논문 2024-19-11

Instant NGP를 활용한 CNC Tool의 장면 생성 및 렌더링 성능 평가

(Scene Generation of CNC Tools Utilizing Instant NGP and Rendering Performance Evaluation)

정 태 영, 유 영 준*
(Taeyeong Jung, Youngjun Yoo)

Abstract : CNC tools contribute to the production of high-precision and consistent results. However, employing damaged CNC tools or utilizing compromised numerical control can lead to significant issues, including equipment damage, overheating, and system-wide errors. Typically, the assessment of external damage to CNC tools involves capturing a single viewpoint through a camera to evaluate tool wear. This study aims to enhance existing methods by using only a single manually focused Microscope camera to enable comprehensive external analysis from multiple perspectives. Applying the NeRF (Neural Radiance Fields) algorithm to images captured with a single manual focus microscope camera, we construct a 3D rendering system. Through this system, it is possible to generate scenes of areas that cannot be captured even with a fixed camera setup, thereby assisting in the analysis of exterior features. However, the NeRF model requires considerable training time, ranging from several hours to over two days. To overcome these limitations of NeRF, various subsequent models have been developed. Therefore, this study aims to compare and apply the performance of Instant NGP, Mip-NeRF, and DS-NeRF, which have garnered attention following NeRF.

Keywords : NeRF, Instant NGP, Mip-NeRF, DS-NeRF, Auto Focus, CNC Tool

Ⅰ. 서 론

CNC 공구는 Computer Numerical Control (CNC) 시스템과 연결되어 사용되는 가공 도구로 [1], 자동화된 기계 가공 및 제조 프로세스에서 특정 형상을 가진 부품을 가공하는 데 사용된다. 고정밀 및 반복 가능한 작업을 수행하여 생산성과 제품 품질을 향상시킴으로써, 현대 스마트 팩토리 구축에 큰 도움을 줄 수 있다. 동시에, 손상된 CNC 공구의 사용은 발열, 기기 손상, 시스템 전체에 대한 오류 유발 등 큰 문제를 야기할 수 있으므로 적합한, 공구의 손상을 진단하는 시스템 구축이 필수적이다.

공구의 외형 정보를 정밀하게 진단하기 위해서는 단일 뷰를 통한 솔루션 대비, 다각도의 뷰를 제공하는 3D 솔루션을 구축하는 것이 효과적이다. 하지만 3D 스캐너를 사용할 경우, 장비의 가격이 일반 카메라 대비 상당히 고가에 속하므로 비용적인 측면에서 큰 부담이 발생한다 [2]. 저가형 스캐너의 경우에도 최소 70만원 대 이상에 포진해있으며, 기업에서 사용되는 산업용 3D 스캐너의 경우 천만 원대 이상의 비용을 요구한다.

NeRF (Neural Radiance Fields) [3] 모델을 사용할 경우, 적은 비용을 통해 3D 렌더링 시스템을 구축할 수 있다. NeRF는 2D 이미지를 통해 3D 장면을 생성하는 딥러닝 기술로, 학습을 위해 다량의 이미지와 카메라 파라미터만을 입력으로 넣어 카메라가 접근하지 않은 장면을 새롭게 생성, 물체를 3D로 보는 것과 같은 효과를 나타낸다.

NeRF는 타 3D 렌더링 기술 대비 사실적인 장면을 생성하는 점에서 이점을 갖는다. 흔히 사용되는 렌더링 기술인, 스캐닝 방식과 포토그래메트리 (photogrammetry)의 가장 큰 단점은 빛 반사와 굴절에 취약한 특성을 보이는 점이다 [4, 5]. 이를 해결하기 위해서, 주변 광을 제거하기 위한 무광처리 물질을 스캔해야 할 대상 물체에 분사하거나 추가적인 필터를 사용해야 하며, 빛이 표면에 반사되어 나타내는 정보를 담아내기 어렵다. 반면, NeRF의 경우 훌륭한 빛 반사 묘사가 가능하여 현실 객체에 더욱 가까운 장면을 생성할 수 있다. 또한, NeRF의 후속 연구로, 학습 및 렌더링 시간을 비약적으로 단축시킨 Instant NGP [6]모델이 발표됨에 따라, 활용도가 대폭 증가하였다.

본 실험에서는, NeRF의 후속 연구인 Instant NGP를 활용하여, 산업에 적용될 수 있는 CNC 공구의 렌더링 시스템

*Corresponding Author (youdalj@kitech.re.kr)
Received: Jan. 8, 2024, Revised: Feb. 6, 2024, Accepted: Feb. 27, 2024.
T. Y. Jung: Pukyong National University (B.S.)
Y. J. Yoo: Korea Instituted of Industrial Technology (Senior Researcher)
※ 본 논문은 2024년 산업통상자원부 산업기술국제협력 (R&D)의 지원을 받아 수행하고 있는 3차원 비전 측정 기반, 200 마이크로미터 정밀도를 보장하는 정밀 생산공정 로봇 자동 제어 솔루션 개발 및 실증 [P0026191]과 국가과학기술연구회의의 선행융합연구사업의 재구성 가능한 초고속 저전력 광학 물리신경망 연산장치 개발을 위한 선행연구 [과제 고유번호 :CPS22081-100]의 지원을 받아 수행된 연구임.

ⓒ IEMEK J. Embed. Sys. Appl. 2024 Apr. 19(2) 83-90
ISSN : 1975-5066
http://dx.doi.org/10.14372/IEMEK.2024.19.2.83

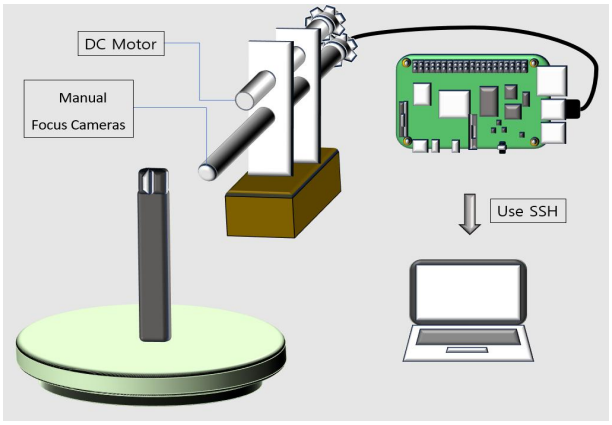

그림 1. 카메라를 이용한 데이터 수집 시스템
Fig. 1. System for data collection utilizing a microscope camera

을 구축하고 타 NeRF 모델들과 비교한다. 도출된 결과물을 통해 CNC 공구에 대한 이미지 학습 및 렌더링 과정까지의 소요 시간과, 렌더링된 이미지들의 품질을 비교하고 Instant NGP의 활용이 유효한지 살펴본다. 또한 해당 학습에 정밀하게 촬영된 이미지를 사용하기 위해, 단일 현미경 카메라를 사용하여 자동 초점 조절 기능을 구현하고, 촬영된 이미지들을 통해 데이터 셋을 구성한다.

구축된 시스템을 통해서는 촬영되지 않은 3D 장면을 예측 및 생성할 수 있다. 다각도에서의 장면 생성이 가능해짐에 따라, 카메라가 직접 접근하지 않은 영역에 대한 장면 확인이 가능하며, 카메라 및 객체를 이동시키는데 필요한 비용을 최소화할 수 있다. 또한, 외형 분석에 필요한 추가적인 view를 제공함으로써 CNC 공구의 외형 분석에 도움을 줄 수 있다.

본 연구는 3D 렌더링을 위해 고가의 장비나 특수한 카메라를 사용하지 않고, 저가형 현미경 카메라와 NeRF 기술만을 활용하여도 3D 렌더링 시스템을 구축하는데 준수한 성능을 달성할 수 있음을 서술한다. 이를 통해 경제적이면서도 효과적으로 CNC 공구의 외형적 상태를 분석하는데 적용할 수 있으며, 소규모의 제조업이나 예산이 제한된 환경에서도 유용할 것으로 사료된다.

Ⅱ. 자동 초점 시스템 및 데이터 셋 구축

1. NeRF 구현을 위한 데이터 수집 시스템

그림 1은 데이터 수집을 위한 시스템의 구성을 나타낸 것이다. 촬영해야 할 공구의 경우, 360도 회전을 반복하는 턴테이블의 중심에 위치시킨다. 현미경 카메라와 DC 모터는 라즈베리파이와 연결하였으며, 카메라의 초점 링 부분과 DC 모터에 기어를 설치한 후, DC 모터의 회전에 따라 카메라의 초점이 변경될 수 있도록 설치한다.

그림 2는 그림 1에서 제시한 구성을 통해, 자동 초점 기능을 구현하는 알고리즘을 나타낸 것이다. DC모터를 0.2초간 회전 시키고, 해당 초점 상태의 이미지를 촬영하는 과정

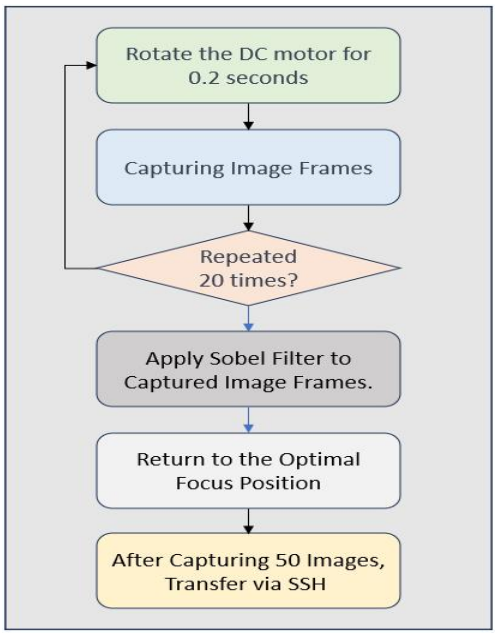

그림 2. 자동초점 기능 순서도
Fig. 2. Autofocus function flowchart

을 20회 반복한다.

다음, 촬영된 이미지 프레임들에 Sobel 필터를 적용하여 [7, 8] 가장 초점이 잘 맞는 이미지 프레임을 추출한 후, DC 모터를 구동시켜 해당 이미지 프레임을 촬영했을 때의 초점 상태로 돌아간다. 그 후, 턴테이블 위의 회전하는 공구에 대해 50장의 이미지를 촬영하고, 해당 이미지들을 SSH 통신을 통해 딥러닝 기능을 구현할 labtop 장비로 전송한다.

2. Sobel 에지 검출을 통한 자동초점 구현

영상 처리에서의 edge는 이미지 프레임의 픽셀 값이 명확히 바뀌는 부분을 의미한다. 이는 객체 간의 경계, 배경과 객체 간의 경계에서 주로 발생하므로, 해당 경계를 정확하게 도출해내었을 때를 초점이 가장 명확한 상태로, 경계가 부정확한 경우를 초점이 맞지 않는 상태로 간주할 수 있다.

본 연구에서는 Sobel 필터를 적용. 촬영된 20장의 이미지 프레임들에 대해 경계를 검출할 수 있도록 한다. Sobel 필터에서는 3x3 크기의 커널을 사용한다. 이는 각각

$$G_x = \begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix}, \quad G_y = \begin{pmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{pmatrix}. \tag{1}$$

와 같다. 해당 커널들을 컨볼루션을 통해 이미지 프레임 A에 적용하여, 픽셀별 gradient를 도출해낸다.

$$G = \sqrt{(G_x * A)^2 + (G_y * A)^2}\,. \tag{2}$$

해당 gradient 값을 기준으로 초점이 가장 이상적인 경우를 계산한다. 각 이미지 프레임 중, 픽셀 별 gradient 평균이 가장 높은 경우를 초점이 가장 잘 맞는 경우로 설정한다.

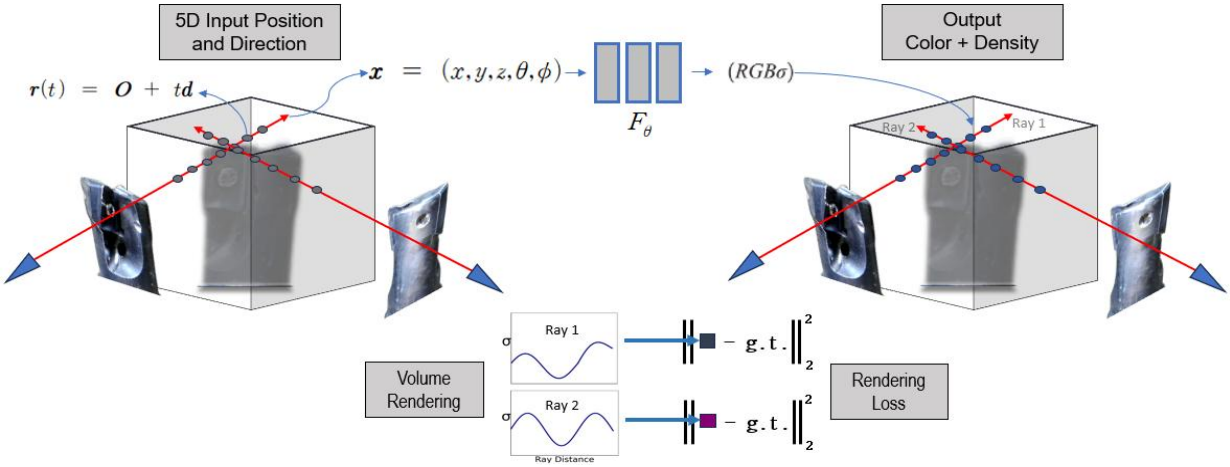

그림 3. NeRF 알고리즘의 주요 파이프라인
Fig. 3. The main pipeline of the NeRF algorithm.

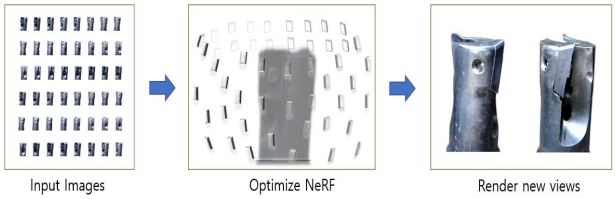

그림 4. NeRF 렌더링 과정 도식
Fig. 4. Schematic illustration of NeRF rendering process

Ⅲ. 3D 렌더링 알고리즘

1. NeRF

그림 3과 4는 각각 NeRF 학습 과정의 주요 파이프라인과 NeRF의 렌더링 과정을 도식한 것이다. NeRF는 3차원 좌표 $\boldsymbol{x} = (x,y,z)$ 와 해당 좌표에서 객체를 바라보는 시점 (θ,ϕ)를 입력받아, MLP 네트워크를 통해 색상 정보와 volume density를 추정한다. 이 때, 방향 (θ,ϕ)를 Cartesian 좌표계로 나타낸 단위 벡터 $\boldsymbol{d}$로 나타낸다. $\boldsymbol{O}$는 객체를 바라보는 시작점을 나타내며, $\boldsymbol{O}$시점에서 객체를 향해 조사하는 가상 레이저 광선 camera ray는 수식 (3)과 같다. 이는 시작점 $\boldsymbol{O}$에서 바라본 벡터 $\boldsymbol{d}$위의 한 점을 의미한다.

$$\boldsymbol{r}(t) = \boldsymbol{O} + t\boldsymbol{d}. \tag{3}$$

1.1 Volume Rendering

Volume Rendering 과정은, MLP를 통해 얻은 정보를 이용하여 한 픽셀에서 나타날 수 있는 예상 color값을 도출해내는 과정이다. 이는 수식 (4)과 같이 나타낼 수 있다.

$$C(\boldsymbol{r}) = \int_{t_n}^{t_f} T(t)\sigma(\boldsymbol{r}(t))\boldsymbol{c}(\boldsymbol{r}(t),\boldsymbol{d})dt, \tag{4}$$

$$T(t) = \exp\left(-\int_{tn}^{t}\sigma(\boldsymbol{r}(s))ds\right). \tag{5}$$

t_n과 t_f 렌더링할 3D 공간의 깊이를 나타내며, $\sigma(\boldsymbol{r}(t))$와 $\boldsymbol{c}(\boldsymbol{r}(t),\boldsymbol{d})$, $T(t)$는 각각 MLP에서 도출된 density, color값, transmittance를 의미한다. 이때, $T(t)$는 density와 연관되어, density가 1에 가까울수록 투과가 잘 이뤄지지 않음을 나타낸다.

ray가 접촉한 모든 점에 대해 $C(\boldsymbol{r})$의 계산이 이뤄질 경우, 불필요한 픽셀도 계산에 포함되어 성능이 제한된다. 때문에, t_n과 t_f사이를 N개의 구간으로 나누고, 해당 구간에서 하나의 샘플을 무작위로 추출하는 stratified sampling 접근 방식을 사용하여 샘플링 시점 t_i를 얻는다.

$$t_i \sim u[t_n + \frac{i-1}{N}(t_f - t_n),\ tn\ +\ \frac{i}{N}(t_f - t_i)], \tag{6}$$

$$\hat{C}(\boldsymbol{r}) = \sum_{i=1}^{N} T_i(1-\exp(-\sigma_i\delta_i))\boldsymbol{c}_i, \tag{7}$$

$$T_i = \exp\left(-\sum_{j=1}^{i-1}\sigma_i\delta_i\right). \tag{8}$$

이는 수식 (6)와 같이 표현할 수 있으며, 수식 (4)에 적용. 실제 도출되는 기대 RGB값은 수식 (7)과 같다. 해당 결과값을 바탕으로, 렌더링 과정에서 도출된 값과 실제 값의 오차를 사용하여 backpropagation을 수행한다.

1.2 Positional Encoding

(x,y,z,θ,ϕ) 값만을 통해 학습을 수행할 경우, 저주파 영역에 편향되어, 고주파 영역에서의 렌더링 성능이 떨어진다. 해당 문제점을 해결하기 위해, 고주파 함수를 사용하여 입력 데이터를 다차원 데이터로 변환한다. 입력에 대한 정보를 늘려, 네트워크의 입력 값으로 사용하며, 해당 매핑 함수 γ는 수식 (9)와 같이 나타낼 수 있다..

수식에서, L은 hyperparameter로, L이 클수록 더 높은 차원의 값이 네트워크로 입력된다. γ은 입력 $\boldsymbol{x} = (x,y,z)$와 camera ray 단위벡터 $\boldsymbol{d}$의 원소 x,y,z를 모두 사용하여, -1부터 1 사이 값으로 정규화한다.

$$\gamma(p) = (\sin(2^0\pi p), \cos(2^0\pi p),\ \ldots\ , \sin(2^{L-1}\pi p), \cos(2^{L-1}\pi p)). \tag{9}$$

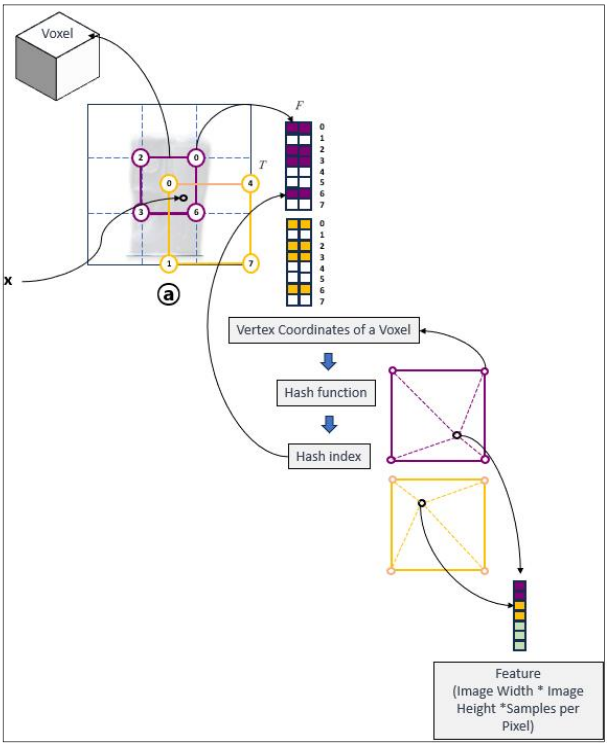

그림 5. Multi-resolution Hash Encoding 방식 도식
Fig. 5. Diagram of Multi-resolution Hash Encoding method

1.3 Hierarchical volume sampling

1.1 절에서 다룬, 한 개의 ray에서 random sampling을 통해 도출된 N개의 포인트를 활용하는 coarse 네트워크를 완성하였다. 해당 방식은 빈 공간과 가려지는 부분에 대한 샘플 포인트를 포함하기 때문에 좋은 결과를 도출하기 어렵다. NeRF에서는 렌더링 성능을 높이기 위해, fine 네트워크를 추가적으로 최적화한다. 해당 신경망에서는 coarse 네트워크에서의 출력을 바탕으로, ray위 물체 관련 정보를 다량 담고 있는 N_f개의 샘플들을 사용하여 학습한다. 이 때, coarse에 입력되었던 샘플들을 N_c라 하면, fine 네트워크에는 N_c+N_f개의 데이터가 입력으로 모두 포함된다. 해당 과정을 통해 도출된 loss는 coarse 네트워크에서 예측된 RGB $\hat{C}_c(\boldsymbol{r})$, fine 네트워크에서 예측된 RGB $\hat{C}_f(\boldsymbol{r})$, 실제 RGB $C(\boldsymbol{r})$를 통해 수식 (10)과 같이 나타낼 수 있다. 수식에서 R은 각 batch에서의 ray 집합을 의미한다.

$$Loss = \sum_{r\in R}[\,||\hat{C}_c(\boldsymbol{r})-C(\boldsymbol{r})||_2^2 + ||\hat{C}_f(\boldsymbol{r})-C(\boldsymbol{r})||_2^2]. \quad (10)$$

2. Instant NGP

Positional Encoding 방법 대신 Multi-resolution Hash Encoding을 적용하여, 비약적으로 학습 시간을 단축시킨 모델이다.

객체를 바라보는 지점으로부터 조사하는 가상 ray에는 다수의 샘플 포인트가 위치한다. 이는 그림 5의 ⓐ와 같이 나타낼 수 있다. 샘플 포인트는 3d 상에 위치한 정육면체의 voxel으로 감싸져 있으며, 해당 voxel의 좌표를 hash function에 입력한다. 이 때의 hash function은 수식 (11)과 같다. $h(x)$는 hash index, T는 mod 연산으로 인한 Hash index의 크기 제한을 의미한다. π는 mod 연산 시 나머지가 생기는 것을 방지하기 위해, 매우 큰 소수 값을 사용한다.

$$h(x) = \left(\left(\bigoplus_{i=1}^{d} x_i\pi_i\right)\right) \bmod T. \quad (11)$$

각 ray위의 voxel 꼭지점마다 계산된 hash index를 통해 매칭되는 2D feature를 모두 사용하여, 샘플 포인트와 voxel 꼭지점 간의 거리를 가중치로 두어 linear intepolation 연산으로 신경망에 입력할 2D feature를 생성한다. 이 때 한 level에서의 ray에서 생성할 수 있는 feature는 이미지의 가로와 세로 픽셀의 크기, ray위 샘플 포인트 수의 곱만큼 나타난다. 16개의 level의 다른 사이즈로 구성된 voxel을 사용하여 feature를 도출하고, 모두 concatenation 하여, Multi-resolution Hash Encoding 과정을 완료한다.

또한, Multi-resolution Hash Encoding 방식의 적용 외에도 Cuda 및 C++ 기반의 Tiny-Cuda-NN 라이브러리와, 신경망의 각 layer의 차원 수를 감소시킨 simple MLP를 사용하여 학습 및 렌더링 시간을 줄였다. 서술한 방식들의 적용으로 뛰어난 메모리 활용도를 달성하였으며, 짧은 시간의 학습만으로도 좋은 결과물을 생성할 수 있다.

3. Mip-NeRF

Mip-NeRF [9]는 ray를 사용하는 방식 대신, anti-aliased conical frustums (원뿔형 절두체)을 사용하여 aliasing 감소와 디테일한 표현의 향상을 갖춘 모델이다.

그림 6의 점으로 나타낸 부분은 기존 NeRF 모델의 point 단위 샘플링을 나타낸다. 이미지의 픽셀에 조사한 가상 ray의 샘플링 좌표를 지정하여 동작하며, 해당 샘플링 방법을 사용할 경우, 객체와 서로 다른 거리와 위치에서 촬영한 이미지들을 입력으로 넣었을 때 각 ray에서 보는 볼륨의 모양과 크기가 무시되어 학습 시 aliasing이 발생한다.

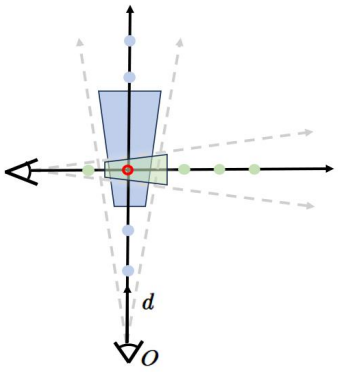

그림 6. NeRF와 Mip-NeRF에서의 sampling 방법
Fig. 6. Sampling method in Mip-NeRF

Mip-NeRF에서는 위 현상을 개선하기 위해 ray형태의 casting 방식이 아닌, 원뿔 형태를 casting하는 방식을 갖는다. 이는 ray위 sample에서 교차되는 모든 것을 렌더링하는 방식과 달리, 원뿔형 좌절 위 표시되는 정보를 평균화하여 카메라와 객체 간의 거리를 반영한 결과를 렌더링할 수 있다. 그림 6의 녹색으로 표현된 절두체의 경우, 카메라와 샘플 구간의 거리가 가깝기 때문에 평균화할 영역의 크기가 적은 반면, 청색으로 표시된 구간은 카메라와 샘플간의 거리가 멀기 때문에, 평균화할 영역이 넓은 것을 확인할 수 있다. 해당 거리별 평균화 방식을 통해, 카메라와 객체 간 거리 정보를 반영하여 렌더링할 수 있게 하고, aliasing 현상을 막을 수 있게 한다.

또한, 자체적으로 객체와의 거리를 학습하기 때문에, coarse 네트워크와 fine 네트워크를 사용하지 않고 단일 네트워크만을 통해 학습을 진행할 수 있다. 이를 통해, 학습 및 렌더링 속도가 향상되었으며, 기존 NeRF 대비 절반 크기의 모델을 갖는다.

4. DS-NeRF (Depth-supervised NeRF)

DS-NeRF [10]는 SFM (structure-from-motion) 기반의 Colmap 소프트웨어를 통해 이미지가 3D 공간상 어느 위치에 존재하는지에 대한 3D point clouds 정보를 얻고, depth를 추출, loss에 반영함으로써 개선된 렌더링 결과물을 얻어낸다.

DS-NeRF에는 기존 NeRF에서 사용하는 color 값에 대한 loss 외에 추가적으로 Ray Distribution Loss L_D를 사용하며, 이는 수식 (12)와 같다. 이때, $h(t)$는 transmittance $T(t)$와 volume density $\sigma(t)$를 통해, 시점 t에서부터 ray의 투과가 불가능함을 나타내는 확률분포이며 수식 (13)와 같다. 또한, x_i는 j번째 이미지에 대한 i번째 3d point cloud 정보를 나타내며, $\boldsymbol{D}_{ij}$는 j번째 이미지에서 i번째 포인트에 대해 계산한 깊이 정보를 의미한다.

$$L_D \approx E_{x_i \in X_j} \sum_k \log h_k \exp\left(-\frac{(t_k - \boldsymbol{D}_{ij})^2}{2\widehat{\sigma_i^2}}\right) \triangle t_k. \tag{12}$$

$$h(t) = T(t)\sigma(t). \tag{13}$$

구해진 L_D 값은, color loss L_C와 합쳐져 DS-NeRF의 최종적인 training loss가 된다. 이는 수식 (14)와 같으며, λ_D는 color loss와 Ray Distribution Loss의 균형을 조절하기 위한 hyperparameter이다.

$$L = L_C + \lambda_D L_D. \tag{14}$$

Ⅳ. 실험 및 결과

1. 자동초점 알고리즘 평가

그림 7은 수동초점 카메라로 촬영된 20장의 이미지 프레임 (Size: 640x480)에 대해 340x380 사이즈의 이미지로 crop 한 후, 픽셀별 gradient 변화를 나타낸 것이다. 19번째로 촬영된 이미지 프레임에서 가장 높은 픽셀 별 gradient를 도출해냈음을 확인할 수 있고, 해당 결과는 그림 8을 통해 확인할 수 있다.

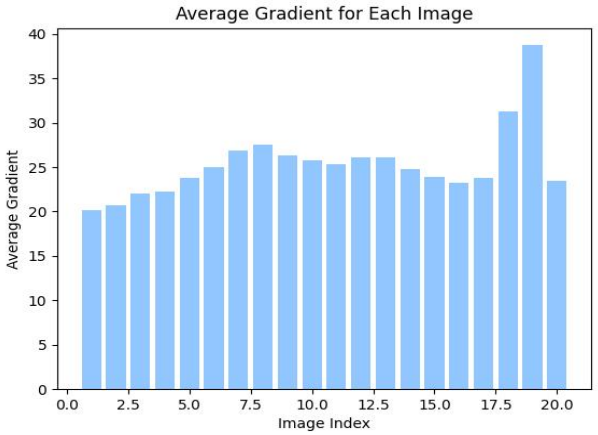

그림 7. 촬영된 이미지 프레임 별 Gradient 변화
Fig. 7. The gradient variations across captured image frames

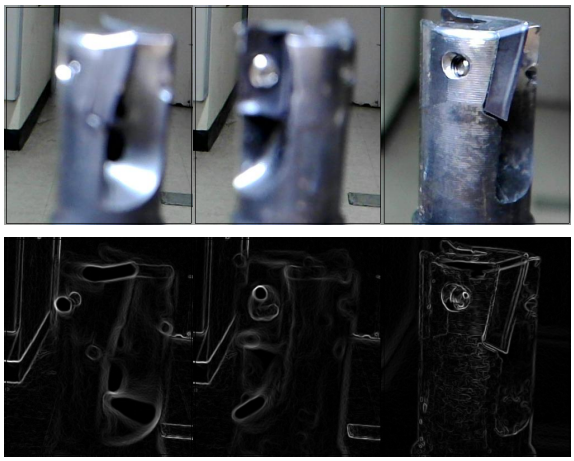

그림 8. 촬영된 6, 12, 19번 째 이미지 및 Sobel 엣지 검출 결과
Fig. 8. The 6th, 12th, and 19th captured images along with the results of Sobel edge detection

2. 학습용 Dataset 구축을 위한 배경 제거 알고리즘

CNC 공구만을 대상으로 3D 렌더링을 진행하기 위해, 배경제거 알고리즘을 적용한다. 해당 과정에는 rembg 패키지를 사용하며, 이는 U2Net (U-Square-Net)을 통한 segmentation 과정을 수행해주는 라이브러리이다.

초점이 정확하게 도출된 상태에서, 턴테이블 위 회전하는 객체를 50여장 촬영한다. 다음, 촬영된 이미지에 대해 배경제거를 수행한다. NeRF 모델들의 테스트 결과, 제거된 배경 외 추가적인 검정 배경을 더해줬을 때 좋은 결과를 보여주었다. 때문에, 가로 180, 세로 100 크기만큼의 0 값을 갖는 픽셀을 더하여 학습을 진행한다. 해당 결과는 그림 9와 같다. 해당 과정을 거쳐 생성된 50 여장의 이미지들을 NeRF 모델 학습에 사용한다.

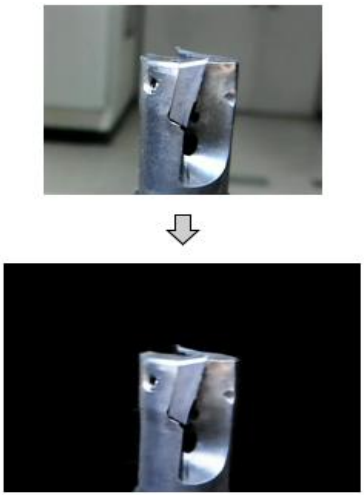

그림 9. 배경제거 알고리즘 적용 결과
Fig. 9. Background removal algorithm application results

3. 실험 결과 평가 지표

렌더링된 이미지의 품질 평가는 PSNR (Peak Signal-to Noise Ratio) [11], SSIM (Structural Similarity Index Map) [12]와 LPIPS (Learned Perceptual Image Patch Similarity) [13]를 통해 수행한다.

3.1 PSNR

PSNR의 경우 생성된 이미지에 대한 손실 정보를 나타내기 위해 사용하며., 이는 수식 (15)와 같다. MAX는 픽셀 값의 최대값을 나타내며, MSE는 각 픽셀 간의 차이를 제곱한 후, 평균을 내는 Mean Squared Error를 의미한다.

$$PSNR = 10\log\frac{MAX^2}{MSE}. \tag{15}$$

3.2 SSIM

SSIM는 PSNR과 달리, 단순 수치적 에러가 아닌 인간이 체감하는 이미지의 품질을 나타내기 위해 사용되는 평가법이다. 이미지간의 휘도, 대비, 구조를 평가하여 계산되며, 이는 수식 (16)과 같다.

$$SSIM(x,y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)}. \tag{16}$$

이 때, μ_x와μ_y는 각각 이미지 픽셀 값 평균을, σ_x와σ_y는 픽셀 값의 표준 편차를 의미한다. σ_{xy}는 이미지 x, y의 픽셀 값 간의 공분산을, C_1은 6.5025, C_2는 58.5225를 의미한다.

3.3 LPIPS

LPIPS는 AlexNet, VGG, SqueezeNet을 이용하여, 사람의 인지적 특성과 흡사하게 이미지의 유사함을 평가하는 모델이다. ImageNet dataset으로 학습된 network의 중간 layer feature를 통해, 두 feature가 유사한지를 측정한다.

입력 이미지 x, x_0에 대한 LPIPS는 수식 (17)과 같으며,

표 1. NeRF 모델 학습용 컴퓨팅 환경
Table 1. Computing environment for training NeRF models

CPU	AMD Ryzen 7 5800H
GPU	NVIDIA RTX 3070
RAM	DDR4 16.0GB
Cuda Version	11.7
Cudnn Version	8.8.1
Nvidia Driver	525.147.05

표 2. NeRF 모델의 렌더링 결과비교
Table 2. Comparison of rendering results in NeRF models

Nerf Models	Evaluation Metrics				
	Time (approximately)	PSNR	SSIM	LPIPS	Peak VRAM
NeRF	2 hours	29.560	0.933	0.073	6540 MB
NeRF-Long	6 hours	30.395	0.944	0.058	
Mip-NeRF	1 hour	27.770	0.920	0.093	7740 MB
Mip-NeRF-Long	2 hours	29.052	0.935	0.069	
DS-NeRF	1 hour	28.945	0.930	0.087	5288 MB
DS-NeRF-Long	3 hours	30.340	0.943	0.068	
Instant NGP	1 minute	27.187	0.927	0.076	2168 MB

l은 layer를, y, y_0는 채널 차원에서 unit-normalize한 feature vector를, w는 scale factor를, h와 w는 height와 width를 나타낸다. PSNR과 SSIM과 달리, LPIPS는 원본 이미지와 생성된 이미지의 특징 벡터간 거리를 측정하므로, 수치가 낮을수록 우수하다.

$$LPIPS(x,x_0) = \sum_l \frac{1}{H_l W_l}\sum_{h,w} \| w_l \odot (\widehat{y^l_{hw}} - \widehat{y^l_{0hw}}) \|_2^2. \tag{17}$$

4. 적용된 NeRF 알고리즘의 성능 비교

NeRF 모델 별로 차지하는 VRAM 용량이 서로 상이하다. 단일 NVIDIA RTX3070 GPU만을 사용하여 학습하기 위해서 Instant NGP를 제외한 NeRF, Mip-NeRF, DS-NeRF의 경우 batch size를 128, iteration을 100k번 및 300k번 진행한다. Instant NGP의 경우, batch size를 262144로, iteration을 4k번 진행한다. NeRF 모델의 학습을 위한 컴퓨팅 환경 조성은 표 1과 같으며, 촬영된 test dataset 7장에 대한 평가 결과는 표 2와 같다. 기재된 NeRF-Long, Mip-NeRF-Long, DS-NeRF-Long은 각각 300k번의 iteration을 진행하였을 때를 나타낸다. Time은 학습 시간과 test dataset에 대한 렌더링 시간의 합산을, Peak VRAM은 학습 및 렌더링 과정에서 VRAM 사용량이 가장 높았을 경우를 나타낸다.

NeRF의 300k번 학습 및 렌더링 과정에 6시간 가량이 소요되었으며. Mip-NeRF와 DS-NeRF 또한 2~3시간 가량의 시간이 요구되었다. Instant NGP의 경우, 약 1분의 학습 시간 대비 뛰어난 품질의 렌더링 결과물을 도출해냈다. 100k

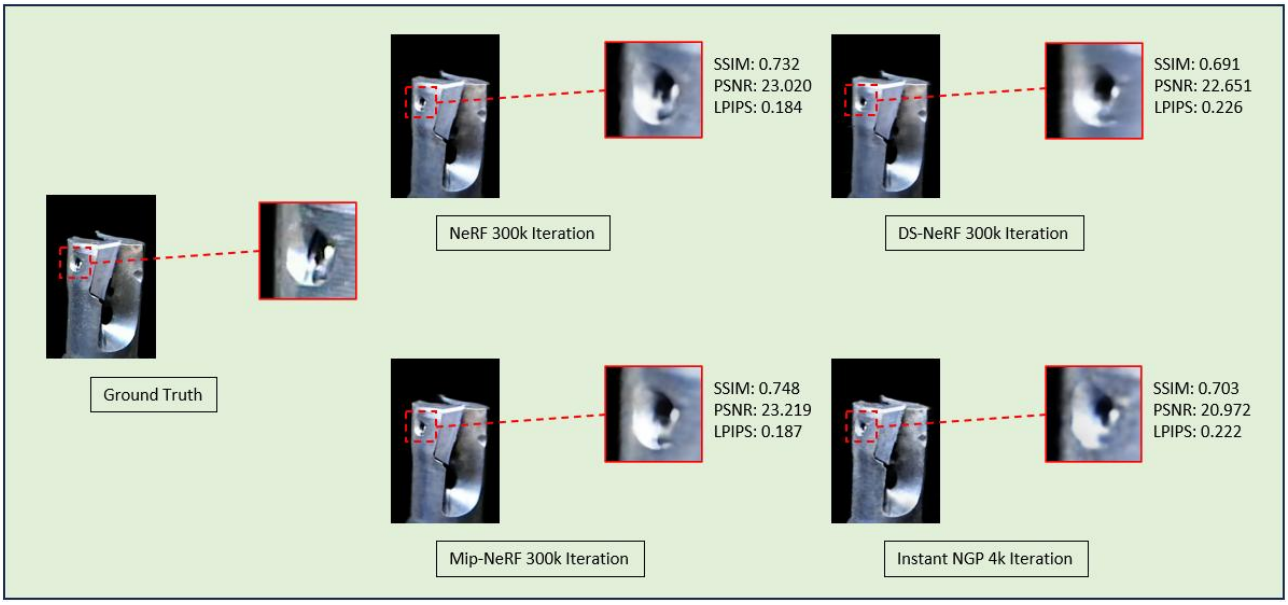

그림 10. NeRF 모델별 렌더링 결과 비교
Fig. 10. Comparison of Rendering Results for NeRF Models

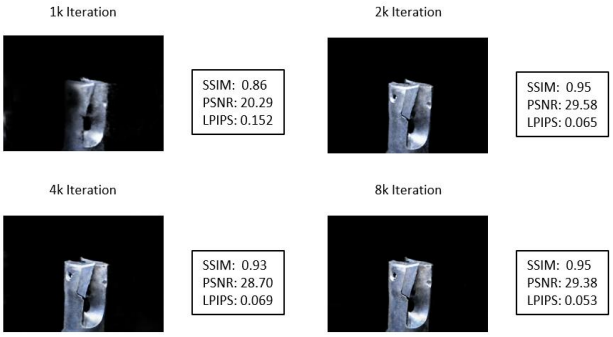

그림 11. Iteration별 Instant NGP 학습 결과물
Fig. 11. Iteration-wise Instant NGP Training Results

번의 iteration을 진행한 Mip-NeRF 보다 SSIM 수치가 더 뛰어난 것을 확인할 수 있으며, LPIPS 수치에 대해, 100k번 iteration을 진행한 Mip-NeRF와 DS-NeRF보다 우수한 결과를 나타낸다. 그림 10과 같이 국소적인 부분을 평가했을 때는 DS-NeRF보다 SSIM, LPIPS 수치가 더 뛰어난 것을 볼 수 있으며, 그림 11의 iteration별 학습 결과를 통해, 빠르게 학습 결과물에 수렴하는 것을 확인할 수 있다. 또한 타 NeRF 모델들에 비해 batch size 증가 대비 메모리 사용량이 상당히 적었기 때문에, 높은 batch size를 적용하여 학습을 진행할 수 있었으며 최대 VRAM 사용량 또한 2168MB로 가장 적다. 서술된 모델 중, Instant NGP가 가장 메모리 효율이 좋은 것을 볼 수 있다.

Ⅳ. 결 론

본 연구는 공구 상태 진단을 위한 3D 렌더링 시스템의 구축에 Instant NGP를 적용하였으며, NeRF, Mip-NeRF, DS-NeRF 모델들과의 성능을 비교해보았다. 객체의 사실적인 장면을 복원해낼 수 있었으며, Instant NGP를 적용한 시스템을 통해 경제적이고 효과적인 3D 렌더링 시스템을 구축할 수 있음을 보여준다. 렌더링 과정을 통해 추출된 이미지 및 영상에 image segmentation 및 object detection, visual anomaly detection [14]등의 알고리즘을 적용하여 공구의 마모 상태 판단에 활용할 수 있다.

하지만 이미지 수집과 학습 및 3D 렌더링 과정까지 총 소요되는 3분 가량의 시간은 실시간 렌더링 시스템 구축에는 어려움이 있음을 나타낸다. 또한 학습 iteration을 대폭 증가시켜도 화질 개선이 뚜렷하게 나타나지 않는 한계를 보였다. 매년 NeRF 모델의 학습 및 렌더링 속도 문제에 대해 많은 개선을 이룬 연구들이 계속해서 이뤄지고 있으며, 앞서 서술한 한계점을 개선하기 위해 다양한 NeRF 모델들을 추가로 적용하여 향상된 시스템을 개발해나갈 계획이다.

References

[1] S. J. Kim, J. H. Lyu, S. H. Ahn, B. D. Youn, "Auto-encoder Based Incipient Anomaly Detection of CNC Milling Tool Considering Machining Tool Path," Proceeding of KSPE, pp. 377-377, 2023. (in Korean)

[2] J. D. Choi, M. Y. Kim, B. H. Kim, "Dynamic 3D Worker Pose Registration for Safety Monitoring in Manufacturing Environment based on Multi-domain Vision System," IEMEK, Vol. 18, No. 6. pp. 303-310, 2023. (in Korean)

[3] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, R. Ng, "Nerf: Representing Scenes as Neural Radiance Fields for View Synthesis," Communications of the

ACM, Vol. 65, No. 1, pp. 99-106, 2021.
[4] W. S. Kim, J. H. Joe, D. S. Kim, D. G. Kim, S. M. Hong, "Development of Auto-spray system to improve the quality of 3D Scanning Quality," Journal of the Korea Academia-Industrial Cooperation Society, Vol. 17, No. 4, pp. 100-105, 2016. (in Korean)
[5] D. Y. Um, J. H. Kim, "The Reflected Property Analysis of 3D Laser Scanning System as Object Surface Materials," Journal of the Korean Society of Surveying, Geodesy, Photogrammetry and Cartography, Vol. 27, No. 3, pp. 347-356, 2009. (in Korean)
[6] T. M¨uller, A. Evans, C. Schied, A. Keller, "Instant Neural Graphics Primitives with a Multiresolution Hash Encoding," ACM Transactions on Graphics (ToG), Vol. 41, No. 4, pp. 1-15, 2022.
[7] N. Kanopoulos, N. Vasanthavada, R. L. Baker, "Design of an Image Edge Detection Filter Using the Sobel Operator," IEEE Journal of Solid-state Circuits, Vol 23, No. 2, pp. 358 - 367, 1988.
[8] J. H. Jeon, I. H, Yoon, J. H. Lee, J. K. Paik, "Subject Region-Based Auto-Focusing Algorithm Using Noise Robust Focus Measure," Journal of the Institute of Electronics Engineers of Korea, Vol. 48 No. 2, pp. 80-87, 2011. (in Korean)
[9] J. T. Barron, B. Mildenhall, M. Tancik, P. Hedman, R. Martin-Brualla, P. P. Srinivasan, "Mip-nerf: A Multiscale Representation for Anti-aliasing Neural Radiance Fields," In Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 5855 - 5864, 2021.
[10] K. Deng, A. Liu, J. Zhu, D. Ramanan, "Depth-supervised Nerf: Fewer Views and Faster Rraining for Free," In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 12882 - 12891, 2022.
[11] A. Horé, D. Ziou, "Image Quality Metrics: PSNR vs. SSIM." 20th International Conference on Pattern Recognition, pp. 2366-2369, 2010.
[12] Z. Wang, A. C. Bovik, H. R. Sheikh, E. P. Simoncelli, "Image Quality Assessment: From Error Visibility to Structural Similarity," IEEE Transactions on Image Processing, Vol. 13, No. 4, pp. 600-612, 2004.
[13] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, O. Wang, "The Unreasonable Effectiveness of Deep Features as a Perceptual Metric," In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 586 - 595, 2018.
[14] J. Yang, R. Xu, Z. Qi, Y. Shi, "Visual Anomaly Detection for Images: a Systematic Survey," Procedia Computer Science, Vol. 199, No. 61, pp. 471-478, 2022.

TaeYeong Jung (정 태 영)

2024 Information and Communication Engineering from Pukyong National University (B.S.)

Career:
2023 Korea Instituted of Industrial Technology, Undergraduate researcher
Field of Interests: Embedded System & Deep Learning Application
Email: jung75688@naver.com

YOUNGJUN Yoo (유 영 준)

2005~2009 Electrical Engneering from Pohang university of science and technology (B.S.)
2009~2014 Electrical Engneering from Pohang university of science and technology (Ph.D.)

Career:
2014~2019 Senior researcher, Samsung Heavy Industries
2019~Senior researcher, Korea Institute of Industrial Technology
Field of Interests: Inspection equipment using AI and IoT equipment, data analysis, and smart manufacturing
Email: youdalj@kitech.re.kr