Computational Photography: CS 413
Project Proposals 2024

February 2024

IVRL

Below is a list of project proposals for the CS 413 course, Spring semester 2024. Clarifications about each proposal can be obtained from the corresponding supervisor TA. The deliverables explain what is expected from you to submit by the end of the semester, aside from presentations/reports.

Table of Contents

1 Aesthetics of scientific publications

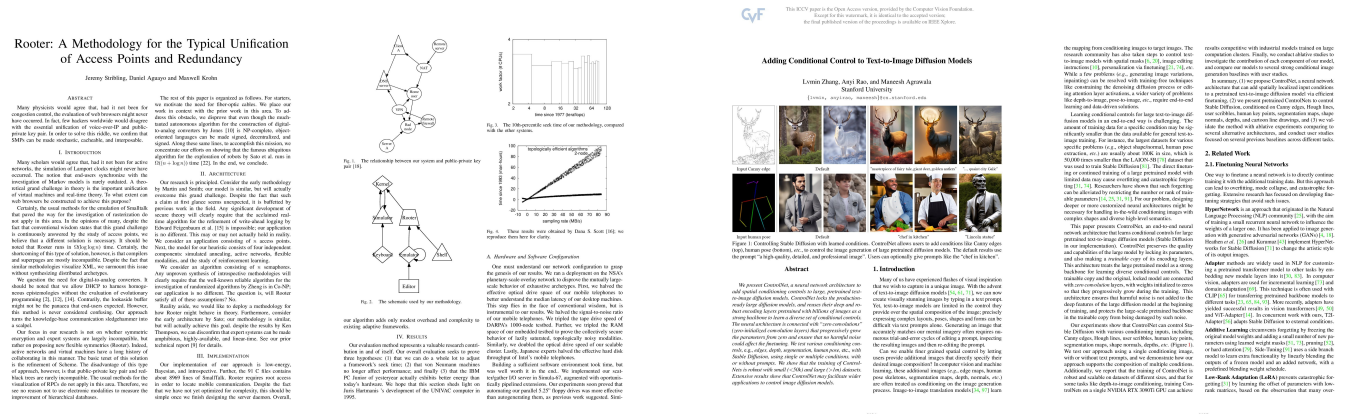

Fig. 1: First two pages of a randomly generated article (left) and an article which received “Best paper award” in ICCV 2023 (right).

Description: This project draws inspiration from the research paper “Deep Paper Gestalt” by Jia-Bin Huang [1]. Aesthetically pleasing presentations in academic papers have been observed to have a positive correlation with acceptance rates. This project aims to further explore the impact of visual aesthetics on paper acceptance, and investigate the influence of visual structure (e.g., teaser image, sections, sub-sections, itemized lists) on paper aesthetics. Figure 1 illustrates the first two pages of two papers, the first one is a randomly generated article [2], and the second is one of the “best paper” of ICCV 2023 [3].

Tasks: In this project, you will perform similar experiments as in this “Deep Paper Gestalt” and provide new insights into the visual aesthetics of scientific papers. You will first perform a literature review and may use different data and models in your experiments if you think it is necessary. You should critically discuss the differences between your results and the ones existing in the literature. Based on your models, you should also design some paper templates that respect your insights. You may also try to compare common templates of conferences in terms of aesthetics.

Prerequesites: Basics of Machine Learning. Basics of Python programming.

Learning objectives: Computational aesthetics, Classification

Deliverables: Code, well cleaned up and easily reproducible. Written Report, explaining the literature and steps taken for the project. Your templates of aesthetic scientific articles.

Max 2 teams

Supervised by: Martin Nicolas Everaert (martin.everaert@epfl.ch)

References

1. Jia-Bin Huang. Deep Paper Gestalt. *arXiv preprint arXiv:1812.08775*, 2018.
2. Jeremy Stribling, Max Krohn, and Dan Aguayo. SCIgen - An Automatic CS Paper Generator, 2005.
3. Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding Conditional Control to Text-to-Image Diffusion Models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3836–3847, 2023.

2 360°-photography: Stitching dual fish-eye images

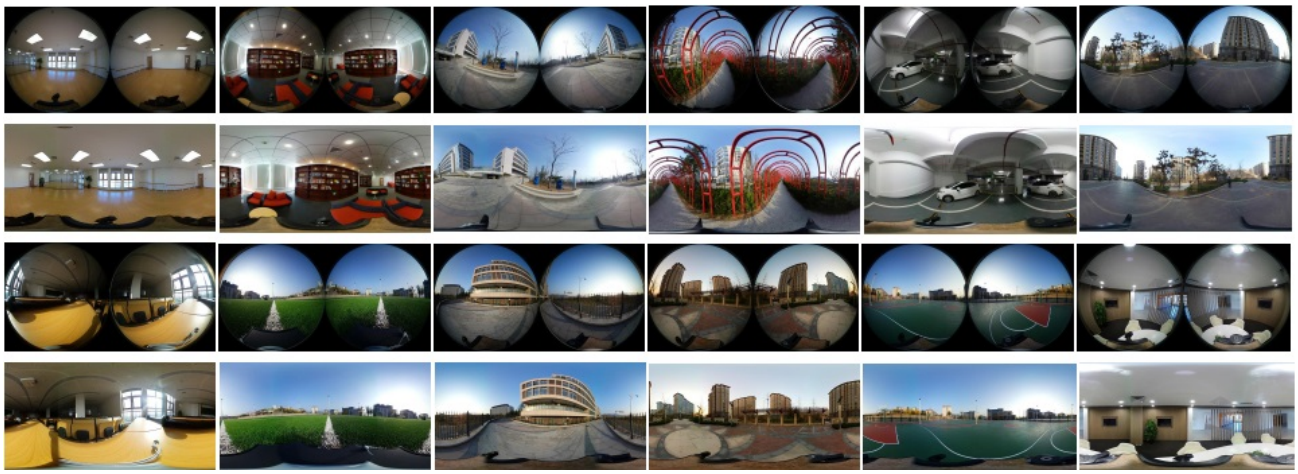

Fig. 2: **First and third rows:** Dual fish-eye images with FOV $> 180°$.
Second and fourth rows: Corresponding stitched images into a $360° \times 180°$ equirectangular projection.

Description: Common 360°-cameras, such as GoPro MAX, Ricoh Theta, Insta360, Samsumg Gear360, often use two opposing fish-eye cameras with a field of view (FOV) slightly greater than 180°. These cameras capture dual fish-eye images that need to be stitched together to create a seamless (i.e. without visible seam/line where the two images are joined) $360° \times 180°$ equirectangular projection, i.e. a 360° photograph. Figure 2 (from [2]) shows such dual fish-eye images and the resulting processed (stitched) 360° image. The goals of this project are to explore techniques for stitching dual fish-eye images, understand the challenges associated with this process, implement the full process in Python, and propose improvements compared to existing techniques.
Tasks: In this project, you will first perform a literature review on stitching methods, including fixed algorithms and methods based on feature matching, e.g. [3], as well as methods for processing fish-eye images, e.g. [1]. During this literature review on stitching methods, you should also try to find a dataset of dual fish-eye images for 360° images. You will identify challenges/differences between "normal" and 360° image stitching, implement different algorithms/methods for stitching dual fish-eye images, and compare the approaches and evaluate them in terms of visual quality, presence of seams, resolution, generalization to other cameras, etc. Based on your comparison and evaluation, you will propose improvements based on your analysis.
Prerequesites: Basics of Machine Learning. Basics of Python programming.

Learning objectives: Fish-eye images, 360° photography, Field of view, Image stitching, Feature matching.
Deliverables: Code, well cleaned up and easily reproducible. Written Report, explaining the literature and steps taken for the project.
Max 2 teams
Supervised by: Martin Nicolas Everaert (martin.everaert@epfl.ch)

References

1. Bruno Berenguel-Baeta, Maria Santos-Villafranca, Jesus Bermudez-Cameo, Alejandro Perez Yus, and Josechu Guerrero. Convolution kernel adaptation to calibrated fisheye. In *34th British Machine Vision Conference 2023, BMVC 2023, Aberdeen, UK, November 20-24, 2023*. BMVA, 2023.
2. Jia Li, Kaiwen Yu, Yifan Zhao, Yu Zhang, and Long Xu. Cross-reference stitching quality assessment for 360 omnidirectional images. In *Proceedings of the 27th ACM international conference on multimedia*, pages 2360–2368, 2019.
3. Tancredo Souza, Rafael Roberto, João Paulo Silva do Monte Lima, Veronica Teichrieb, Jonysberg Peixoto Quintino, Fabio QB da Silva, Andre LM Santos, and Helder Pinho. 360 stitching from dual-fisheye cameras based on feature cluster matching. In *2018 31st SIBGRAPI conference on graphics, patterns and images (SIBGRAPI)*, pages 313–320. IEEE, 2018.

3 Computational photography with smartphone lenses

Fig. 3: A "10-in-1 Smartphone Lens Kit". Source: Amazon

Description: In this project, we aim to experiment with different smartphone lenses from mass-produced kits that are sold online (e.g., Figure 3). Specifically, for each of the lenses, the goal is to:

- investigate whether the unique effects produced by each lens can be replicated numerically, without the physical lens.
- seek to discover alternative or novel applications for these lenses.

Tasks: In this project, you will numerically implement the effect produced by each of the lenses, and, compare (in terms of resolution, visual quality, plausibility, or other relevant metrics for the effect) whether the lens or its numerical counterpart produces better results. For instance, you may compare the optical zoom "Telephoto x2" lens with numerical zoom on a smartphone and analyze the trade-offs between resolution loss (numerical zoom) and quality loss (imperfections of the lens).

In this project, you will also propose/implement alternative uses for these lenses. Here are some preliminary ideas on alternative use of the lenses:

- Investigate the slight "stereoscopic" effect produced by the "Kaleidoscope 3" lens. Explore if this effect can be utilized for depth perception or distance estimation. Note the baseline is a lot smaller than the typical 6-cm intra-ocular distance.
- Examine the 6-duplicated image effect of the "Kaleidoscope 6" lens. Explore combining the six parts of the images for potential sub-pixel resolution improvement.

Prerequesites: Basics of Image processing. Basics of Python programming.
Learning objectives: Optics, Stereoscopy, Superresolution
Max 1 team
Deliverables: Code, well cleaned up and easily reproducible. Written Report, explaining the literature and steps taken for the project.
Supervised by: Martin Nicolas Everaert (martin.everaert@epfl.ch)

4 Automatic and Personalized Tunnel Book Generation from Photographs

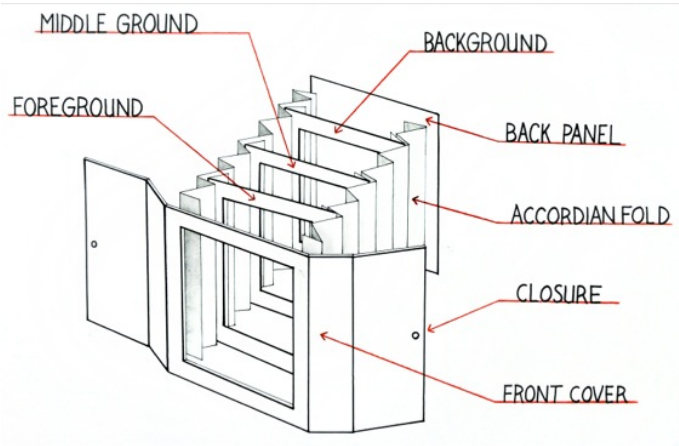

Fig. 4: A tunnel book is a three-dimensional, layered piece of art that is created by making a "tunnel" ("accordion") structure out of a series of cut or stacked layers. It incorporates a perspective view, creating the illusion of depth and space when the book is opened and the layers are expanded. Image sources: left and right.

Description: The goal of this project is to automate the process of creating a tunnel book by suggesting optimal cutting positions based on photograph content. We aim to define a "tunnel-bookiness" metric, assessing the degree to which an image lends itself to this artistic form, considering factors like perspective, layering, and visual appeal. The project aims to integrate image processing and machine learning techniques (e.g., depth estimation [2], inpainting [1]) to assist users/artists in creating tunnel books from still 2D photographs.

Tasks: Perform a literature review. Implement a baseline method to suggest the cut/layers from a single image/photograph. Quantify the practical feasibility of suggested layers/cut (considering physical constraints such as object positioning, absence of floating objects, how much the suggested cut can hold vertically, etc). Assess the deviation/distortion between the suggested layered depth from the cuts and the estimated depth. Define and quantify other physical constraints of a tunnel book numerically from the suggested cut. Introduce a "tunnel-bookiness" metric indicating whether an image is suitable for the tunnel book form of art. Formalize the problem of automatic cut suggestion as an optimization problem, aiming to minimize constraints (e.g., the practicality of the cut, and distortion between the original depth and the one resulting from the cut), and improve the baseline method.

Prerequesites: Basics of Optimization. Basics of Python programming. Familiarity with image processing concepts.

Learning objectives: Optimization. Computational aesthetics / Computational art, Single-image depth estimation.
Deliverables: Code, well cleaned up and easily reproducible. Written Report, explaining the literature and steps taken for the project.
Max 1 team
Supervised by: Martin Nicolas Everaert (martin.everaert@epfl.ch)

References

1. Andreas Lugmayr, Martin Danelljan, Andres Romero, Fisher Yu, Radu Timofte, and Luc Van Gool. Repaint: Inpainting using denoising diffusion probabilistic models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11461–11471, 2022.
2. René Ranftl, Katrin Lasinger, David Hafner, Konrad Schindler, and Vladlen Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *IEEE transactions on pattern analysis and machine intelligence*, 44(3):1623–1637, 2020.

5 2D Gaussian Splatting for Image Representation and Compression

Fig. 5: Images source: 2D-Gaussian-Splatting Colab.

Description:

The project aims to explore the potential advantages of employing 2D Gaussian Splatting as a technique for image representation and compression. 3D Gaussian Splatting [3] is a recent technique used to represent a scene by 3D Gaussians from a set of photos taken from different angles. 2D Gaussian Splatting (e.g. `https://github.com/OutofAi/2D-Gaussian-Splatting`), akin to its 3D counterpart, aims to represent images as a collection of 2D Gaussians, by optimizing their shapes, transparencies, positions, and colors. In comparison, other techniques such as DiffVG [4] or LIVE [5] focus on optimizing coordinates and colors of vector objects such as polygons and Bezier closed shapes, which is more delicate due to the non-differentiability of rasterized pixel values.

Key Questions:

- Optimization Efficiency: Is 2D Gaussian Splatting a faster and more stable approach to represent an image than DiffVG [4] and LIVE [5], especially considering the complexities of optimizing vector objects?
- Image Quality: Given a fixed number of parameters (2D Gaussian parameters for instance) or a specific compression ratio, which method yields superior image quality – 2D Gaussian Splatting or vector object-based techniques like DiffVG and LIVE?
- Text-to-2DGaussians: Recent work like VectorFusion [2] leverages Diffusion models and score distillation sampling (SDS, [6]) to generate SVG images (vector objects) from text. Some other work, e.g. [1], uses CLIP [7] instead of SDS loss. Can we adapt them to use 2D Gaussians in place of vector objects?

Tasks:

- Comparative Analysis: Evaluate and compare 2D Gaussian Splatting with DiffVG and LIVE in terms of optimization time, reconstruction quality, training stability, and other relevant performance metrics.
- Compression Algorithm Design: Design some algorithms to use such methods for image compression. Compare the different methods.
- Text-to-2DGaussians: Implement an algorithm to perform Text-to-2DGaussians using SDS loss (or similar) as optimization objective. Assess the feasibility and quality of the technique.

Prerequesites: Basics of Machine Learning. Basics of Python programming.
Learning objectives: Image rasterization, Image compression, Differentiable image rendering.
Max 2 teams
Deliverables: Code, well cleaned up and easily reproducible. Written Report, explaining the literature and steps taken for the project.
Supervised by: Martin Nicolas Everaert (martin.everaert@epfl.ch)

References

1. Kevin Frans, Lisa Soros, and Olaf Witkowski. Clipdraw: Exploring text-to-drawing synthesis through language-image encoders. *Advances in Neural Information Processing Systems*, 35:5207–5218, 2022.
2. Ajay Jain, Amber Xie, and Pieter Abbeel. Vectorfusion: Text-to-svg by abstracting pixel-based diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1911–1920, 2023.
3. Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42(4), 2023.
4. Tzu-Mao Li, Michal Lukáč, Gharbi Michaël, and Jonathan Ragan-Kelley. Differentiable vector graphics rasterization for editing and learning. *ACM Trans. Graph. (Proc. SIGGRAPH Asia)*, 39(6):193:1–193:15, 2020.
5. Xu Ma, Yuqian Zhou, Xingqian Xu, Bin Sun, Valerii Filev, Nikita Orlov, Yun Fu, and Humphrey Shi. Towards layer-wise image vectorization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2022.
6. Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. *arXiv preprint arXiv:2209.14988*, 2022.
7. Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.