Rendered Data pipeline analysis and optimization

Background

The LEGO Group is looking to use rendered images to train their vision system to recognize and localize elements in a scene. The data is currently rendered using BlenderProc. Two kinds of images are currently rendered, one with a controlled environment and a single element reference present in the scene to teach our pose estimation NN, one with randomized scene parameters and multiple references in the scene for reference recognition. In the future, the dataset could potentially be combined.

Problem statement

With the thousands of LEGO elements in the portfolio, the evaluated cost for rendering all the necessary images is high. The pipeline is still at the proof of concept and no optimization of the BlenderProc or analysis of the market has been executed. Additionally, the data generated is oriented towards specific applications. It would therefore require generating additional data for new applications, which would also limit the financial benefits. Additional applications in the LEGO Group would benefit for a more general data set (Quality inspection, sorting, machine optimization).

Purpose

The project is to help LEGO pursue their journey towards using more rendered data for their machine vision application. The first objective is therefore to define if the LEGO Group is using the right tool. The capabilities of BlenderProc should be compared with Isaac Sim and an additional option selected by the project team. The second objective is to generalize our rendering data by combining necessary scene requirements and increasing the data linked to images.

Finaly, because the purpose of the rendered data is solely to optimize our machine vision recipe creation, the project should evaluate the quality of their rendered data in real scenarios.

Task

1. Compare the capabilities of the 3 software for the LEGO application. (resources required, automation capabilities, programming…)
2. Using the preferred software, combine the two rendering applications into one and evaluate its benefit and downfall.
    a. The rendering application should create enough images and instances of an element to teach the NN to localize and identify a given element.
    b. The data linked to the images should at least include reference, position, colors, textures
    c. The evaluation should consider cost and automation challenges.
3. Develop a performance evaluation method to test the current LEGO machine vision algorithms on the newly generated data,
4. Extra rendering task: Optimizing current Blender Proc solution to reduce associated rendering cost.
5. Extra vision task: For a given element, use the rendered data to teach an anomaly detection algorithm to detect scratches on a real element.

Resources

1. Blender Proc current application repository
2. Current vision applications repository
3. LEGO bricks
4. Computational resources

Contact: Rikke Gade, rg@create.aau.dk