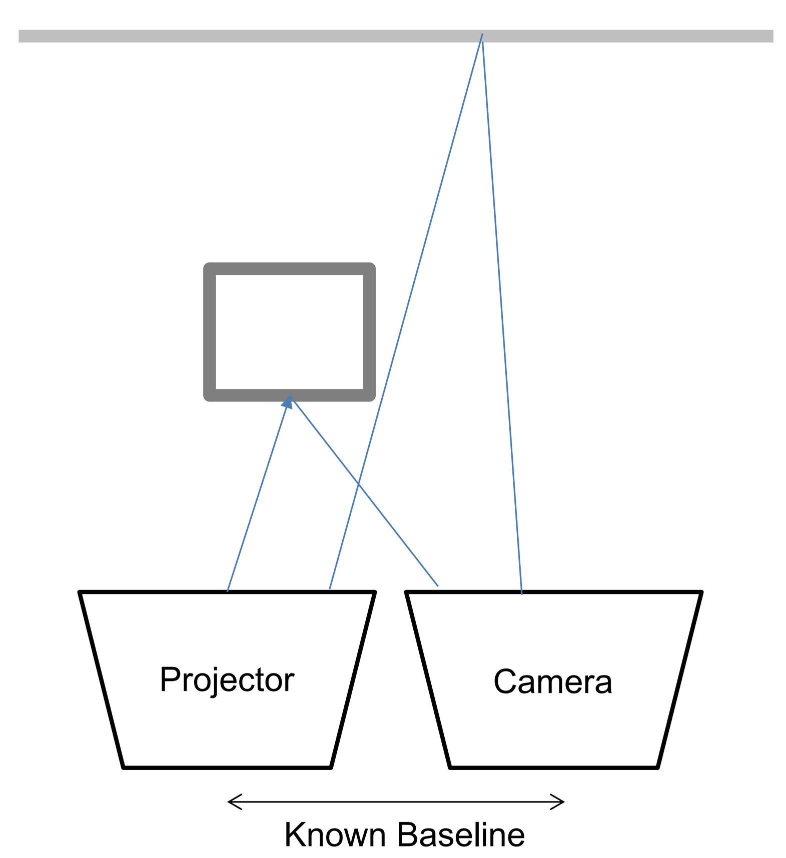
Structured Light Arrays using Differentiable Rendering

Yasasa Abey

Motivation

- We want to design light patterns for systems that use both a **projector** and a **camera** to capture the scene.
- The camera is responsible to find the corresponding projector pixel for every observed pixel:
  $o_{ij} = Decode\left(Neighbourhood(c_{ij})\right)$
- If we can find a pattern such that for every observed camera pixel, we find the corresponding projector pixel then we can do stereo imaging.
- *Can we determine the optimal pattern to project light to make the task of finding correspondences the easiest?*



Related Work

Hand Tuned Patterns

- We can design the patterns and the specific decoder using a list of heuristics, that maximize the ability of the camera to detect under noise assumptions [1]
- However, the patterns are not tuned for specific materials nor is there any criterion for the quality of the pattern.

Optimized Patterns

- Under an epipolar transport model and a simple but optimal decoding algorithm in the maximum likelihood sense, we perform an optimization for the pattern that minimizes the missed correspondences[2].
- The optimization can be carried out using a random texture to tune arbitrary structured light systems[3].
- *Can we perform the pattern optimization using differentiable rendering, with less assumptions on the light transport?*

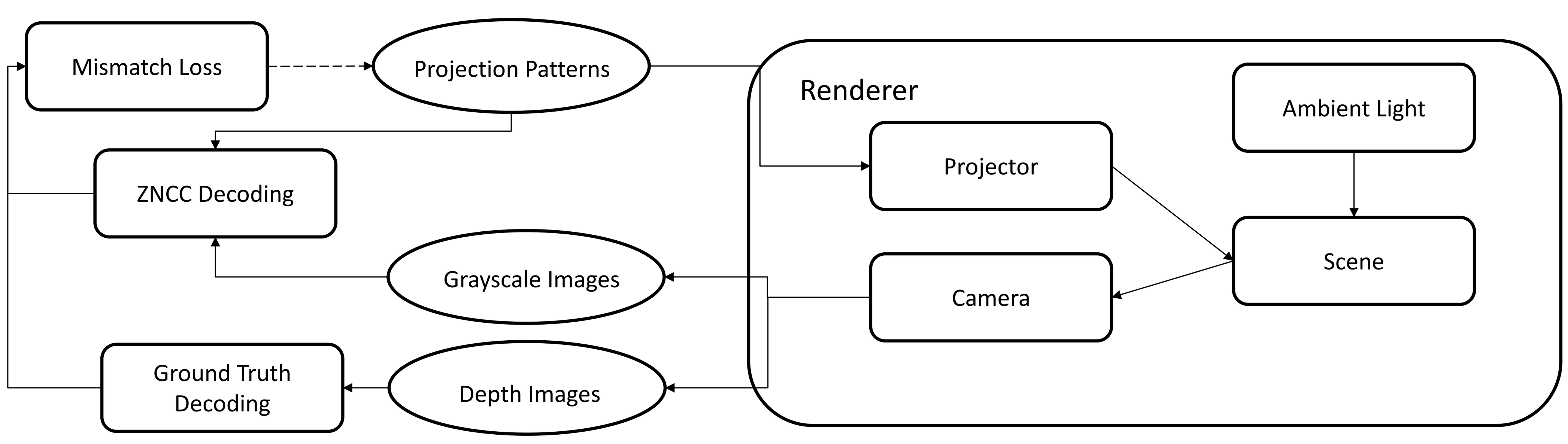
References

[1] M. Gupta and S. Nayar, “Micro Phase Shifting,” CVPR 2012
[2] P. Mirdehghan, W. Chen, and K. N. Kutulakos, “Optimal Structured Light `a La Carte,” CVPR 2018
[3] W. Chen, P. Mirdehghan, S. Fidler and K. N. Kutulakos, “Auto-Tuning Structured Light by Optical Stochastic Gradient Descent,” CVPR 2020
[4] Jakob, Wenzel, et al. "Mitsuba 3 renderer, 2022." URL: https://mitsuba-renderer.org

Method

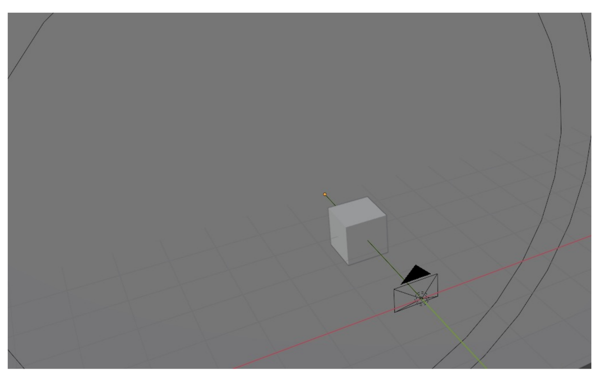
- We would use differentiable rendering to generate light patterns in using end to end optimization between the projector and the camera
- Using end-to-end optimization allows our method to discard assumptions on the environment while generating light patterns using gradient descent.
- We use the fix our decoder to be zero-noise cross correlation decoder presented in [2], where the observations are obtained from the Mitsuba renderer[4].
- For each observation, column in the camera image o_j we compute the probability of observing the corresponding code vector c_i. Define this map to be $\hat{p}\{(j,i)$.
- We compute the ground truth $p(i,j)$ using the camera depth map and the intrinsics, then our loss becomes:

$$L(c) = \sum -p(i,j)\log(\hat{p}(i,j))$$

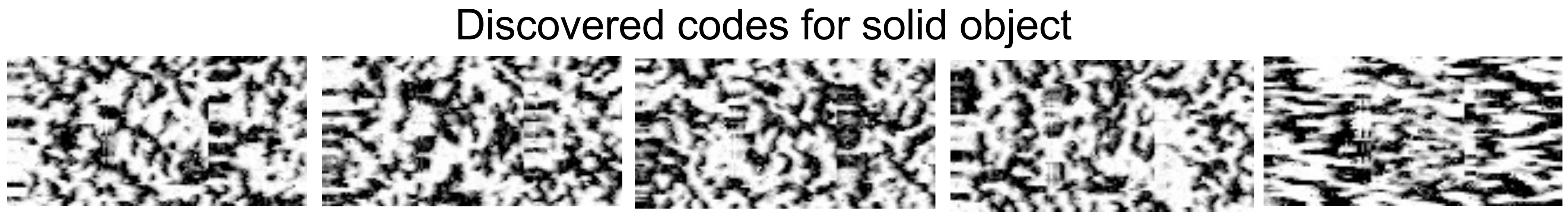


Experimental Results

- **Test setup:**

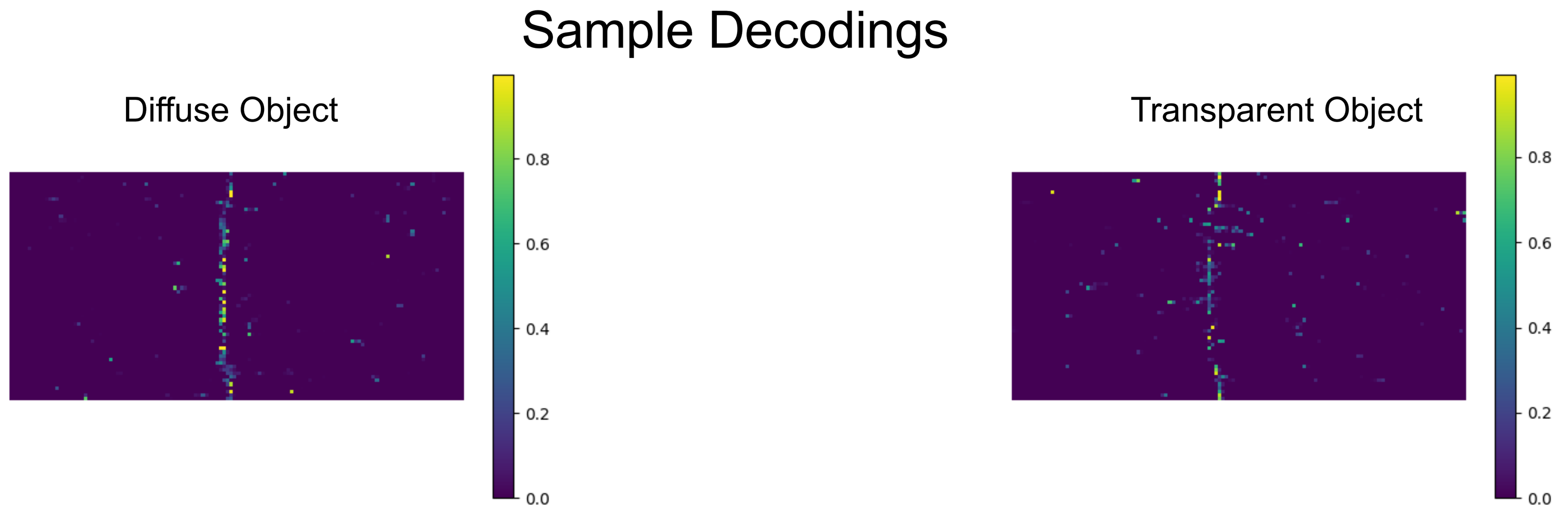


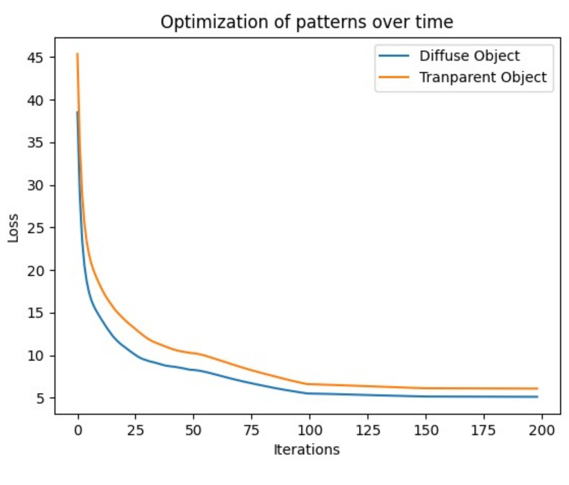
Discovered codes for solid object



Discovered codes for transparent object

Sample Decodings





- Patterns were optimized for 200 gradient steps, each with a batch size of 8
- For each sample of the batch, we sample a random orientation and position for the cube as well as the level of ambient light.