Computational Photography Lab.

Find the project web page here:
http://yaksoy.github.io/dynapix/

DynaPix: Normal Map Pixelization for Dynamic Lighting

Gerardo Gandeaga
Simon Fraser University

Denys Iliash
Simon Fraser University

Chris Careaga
Simon Fraser University

Yağız Aksoy
Simon Fraser University

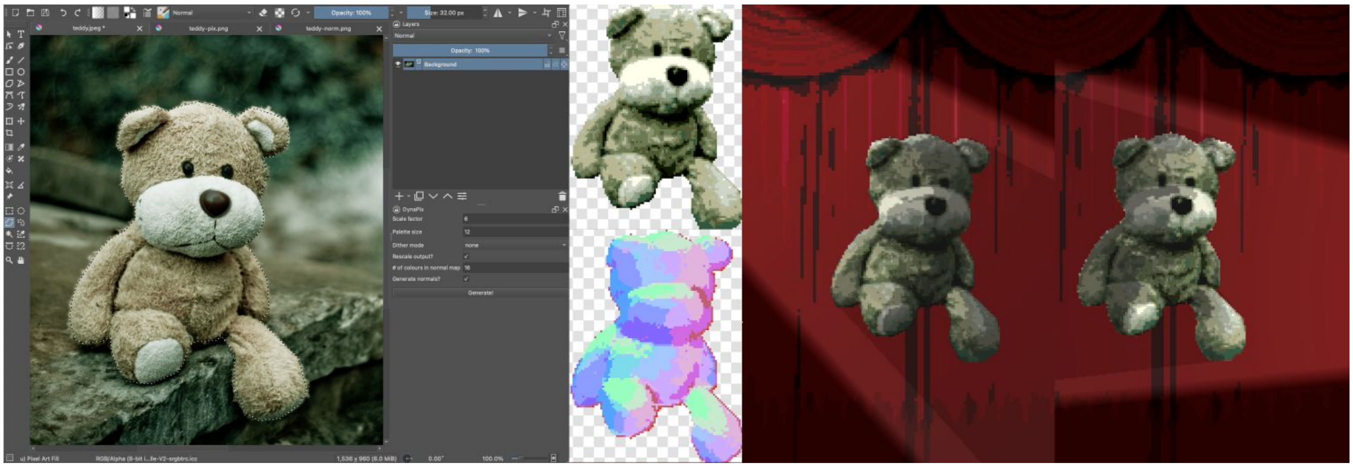

Figure 1: DynaPix is a Krita extension that uses an existing pixelization engine and neural network surface normal estimator to generate pixelated images and their corresponding normal maps. These pixelized representations can be easily integrated into modern game development engines for dynamic relighting.

ABSTRACT

This work introduces DynaPix, a Krita extension that automatically generates pixelated images and surface normals from an input image. DynaPix is a tool that aids pixel artists and game developers more efficiently develop 8-bit style games and bring them to life with dynamic lighting through normal maps that can be used in modern game engines such as Unity. The extension offers artists a degree of flexibility as well as allows for further refinements to generated artwork. Powered by out of the box solutions, DynaPix is a tool that seamlessly integrates in the artistic workflow.

ACM Reference Format:
Gerardo Gandeaga, Denys Iliash, Chris Careaga, and Yağız Aksoy. 2022. DynaPix: Normal Map Pixelization for Dynamic Lighting. In *Special Interest Group on Computer Graphics and Interactive Techniques Conference Posters (SIGGRAPH '22 Posters), August 07-11, 2022.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3532719.3543238

1 INTRODUCTION

Pixel art is an art-form often used in video-game development for it's simplicity and nostalgic effect. When combined with an artist-generated normal map, the pixel art can be used in games with dynamic lighting as well. However, generating quality pixel art assets can be an extremely time-consuming process only accessible to experienced artists. Developing dynamic animations and lighting effects requires the manual creation of key-frame sequences. While tools, such as Sprite Lamp [Morgan 2014], exist to aid in the creation of dynamically relightable assets, they still require users to generate multiple versions of their pixel art assets.

In this work, we present an automatic method for generating pixel art assets and corresponding pixelized normals from a single natural image. We show that normal maps estimated from a photograph using modern neural network-based methods can be used to generate pixel art assets that support dynamic lighting. We introduce a simple extension for the open-source image editing application Krita to enable rapid prototyping of pixel art assets with an integrated pixelization engine and a normal map estimator. We show that our extension can be effectively used in game production by rendering dynamically lit sprites in a game development engine.

2 APPROACH

Surface Normal Estimation. We propose to generate surface normal estimations in the natural image space, allowing for the use of existing normal estimation methods. We utilise the recent neural network based approach of [Zamir et al. 2020] for normal map generation. Although the network is trained on indoor images, it can produce quality results for various diverse scenes. The normal estimation is generated at the same resolution as the input image and is mapped directly to RGB values.

Pixelization. The process of mapping real images to their pixel are equivalent is a non-trivial task. In this work, we utilize an existing pixelization engine called Pyxelate [Nagyfi 2021]. Their method

SIGGRAPH '22 Posters, August 07-11, 2022, Vancouver, BC, Canada

ACM ISBN 978-1-4503-9361-4/22/08.
https://doi.org/10.1145/3532719.3543238

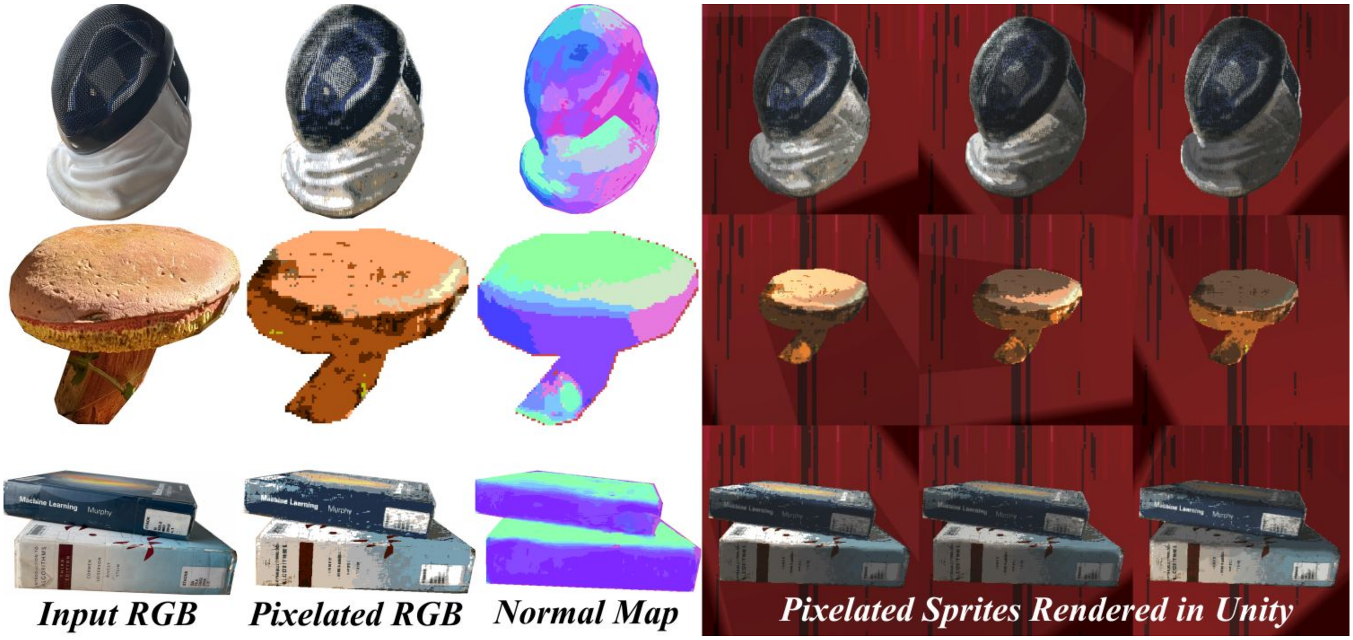

Figure 2: Multiple examples of pixelized inputs and their corresponding relighting results. From left to right: input image, pixelized image, pixelized normal estimation, relighting results in Unity.

is inspired by the Histogram of Oriented Gradients (HOG) technique and perform a downsampling that results in a stylistic pixel art image. This is followed by a Bayesian Gaussian Mixture model to generate a reduced color palette. Pyxelate produces pixel art from any input image with optional user-specified hyperparameters. We find that images pixelated by Pyxelate are aesthetically pleasing and, for many images, the resulting outputs resemble typical video game sprites. In addition to the input image, we additionally pixelate the surface normal esimation using Pyxelate. This generates a pixelized version of the image geometry that can be used for dynamic relighting.

Krita Extension. The main focus of our approach is to artist liberty, and facilitate rapid-prototyping. To this end, we develop a simple extension for the free open-source image editing software called Krita. Our frontend was created using Krita's Python API and PyQT5 GUI library to create a docker extension the user can easily interact with. User-specified configurations are exposed through the docker. The scale factor, palette size, and dither mode specifically expose hyperparameters for Pyxelate, the pixelization engine. The scale factor is the factor by which the dimensions of the image will be divided, for example, a scale factor of 4 would generate an output $\frac{1}{4}$ of the original size. The palette size suggests a lower bound on the number of colors there should be in the image to the Pyxelate engine; for valid pixelization, the minimum palette size must be 2. However, the choice of the palette size is left to the artist's discretion as the desired effect varies from image to image. The number of colors in the normal map is a palette size choice as well, it is passed to Pyxelate when pixelating the normal maps. However, the palette size for an effective normal map must be larger, in our testing, we find that the range of 16-20 created satisfactory maps. Once the desired settings are specified, the system generates the pixelated image and (if specified) the normals from the input image. The resulting images are automatically revealed in new Krita documents for editing and fine tuning. It is typically desirable for game art to maintain consistency across a large set of assets. Although our approach does not guarantee this stylistic consistency, by creating this extension artists are given instant feedback allowing for more creative control.

Results. The resulting images can be directly imported into a game engine that supports normal maps for dynamic lighting. To demonstrate our results we utilized the Unity game engine. Unity is a popular and robust game engine used by indie developers and supports dynamic 2D lighting. The Unity engine itself corrects and tolerates errors produced by DynaPix. You can see all the results generated by DynaPix and their use in Unity scene with dynamic lighting in the Figure 2.

3 CONCLUSION

We utilise off-the-shelf deep learning approach to generate surface normal estimations from natural images. From there, we use an existing algorithm to map the image and normal map into the style of a pixel art asset. The resulting representations can be easily imported into existing game engines for dynamic relighting. In order to enable rapid prototyping of game assets and corresponding normal maps, we implement a Krita extension to allow users to iteratively interact with our proposed pipeline and incorporate it into their game development work-flow.

REFERENCES

Finn Morgan. 2014. *Sprite Lamp*. http://www.spritelamp.com/
Richard Nagyfi. 2021. *pyxelate*. https://github.com/sedthh/pyxelate
A. R. Zamir, A. Sax, N. Cheerla, R. Suri, Z. Cao, J. Malik, and L. J. Guibas. 2020. Robust Learning Through Cross-Task Consistency. In *Proc. CVPR*.