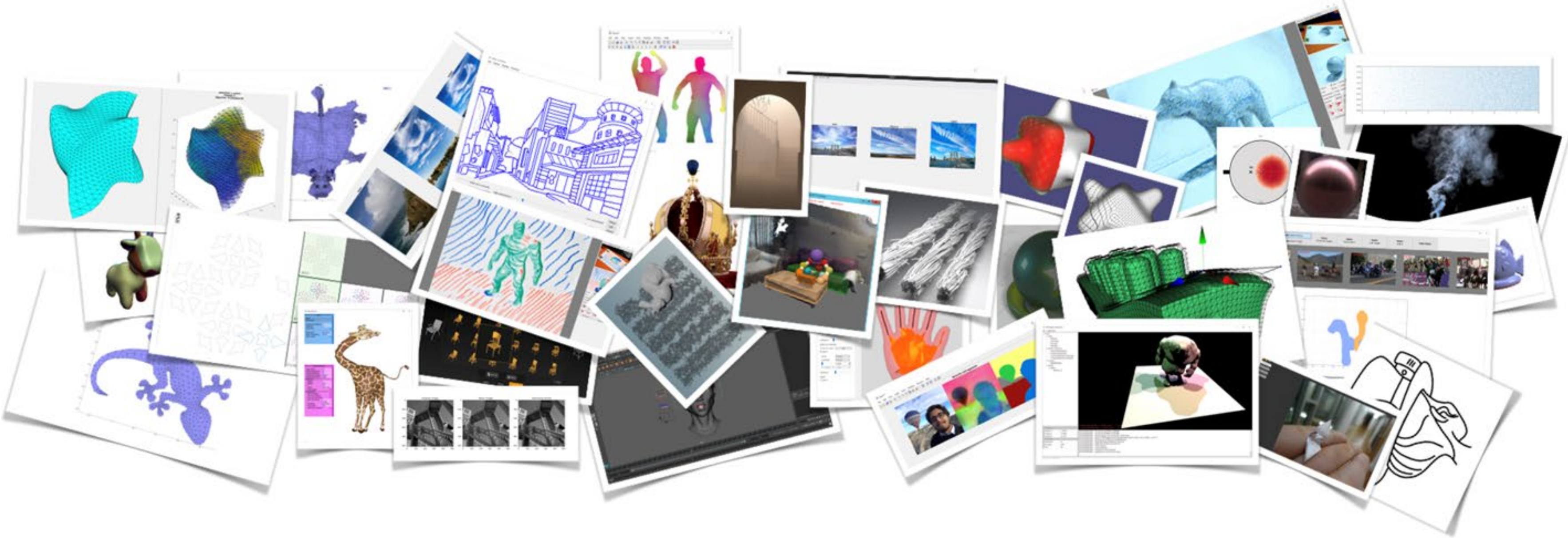
- Analyze trends
  - In time
  - With respect to paper impact, sub-communities, authors (academia/industrial)

- Encourage sharing of codes
  - Development of a community website http://replicability.graphics/

PROCESS

From 374 papers...

...we ran 151 codes



STATE OF THE ART

CONTEXT

- « Reproducibility Crisis » in experimental sciences
  - Psychology/social sciences replication rates of 36% [Open Science Collaboration 2015] to ~79% [Makel et al. 2012]
  - Wide disparities across fields: 55% results deemed reproducible in engineering, 73% in physics [Baker 2016]

- Reproducibility in computational sciences
  - Hydrology: 0.6-6.8% of 1,989 papers deemed reproducible using available data/code/software. [Stagge et al. 2019]
    - Important problem: Lack of documentation (89% tested articles)
  - Artificial Intelligence: Over 400 IJCAI/AAAI papers: 6% share code, 56% training data, 30% test data [Hutson 2018]
  - Image processing: at IEEE TIP, code availability = 2x citations [Vandewalle 2019]

No existing study for CG

REPRODUCIBILITY/REPLICABILITY EFFORTS

- Reproducibility challenge at ICLR in Machine Learning
- Reproducibility labels in Pattern Recognition

- Image Processing On Line (IPOL) journal, Journal of Computer Graphics Techniques (JCGT)
- Graphics Replicability Stamp Initiative (GRSI) http://replicabilitystamp.org

- Artifact Evaluation Committee in Programming languages (additional presentation time at conferences, extra page…)

- **Many technical initiatives**: mloss, RunMyCode, ResearchCompendia, paperswithcode
- **Long-term storage of code**: Software Heritage, Github Archive Program

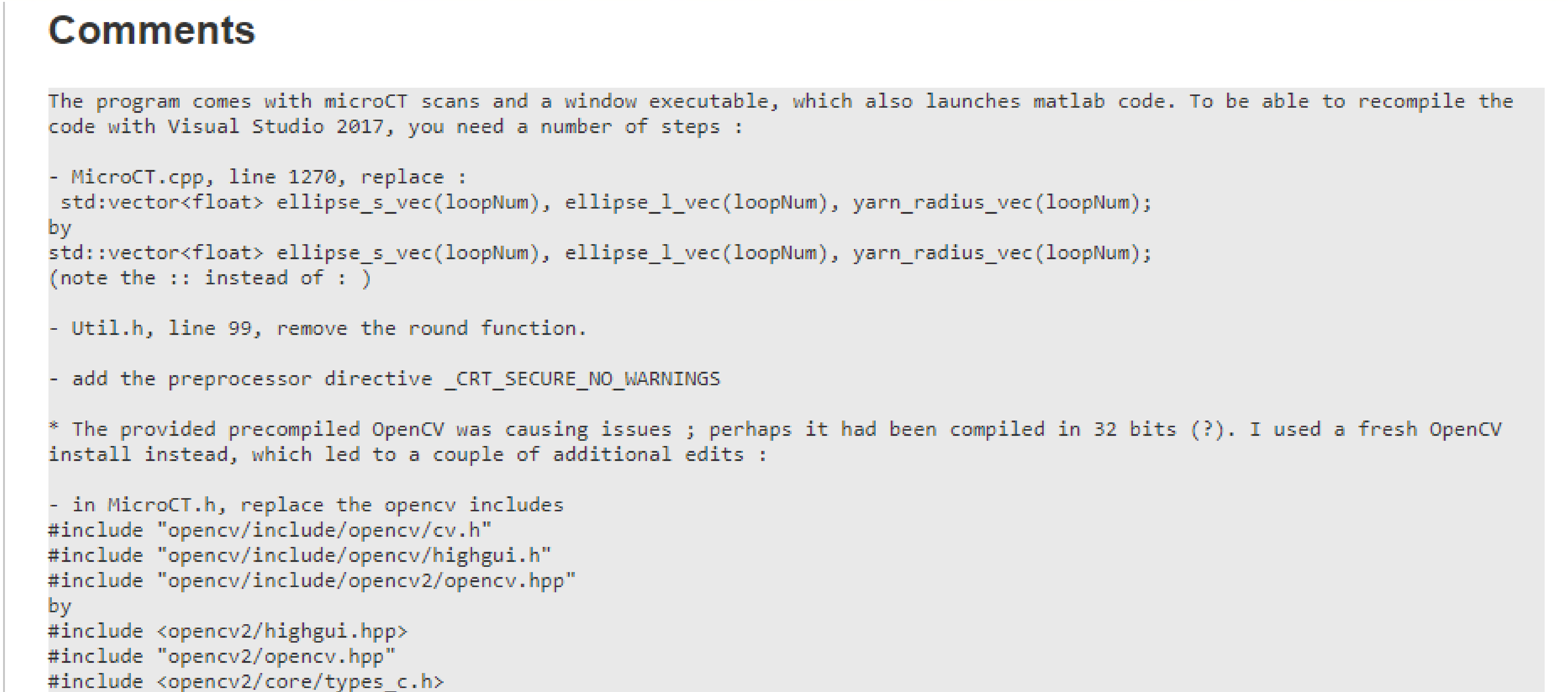
Software Heritage

OUR APPROACH

OUR APPROACH

- We report factual information
  - ACM keywords, paper URL etc.
  - Presence of code, of documentation, of algorithm, type of software (code/binary), license, library dependencies
  - Impact metric: Google scholar citation count, Altmetric
  - Categorize papers: Rendering, Animation and simulation, Geometry, Images, VR, Fabrication
  - How to make the code run, the time we spent making it run
- We report subjective assessment
  - Documentation score [1..3]
  - Algorithm reproducibility score [1..5]
  - Dependencies score [1..5]
  - Fixing bugs score [1..5]
  - Build score [1..5]
  - Code replicability score [1..5]

Our approach



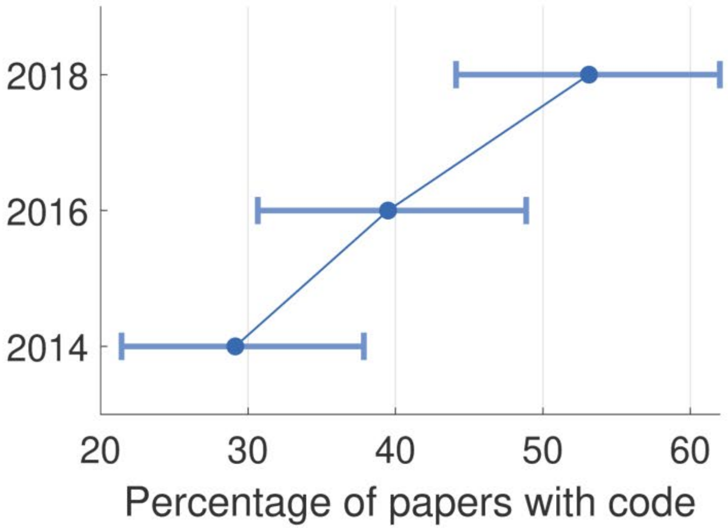
Comments

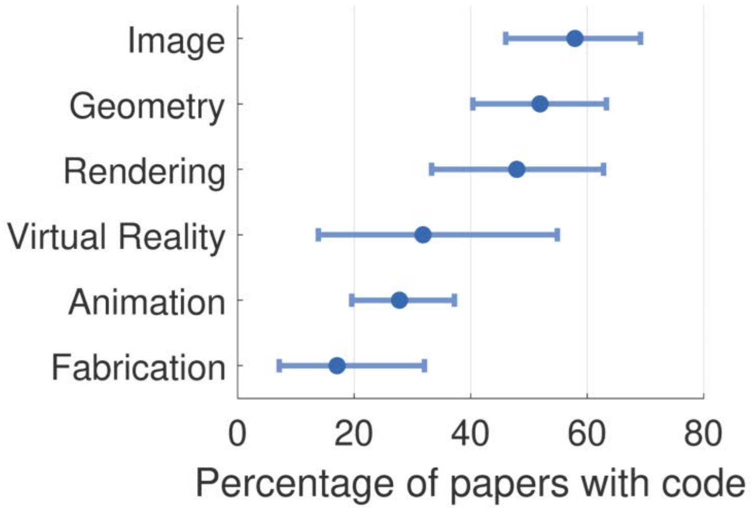
```
The program comes with microCT scans and a window executable, which also launches matlab code. To be able to recompile the
code with Visual Studio 2017, you need a number of steps :

- MicroCT.cpp, line 1270, replace :
 std:vector<float> ellipse_s_vec(loopNum), ellipse_l_vec(loopNum), yarn_radius_vec(loopNum);
by
std::vector<float> ellipse_s_vec(loopNum), ellipse_l_vec(loopNum), yarn_radius_vec(loopNum);
(note the :: instead of : )

- Util.h, line 99, remove the round function.

- add the preprocessor directive _CRT_SECURE_NO_WARNINGS

* The provided precompiled OpenCV was causing issues ; perhaps it had been compiled in 32 bits (?). I used a fresh OpenCV
install instead, which led to a couple of additional edits :

- in MicroCT.h, replace the opencv includes
#include "opencv/include/opencv/cv.h"
#include "opencv/include/opencv/highgui.h"
#include "opencv/include/opencv2/opencv.hpp"
by
#include <opencv2/highgui.hpp>
#include "opencv2/opencv.hpp"
#include <opencv2/core/types_c.h>
```

OUR RESULTS

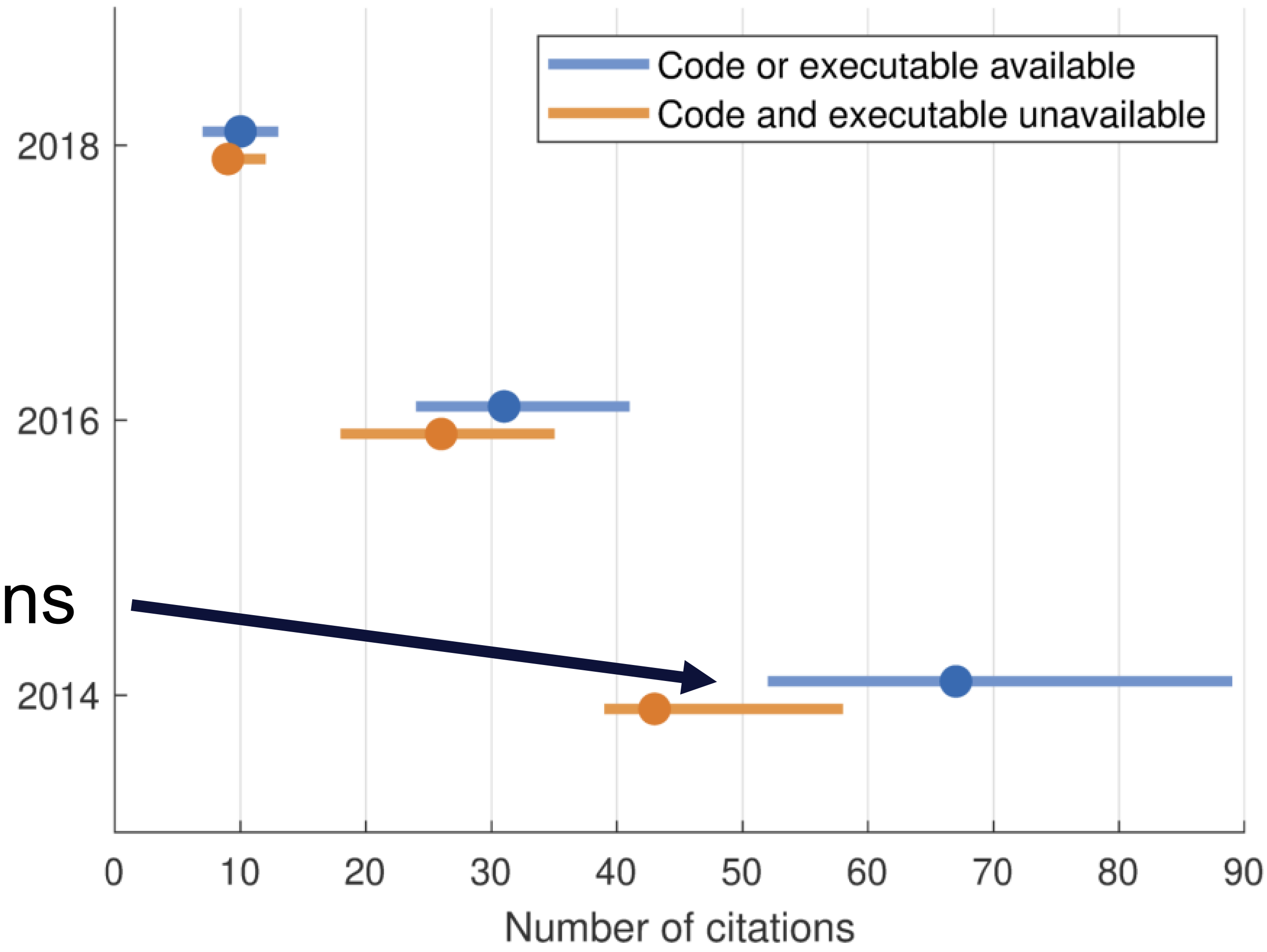
RESULTS

- Paper availability
  - Over 374 papers, only 2 are only available with subscription on the ACM Digital Library
  - 27 as archived pre-prints (HAL, arXiv), 44 are ACM Open Access
- Code availability
  - Over 374 papers: 133 codes (19 we could not run due to tech issues, 5 due to hardware), 18 binaries
  - 60 codes without license information, 11 without documentation nor instructions at all.





RESULTS



+55% citations

RESULTS

- No impact of year on replicability score (!)

- Had to modify 68 out of 133 codes (!)
  - 20 codes deemed hard to fix
  - 27 took more than 100 minutes (spoiler: could have taken days)

- Effect of academia (45.4%) vs. industrial (31.3%)
  - Real issue here: no double standard should be accepted in science.

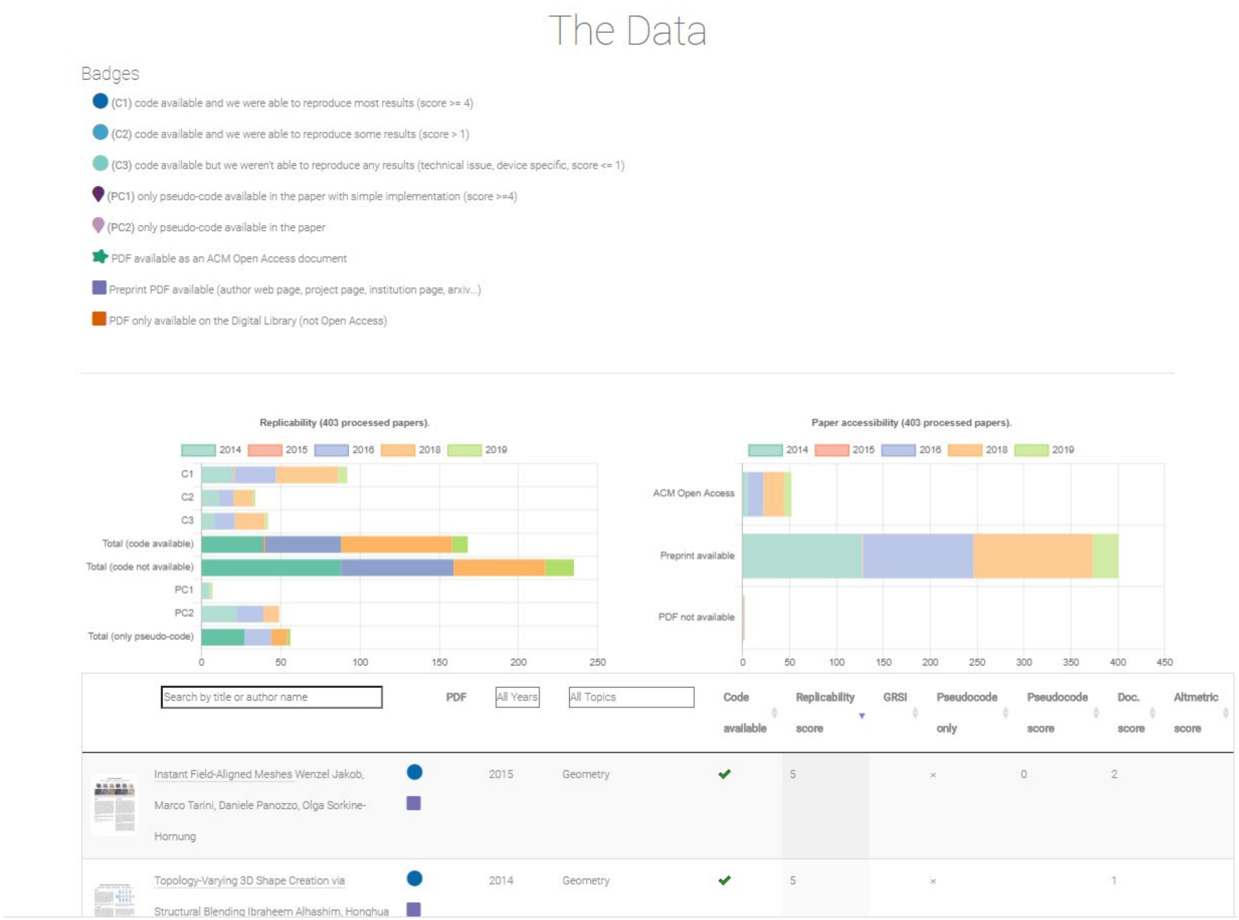
RESULTS

- Common issues
  - **Evolved dependencies / missing versions / missing makefiles**
    - E.g. TensorFlow 1.4.0 require to downgrade CUDA drivers to v8
    - Python 2.7 not maintained as of January 1st, 2020, Caffe not maintained anymore, syntax changes in Pytorch, Qt, etc.
    - Precompiled libraries for outdated compilers
    - Real and important issue, notably in deep learning.
  - **Missing pre-trained neural networks**
    - Re-training can take days, datasets can be huge
  - Lack of instructions
    - No default parameters
    - Undocumented output (sometimes, raw numbers in the console) or file types

- Occasionally: Code that would have never run as is (e.g., merge conflict, syntax errors)

WEBSITE

- Browse and contribute: https://replicability.graphics/browse.html#data



The Data

Badges

- ● (C1) code available and we were able to reproduce most results (score >= 4)
- ● (C2) code available and we were able to reproduce some results (score > 1)
- ● (C3) code available but we weren't able to reproduce any results (technical issue, device specific, score <= 1)
- (PC1) only pseudo-code available in the paper with simple implementation (score >=4)
- (PC2) only pseudo-code available in the paper
- PDF available as an ACM Open Access document
- ■ Preprint PDF available (author web page, project page, institution page, arxiv...)
- ■ PDF only available on the Digital Library (not Open Access)

	Search by title or author name	PDF	All Years	All Topics	Code available	Replicability score	GRSI	Pseudocode only	Pseudocode score	Doc. score	Altmetric score
	Instant Field-Aligned Meshes Wenzel Jakob, Marco Tarini, Daniele Panozzo, Olga Sorkine-Hornung	● ■	2015	Geometry	✓	5		×	0	2	
	Topology-Varying 3D Shape Creation via Structural Blending Ibraheem Alhashim, Honghua	● ■	2014	Geometry	✓	5		×		1	

LIMITATIONS

- Hardware papers
  - Could not assess some papers: require Hall sensors, microcontrollers, spatial light modulators etc.
- Assessment at time *t*
  - Maybe these codes will not run next year
  - Hopefully authors will fix their codes
- No perfect solution for dependencies issues
  - Virtualization, docker, anaconda, Nix…

RECOMMENDATIONS

RECOMMENDATIONS

- ***For authors***
  - Avoid libraries when possible (e.g., avoid OpenCV just to load image files)
  - At least describe library versions, or ship libraries
  - Document code
  - Provide data
  - Provide pre-trained neural networks
- ***For program chairs***
  - Set (later) deadline for code
  - Communication around replicability and code
- ***For publishers***
  - Identifying codes (separate from "supplementary materials")
  - Long-term code storage (Software Heritage ?)

And now...

And now...

- ***Siggraph 2019, Siggraph 2020, Siggraph Asia 2015 partially analyzed***
  - S2019: 66 papers analyzed (27 with code)
  - S2020: 9 papers analyzed (4 with code)
  - SA2015: 1 paper analyzed (1 with code)
- ***Almost no contribution outside of the authors***
  - Reasons:

    submission process too complicated?
    Extra work not deemed valuable enough?
    An unconscious will to preserve the status quo on replicability in Graphics?
  - But: A few corrections by the paper authors (e.g. when code was not found in our early analysis)
- ***A link was established with the Graphics Replicability Stamp***
  - Papers with good replicability scores were invited to submit their paper to GRSI
  - The authors were invited to join the GRSI committee