생성형 AI를 이용한 시각적 질감 기반 하이트맵 생성 연구

이호준 전석희

경희대학교 컴퓨터 공학과

hojun313@khu.ac.kr jeon@khu.ac.kr

Generative AI-Driven Algorithm for Adaptive Haptic Feedback

HoJun Lee SeokHee Jeon

School of Computer Science and Engineering, Kyung Hee University

요 약

본 연구는 생성형 인공지능(AI)을 이용하여 시각적 질감 이미지로부터 해당 표면의 3차원적 굴곡 정보를 담은 하이트맵(height map)을 생성하는 알고리즘 개발을 목표로 한다. 디지털 콘텐츠의 몰입감을 높이는 데 있어 현실적인 촉감 피드백은 중요하지만, 다양한 질감을 섬세하게 표현하는 데는 한계가 있었다. 이러한 문제를 해결하기 위해, 본 연구에서는 Stanford-Gelsight 데이터셋으로부터 RGB 질감 이미지와 그에 대응하는 하이트맵 쌍을 구축하는 데이터 파이프라인을 설계하였다. 생성 모델로는 사전 학습된 EfficientNet-B7을 인코더로 사용하는 U-Net 아키텍처를 채택하였으며, 픽셀 단위의 구조적 유사성을 위한 L1 손실과 인간의 시지각적 특성을 반영하는 LPIPS 손실을 조합하여 모델을 학습시켰다.

학습에 사용되지 않은 9개의 재질로 구성된 테스트 세트로 모델의 성능을 평가한 결과, 생성된 하이트맵은 원본의 통계적 질감 특징을 매우 높은 유사도(GLCM Cosine Similarity: 0.9958)로 재현하였다. 하지만 생성된 질감의 대비(Contrast) 값이 원본보다 약 두 배 가까이 높게 나타나, 특징을 과장하는 경향을 보였다. 전체적인 구조적 유사도(SSIM)는 0.5745, 지각적 유사도(LPIPS)는 0.4727로 보통 수준의 성능을 보였으며, 특히 엣지(경계선)의 정확도를 나타내는 F1 점수는 0.0164로 매우 낮아 정밀한 경계선 생성에 명확한 한계가 있음을 확인하였다.

본 연구는 Gelsight 데이터를 이용한 하이트맵 생성 파이프라인을 구축하고, U-Net 기반 모델의 성능 특성을 다각도로 분석하여 통계적 질감 재현이라는 강점과 특징 과장 및 엣지 표현력 부족이라는 한계를 규명했다는 점에서 의의를 가진다. 이 결과는 향후 손실 함수 고도화 및 아키텍처 개선을 통해 더욱 현실적인 촉감 생성 기술로 발전할 수 있는 기반을 제공할 것이다.

1 서 론

최근 인공지능 기술의 발전과 함께 다양한 감각 피드백을 제공하는 인터페이스에 대한 연구가 활발히 진행되고 있다. 그중 촉감은 사용자와 디지털 콘텐츠 간의 몰입도를 높이고 현실감을 더하는 중요한 요소로 인식되고 있다. 그러나 기존의 촉감 피드백 기술은 다양한 질감을 섬세하게 표현하는 데 한계가 있으며, 특히 복잡하고 사실적인 질감을 효과적으로 생성하고 전달하기 위한 새로운 접근 방식이 요구되고 있다.

본 연구는 생성형 인공지능을 활용하여 사용자가 시각적으로 인지가능한 하이트맵을 생성하여, 이를 통해 현실적인 촉감 피드백을 제공할 수 있는 기반을 마련하고자 한다. 구체적으로, 질감 이미지 정보를 입력받아 해당 질감 표면의 미세한 굴곡 형태를 2D 지오메트릭 형태, 즉 하이트맵(height map)으로 생성하고, 이를 통해 현실적인 촉감 피드백을 제공할 수 있는 기반을 마련하고자 한다. 본 보고서는 이러한 연구 목표를 달성하기 위한 과정, 실험 결과, 그리고 향후 연구 방향을 상세히 기술한다.

2 관련 연구

본 연구는 생성형 AI를 활용하여 시각적 질감 정보로부터 촉감 데이터(하이트맵)를 생성하는 것을 목표로 한다. 이러한 목표를 달성하기 위해, 본 장에서는 이미지-촉감 변환 연구, 하이트맵 생성 기술, 그리고 관련 연구에 활용된 주요 질감 데이터셋에 대한 기존 연구들을 검토하여 본 연구의 기술적 배

경과 차별점을 명확히 하고자 한다.

2.1 생성형 AI를 활용한 이미지-촉감 변환 연구

2.1.1 전통적인 접근법과 한계

기존의 이미지-촉감 변환 연구들은 주로 규칙 기반의 컴퓨터 비전 기법에 의존했으나, 다양한 특징을 포착하는 데 한계가 있었다. 특히 KAIST의 연구(2012) [8]에서 보여준 바와 같이, 전통적인 센서 기반 접근법은 다양한 촉각 수용기의 복합적 정보를 충분히 모사하지 못하는 문제점을 드러냈다.

2.1.2 생성형 AI의 등장과 새로운 가능

최근 몇 년간 생성형 인공지능(Generative AI) 기술은 이미지 생성 분야에서 괄목할 만한 발전을 이루었다. 특히 2022년 이후 확산 모델(Diffusion Model)의 등장으로 Stable Diffusion, DALL-E 2와 같은 모델들이 텍스트 설명으로부터 고품질 이미지를 생성하는 능력을 보여주었다. 이러한 발전은 촉감 데이터 생성 분야에도 새로운 탐색 방향을 제시했으나, 이들 모델은 주로 일반적인 이미지 생성에 초점을 맞추고 있어, 촉감 정보의 핵심인 미세한 물리적 특성을 정확히 반영하고 제어하는 데는 여전히 연구가 필요한 상황이다. 본 연구는 이러한 대규모 생성 모델과는 다른 접근 방식으로, 특정 도메인(질감 이미지에서 하이트맵으로의 변환)에 특화된 모델 구조를 활용하고자 한다.

2.2 하이트맵 생성 기술 동향

2.2.1 딥러닝 기반 생성 모델의 활용

질감의 3차원적 표면 정보를 2차원 하이트맵으로 생성하는 것은 일종의 이미지 대 이미지 변환(Image-to-Image Translation) 문제로 볼 수 있다. 이러한 문제 해결을 위해 U-Net [6]과 같은 컨볼루션 신경망(CNN) 기반의 인코더-디코더 아키텍처가 효과적으로 활용되어 왔다. U-Net은 특히 의료 이미지 분할 분야에서 뛰어난 성능을 보이며 등장했지만, 입력 이미지의 공간적 정보를 잘 보존하면서 원하는 형태로 변환하는 능력 덕분에 다양한 이미지 변환 문제에 응용되고 있다. U-Net의 핵심적인 특징인 스킵 연결(skip-connections)은 인코더의 저수준 특징(low-level features)을 디코더의 상응하는 레이어에 직접 전달함으로써, 고해상도의 세밀한 출력 생성에 강점을 가진다.

본 연구에서는 이러한 U-Net 아키텍처의 장점을 활용하여 질감 이미지로부터 하이트맵을 생성한다. 특히, 다양한 인코더 백본(예: ResNet, EfficientNet)에 대해 사전 학습된 가중치를 활용함으로써, 제한된 데이터셋에서도 효율적인 학습과 높은 성능을 기대할 수 있다.

segmentation_models_pytorch와 같은 라이브러리는 이러한 사전 학습된 인코더를 U-Net과 손쉽게 결합하여 사용할 수 있는 환경을 제공한다.

2.3 실시간 촉각 인식 및 햅틱 피드백

한편, 생성된 촉감 정보를 사용자에게 효과적으로 전달하기 위한 연구도 활발히 진행 중이다. KAIST 연구팀이 개발한 인공 신경 촉각 감지 시스템은 나노입자 기반 복합 센서와 실제 신경 패턴 기반 신호 변환을 결합하여 인간의 촉각 인식 프로세스를 모방하는 데 성공했다. 이 시스템은 압력과 진동을 동시에 감지할 수 있지만, 복잡한 표면 질감의 국소적 특징을 분석하여 하이트맵과 같은 형태로 생성하는 연구와는 다소 거리가 있다. 또한, 크랙 기반 高민감 센서와 딥러닝 모델을 결합한 연구는 단일 센서로 복잡한 손가락 움직임을 측정할 수 있음을 보여주었으며, 전이 학습을 통해 적은 데이터로도 효과적인 학습이 가능했으나, 이는 정적 질감 분석보다는 동적 움직임 감지에 특화되어 있다.

2.4 요약 및 본 연구의 차별점

기존 생성형 AI 연구들은 주로 일반적인 시각적 이미지나 비디오 생성에 집중되어, 촉감 표현에 필요한 미세한 물리적 특성을 충분히

반영하지 못하는 경향이 있다. 게임 콘텐츠 생성 분야에서의 절차적 생성 연구(PCGRL 등)는 상호작용 가능한 콘텐츠 생성을 시도했지만, 실제 물리적 촉감 데이터와의 직접적인 연결은 부족하다. 또한, 고도화된 촉각 센싱 시스템 연구들은 실시간 감지에는 성공했지만, Gelsight와 같은 고해상도 촉각 영상 데이터를 분석하여 해당 표면의 하이트맵을 직접 생성하는 연구는 상대적으로 부족하다.

따라서 본 연구는 다음과 같은 차별점을 가진다:

2.4.1 Gelsight 동영상 데이터로부터 프레임을 추출하고, 이를 입력 RGB 이미지와 정답 하이트맵 쌍으로 구성하여 학습 데이터를 구축한다.

2.4.2 U-Net 기반의 이미지 대 이미지 변환 모델을 활용하여 질감 이미지로부터 직접적으로 하이트맵을 생성한다. 이 과정에서 사전 학습된 인코더를 사용하여 학습 효율을 높인다.

2.4.3 모델 학습 시, 픽셀 단위의 L1 손실과 함께 인간의 시지각적 특성을 고려한 LPIPS 손실을 조합하여 사용하여, 생성된 하이트맵이 수치적 정확성뿐만 아니라 시각적 현실성도 갖도록 유도한다.

2.4.4 생성된 하이트맵의 품질 평가는 단순히 전체적인 유사도뿐만 아니라, 그래디언트(Canny edge map 기반 유사도 및 GLCM 등)를 활용하여 질감의 국소적인 특징과 경계선 정보를 반영하는 통합적 접근법을 시도한다.

이러한 접근법은 기존 연구들이 충분히 다루지 않은, 시각적 질감 정보로부터 직접적으로 고품질의 하이트맵을 생성하고 평가하는 새로운 연구 방향을 제시할 수 있을 것으로 기대된다.

3 연구 방법론

본 장에서는 입력된 질감 이미지로부터 해당 표면의 미세한 굴곡 정보를 담고 있는 하이트맵(height map)을 생성하기 위해 제안하는 전체적인 방법론을 기술한다. 데이터셋 구축 과정부터 모델 아키텍처, 학습 전략, 그리고 결과 평가 지표 선정까지의 과정을 상세히 설명한다.

3.1 개요

본 연구는 U-Net 아키텍처 기반의 생성형 인공지능(Generative AI) 모델을 활용하여, 시각적 질감 정보를 담은 RGB 이미지를 입력받아 해당 질감의 3차원적 표면 형태를 나타내는 2차원 하이트맵을 출력하는 것을 목표로 한다. 생성된 하이트맵은 향후 촉감 피드백 장치를 통해 물리적인 질감으로 재현될 수 있는 기초 데이터를 제공한다.

3.2 입력 및 출력 데이터 정의

3.2.1 입력 데이터: 다양한 재질의 표면을 근접 촬영한 RGB 컬러 이미지이다. 각 이미지는 특정 재질 고유의 시각적 질감 패턴을 포함한다.

3.2.2 출력 데이터: 입력된 질감 이미지에 대응하는 2차원 단일 채널(grayscale) 하이트맵이다. 각 픽셀 값은 해당 위치의 상대적인 높낮이 정보를 나타낸다. 초기 연구 방향에서는 진동 파형(waveform) 생성을 고려하였으나, 힘, 방향, 속도 등 다양한 변인에 따른 일관된 질감 표현의 어려움으로 인해, 표면의 굴곡 자체를 직접적으로 표현하는 하이트맵 생성으로 목표를 구체화하였다.

3.3 굴곡 표현 방식: 하이트맵

질감 표면의 미세한 3차원적 굴곡을 효과적으로 표현하기 위해 다양한 방식을 검토하였다. 폴리곤 메쉬(Polygon Mesh) 방식은 복잡하고 잦은 굴곡을 표현하는 데 있어 데이터의 복잡도 및 처리 효율성 측면에서 한계가 있을 것으로 판단되었다. 반면, 2차원 하이트맵 방식은 각 픽셀의 명암 값을 통해 높낮이 정보를 직관적으로 표현할 수 있으며, 점군 데이터(Point Cloud Data, PCD) 등 다른 방식과 비교하여 질감의 섬세한 차이를 나타내고 생성 모델의 출력 형태로 다루기에 유리하다고 판단하여 최종적인 굴곡 표현 방식으로 선정

하였다.

3.4 활용 데이터셋

본 연구의 학습 및 평가에는 Stanford-Gelsight 데이터셋 [5]을 활용하였다. 이 데이터셋은 HaTT(Haptic Texture Toolkit) [7]에 포함된 재질에 대해 프로토타입 Gelsight 센서를 이용하여 수집된 동영상 형식의 질감 정보를 포함한다.

3.4.1 데이터셋 구조 및 특성

수집된 동영상 데이터는 각 재질에 대해 센서의 압력 변화에 따른 질감 정보의 동적인 변화를 담고 있어, 정지 이미지보다 풍부한 학습 데이터를 제공한다. 데이터 수집에 사용된 센서는 프로토타입의 Gelsight Mini [3]로 추정되며, 수집 영역 크기는 약 18.6mm x 14.3mm, 카메라 해상도는 8MP, 약 30FPS로 파악되었다.

3.4.2 데이터 준비 및 전처리 파이프라인

효율적인 모델 학습과 성능 향상을 위해 TextureHeightmapDataset 클래스를 정의하고 다음과 같은 데이터 준비 및 전처리 과정을 체계적으로 수행하였다.

3.4.2.1프레임 추출 및 쌍 구성: Stanford-Gelsight 데이터셋의 각 재질별 동영상으로부터 개별 프레임 이미지를 추출하였다. OpenCV 라이브러리를 활용하여 각 비디오를 초당 30프레임으로 샘플링하였다. _build_image_pairs 함수는 지정된 데이터 루트 경로 (data_root)에서 각 재질 폴더 내의 input_*.png (또는 다른 지원 형식) 파일과, 그 하위 output/조건폴더/heightmaps/ 내의 하이트맵 이미지들을 탐색하여 (입력 이미지, 하이트맵 이미지) 쌍을 구성한다.

3.4.2.2하이트맵 변환: 추출된 Gelsight 프레임 이미지(RGB)로부터 대응하는 3D 하이트맵 데이터를 생성하기 위해, GelSight Inc.에서 제공하는 공식 SDK 및 관련 오픈소스 코드(sdkdemo/gsrobotics) [1, 2]를 활용하였다. Gelsight 원본 이미지를 직접 학습에 사용하는 것보다, 이를 먼저 하이트맵으로 변환하여 정답 데이터로 사용하고 모델이 하이트맵을 직접 생성하도록 학습하는 방향이 학습 시간 단축 및 생성 결과물의 직관성 측면에서 더 적합하다고 판단하였다.

3.4.2.3이미지 정제 (마커 제거): 데이터 수집에 사용된 Gelsight 기기 프로토타입의 특성상 수집된 이미지에는 센서 표면에 인쇄된 검은색 마커(점)들이 존재한다. 이는 실제 질감 정보와 무관하므로 학습에 방해가 될 수 있어, 특정 색상 범위를 기준으로 마스크를 생성하고 OpenCV [14]의 인페인팅(Inpainting, cv2.inpaint) 기술을 적용하여 마커 영역을 주변 픽셀 정보로 복원하는 전처리 과정을 거쳤다.

3.4.2.4비접촉 프레임 제외: 동영상 촬영 시 Gelsight 센서가 재질 표면에 완전히 접촉하기 전의 초기 프레임들은 유효한 질감 정보를 포함하지 않는다. TextureHeightmapDataset 클래스 초기화 시 exclude_heightmap_indices_up_to 인자를 통해, 파일명 끝이 _00000부터 _00015 (예시)까지의 하이트맵 파일들을 학습 데이터에서 제외하였다.

3.4.2.5이미지 리사이즈 및 정규화: 모델 학습의 효율성 및 일반성을 고려하여 원본 영상(960x720 픽셀) 대신 256x256 픽셀 해상도를 최종 입력 및 출력 크기로 선정하였다. re_train.py의 transform_texture와 transform_heightmap에 정의된 바와 같이, 모든 입력 RGB 이미지와 출력 하이트맵(단일 채널)은 transforms.Resize((args.image_size, args.image_size))를 통해 지정된 크기로 리사이즈된 후 transforms.ToTensor()로 텐서 변환되고, 각 채널별로 평균 0.5, 표준편차 0.5로 정규화(transforms.Normalize)되어 -1에서 1 사이의 값을 갖도록 하였다.

3.5 생성 모델 아키텍처

본 연구에서는 이미지 대 이미지 변환 문제에 효과적인 것으로 알려진 U-Net [6] 기반의 아키텍처를 주요 생성 모델로 채택하였다.

3.5.1 기본 구조: U-Net: U-Net은 인코더-디코더 구조에 스킵 연결(skipconnections)을 추가하여 입력 이미지의 저수준 특징(low-level features)을 디코더의 상응하는 레이어에 직접 전달함으로써, 고해상도의 세밀한 출력을 생성하는 데 강점을 가진다. 본 연구에서는 segmentation_models_pytorch (SMP) [9] 라이브러리에서 제공하는 U-Net 구현체 (smp.Unet)를 활용하였다.

3.5.2 인코더 및 사전 학습: U-Net의 인코더로는 ImageNet 데이터셋으로 사전 학습된 다양한 컨볼루션 신경망 모델들을 사용하였다. encoder_name 인자를 통해 "resnet34", "resnet50" [10], "efficientnet-b7" [15] 등 다양한 인코더를 선택할 수 있도록 구현하였고 encoder_weights="imagenet"으로 설정하여 사전 학습된 가중치를 활용하였다. 이를 통해 대규모 데이터셋으로부터 학습된 풍부한 시각적 특징 추출 능력을 전이 학습(transfer learning) 형태로 활용하여 모델의 학습 효율성과 최종 성능을 높이고자 하였다.

3.6 학습 전략

모델 학습의 안정성과 성능 최적화를 위해 다음과 같은 학습 전략을 사용하였다.

3.6.1 손실 함수 (Loss Functions): 생성된 하이트맵과 실제 정답 하이트맵 간의 유사성을 측정하기 위해 다음과 같은 손실 함수들을 조합하여 사용하였다.

3.6.1.1L1 손실 (Mean Absolute Error, MAE): 픽셀 단위의 절대적인 차이를 최소화하여 전반적인 구조적 유사성을 확보하고자 하였다. torch.nn.L1Loss()를 사용하였으며 손실 함수는 다음과 같이 정의된다: $L_{L1} = ||G(x)-y||_1$ 여기서 G(x)는 생성된 하이트맵, y는 정답 하이트맵이다.

3.6.1.2지각 손실 (Perceptual Loss) - LPIPS: 질감 표현의 한계를 극복하고 사람의 지각 판단과 유사한 유사성을 측정하기 위해 LPIPS (Learned Perceptual Image Patch Similarity) [12]를 지각 손실로 도입하였다. lpips.LPIPS(net='alex', verbose=False)를 사용하였으며, AlexNet 기반의 사전 학습된 네트워크를 활용한다. 손실 함수는 다음과 같이 정의된다: $L_{LPIPS} = LPIPS(G(x), y)$

3.6.1.3최종 손실 함수: 생성자(모델)의 최종 손실 함수는 L1 손실과 LPIPS 손실의 가중합으로 구성된다. re_train.py의 train_fn 함수 내에서 다음과 같이 계산된다: $L_{total} = L_{L1} + \lambda_{LPIPS} * L_{LPIPS}$ 여기서 λ_{LPIPS}는 LPIPS 손실의 가중치를 나타내는 하이퍼파라미터로, re_train.py의 args.lambda_lpips (기본값 0.5)에 해당한다.

3.6.2 최적화 (Optimization)

모델의 가중치 업데이트에는 Adam 옵티마이저 [13]를 사용하였다. Adam은 학습률을 적응적으로 조절하고 모멘텀을 활용하여 효율적인 수렴을 돕는다. (지수 평균용 beta1: 0.9, 제곱 평균용 beta2: 0.999)

3.6.3 학습률 스케줄링 (Learning Rate Scheduling)

학습 과정 중 손실 값의 정체 현상을 극복하고 안정적인 수렴을 돕기 위해 ReduceLROnPlateau 학습률 스케줄러를 사용하였다. 이 스케줄러는 특정 에포크 동안 검증 손실(또는 학습 손실)의 개선이 없을 경우 학습률을 일정 비율로 감소시킨다. (patience: 5, factor: 0.5)

3.7 평가 지표(re_evaluate.py 기반)

생성된 하이트맵의 품질을 객관적으로 평가하기 위해 다음과 같은 정량적 평가를 병행하였다. re_evaluate.py 스크립트는 이러한 정량적 지표 계산을 지원한다.

3.7.1 LPIPS (Learned Perceptual Image Patch Similarity): 사람이 인지하는 이미지 간의 지각적 유사도를 평가한다.

lpips.LPIPS(net=args.lpips_net) (AlexNet 또는 VGGNet 선택 가능)를 사용하여 점수를 계산한다. (낮을수록 좋음)

3.7.2 SSIM (Structural Similarity Index Measure):

두 이미지 간의 구조적 유사성을 측정한다. skimage.metrics.structural_similarity를 사용하여 원본 하이트맵과 생성된 하이트맵 간의 SSIM을 계산한다. (높을수록 좋음)

3.7.3 엣지 기반 유사도 (Edge-based Similarity):

질감의 국소적인 변화와 경계선 정보를 반영하기 위해, 두 하이트맵의 Canny 엣지 맵을 추출하고 이들 간의 유사도를 측정한다.

1. Edge SSIM: Canny 엣지 맵 간의 SSIM

2. Edge F1-Score, Precision, Recall: 엣지 맵을 이진화하여 픽셀 단위로 F1 점수, 정밀도, 재현율을 계산.

3.7.4 GLCM (Gray Level Co-occurrence Matrix) 특징 비교:

질감의 통계적 특성을 분석하기 위해 GLCM을 활용한다. skimage.feature.graycomatrix와 graycoprops를 사용하여 Contrast, Dissimilarity, Homogeneity, Energy, Correlation, ASM (Angular Second Moment)과 같은 특징들을 추출, 두 하이트맵에서 추출된 GLCM 특징 벡터 간의 코사인 유사도를 계산한다.

4 실험

본 장에서는 3장에서 제안한 방법론을 바탕으로 수행한 실제 실험 환경, 도출된 결과, 그리고 그 결과에 대한 분석 및 논의를 상세히 기술한다.

4.1 실험 환경

4.1.1 데이터셋 구성 → 학습 및 테스트 데이터 분할 (Holdout 방식):

모델 성능의 일반화 능력을 평가하기 위해 Holdout 방식을 사용하였다. 전체 89개 재질 중, 80개 재질에서 추출된 데이터는 모델 학습에 사용되었으며, 학습 과정에 전혀 사용되지 않은 별도의 9개 재질을 테스트 세트로 분리하여 모델의 최종 성능을 평가하였다. 이러한 재질 기반 분할 방식은 모델이 학습 데이터에 없는 새로운 종류의 질감에 대해 얼마나 잘 일반화하는지 측정하는 데 목적이 있다.

4.1.2 구현 상세

4.1.2.1프레임워크: PyTorch

4.1.2.2주요 라이브러리:

- U-Net 모델 구성 및 인코더 활용: segmentation_models_pytorch (**smp**)
- 지각 손실(LPIPS) 계산: **lpips**
- 이미지 로드, 전처리 및 저장: **torchvision**, **Pillow** (PIL)
- 데이터셋 처리 및 평가 시 이미지 처리: **OpenCV**
- 평가 지표 계산: **scikit-image**, **scikit-learn** (re_evaluate.py 내 사용)

4.1.2.3모델 상세: 3.5절에서 설명한 바와 같이 re_model.py의 create_model 함수를 통해 segmentation_models_pytorch 라이브러리의 U-Net을 사용하였으며, 다양한 사전 학습된 인코더를 선택할 수 있도록 하였다. 본 평가는 EfficientNet-B7을 인코더로 사용한 모델에 대해 수행되었다.

4.1.3 하이퍼파라미터

주요 실험에 사용된 하이퍼파라미터는 다음과 같이 설정되었다.

하이퍼파라미터	값	설명
인코더	efficientnet-b7	U-Net 인코더백본
이미지 크기	256	입출력 이미지의 해상도
배치 크기	40	각횟수 처리 데이터샘플 수
에포크 수	700/1000	1000회 시행, 700모델선택
초기 학습률	1e-4	옵티마이저의 초기 학습률
LPIPS손실가중치	0.5	LPIPS 손실 차지 비중
옵티마이저	Adam	betas: (0.9, 0.999)
손실 함수	$L_{total} = L_{L1} + \lambda_{LPIPS} * L_{LPIPS}$	
LR 스케줄러	ReduceLROnPlateau	
LR 인내심	5	LR감소 전 대기 에포크 수
LR Factor	0.5	학습률 감소 비율
제외 프레임	15	제외할 초기 프레임 인덱스

표 4.1: 하이퍼파라미터

4.1.4 하드웨어 및 소프트웨어 환경

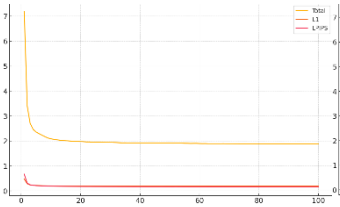
4.1.4.1하드웨어: 경희대학교 세라프 (Moana)

4.1.4.2소프트웨어: Python 3.12, PyTorch 1.12.1, CUDA 11.3, segmentation-models-pytorch 0.3.3, lpips 0.1.4

4.2 실험 결과

4.2.1 학습 곡선 (Training Curves)

L1+LPIPS 손실을 사용, 모델 (EfficientNet-B7 인코더)의 에포크별 손실 변화



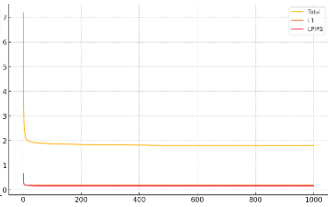


그림 4.1: ~100 에포크 손실변화 | 그림 4.2: ~1000 에포크 손실변화

4.2.2 생성된 하이트맵 예시

4.2.2.1성공적인 사례: PlasticMesh2

완벽히 깔끔하지 못한 면이 있으나 전체적인 구조를 재현하는데 성공하였다.

그림 4.3: 입력이미지 | 그림 4.4: 정답이미지 | 그림 4.5: 추론이미지

4.2.2.2성공적인 사례: ArtificialGrass

엣지 표현이 아쉬우나 재질의 특성을 재현하는데 성공하였다.

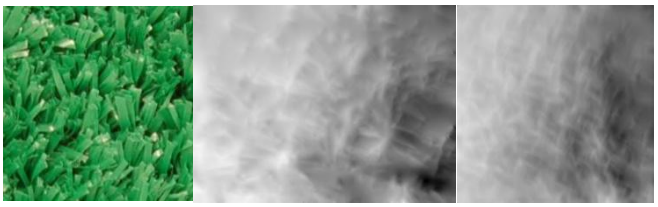


그림 4.6: 입력이미지 | 그림 4.7: 정답이미지 | 그림 4.8: 추론이미지

4.2.2.3한계가 보이는 사례: PaintedWood

표면의 매우 작은 패턴을 재현하려고 노력하였으나 세밀한 표현에서 아쉬움이 보인다.

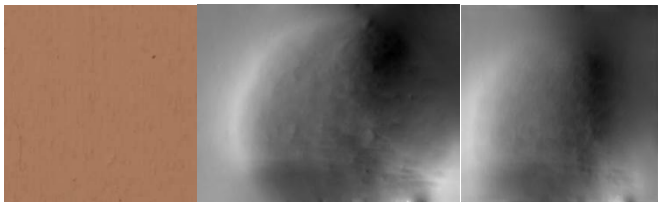


그림 4.9: 입력이미지 | 그림 4.10: 정답이미지 | 그림 4.11: 추론이미지

4.2.3 정량적 평가 결과

학습에 사용되지 않은 9개의 재질로 구성된 테스트 세트에 대해 re_evaluate.py 스크립트를 사용하여 성능을 평가하였다. 그 결과는 다음과 같다.

구분	평가지표	평균	표준편차
전체적 유사도	SSIM↑	0.5745	0.0592
	LPIPS↓	0.4727	0.0711
엣지 기반 유사도	Edge_SSIM↑	0.5366	0.1842
	Edge_F1↑	0.0164	0.0105
	Edge_Precision↑	0.0094	0.0069
	Edge_Recall↑	0.1090	0.0478
통계적 질감	GLCM Cos Similarity↑	0.9958	0.0046
특징 비교	GLCM Contrast	8.51 / 15.77	1.11 / 6.53
	GLCM Homogeneity	0.41 / 0.37	0.02 / 0.04

표 4.2: 정량적 평가 결과

4.3 결과 분석 및 논의

4.3.1 주요 결과 해석

표 4.2의 정량적 평가 결과는 모델의 성능 특성과 한계를 명확하게 보여준다.

4.3.1.1첫째, 모델은 질감의 통계적 특징 프로파일을 매우 잘 모사하지만, 특성의 강도를 과장하는 경향이 뚜렷하게 나타났다. GLCM 특징 벡터 간의 코사인 유사도는 0.9958로 매우 높아, 생성된 하이트맵이 원본과 유사한 종류의 통계적 질감 패턴을 가지고 있음을 시사한다. 하지만 개별 특징 값을 살펴보면, 생성된 하이트맵의 Contrast 평균값(15.77)이 원본(8.51)의 거의 두 배에 달하며, Homogeneity는 원본(0.41)보다 낮게(0.37) 나타났다. 이는 모델이 질감을 재현할 때, 원본보다 더 거칠고 명암 차이가 큰, 즉 과장된 질감을 생성하는 경향이 있음을 의미한다.

4.3.1.2둘째, 전체적인 구조 및 지각적 유사도는

보통 수준에 머물렀다. Original_SSIM 점수는 약 0.57, LPIPS 점수는 약 0.47로, 생성된 이미지가 원본과 구조적으로나 시지각적으로 상당한 차이가 있음을 나타낸다. 이는 모델이 질감의 느낌은 일부 재현하지만, 원본의 정확한 형태와는 거리가 있음을 의미한다.

4.3.1.3셋째, 정확한 엣지(경계선) 생성 능력은 매우 부족하다. Edge_SSIM은 0.54로 낮은 수준이며, 픽셀 단위의 정확도를 측정하는 Edge_F1 점수(0.016)와 정밀도(0.009)는 극히 낮다. 재현율(0.109)이 정밀도보다 월등히 높은 현상은, 모델이 실제 엣지의 존재 위치는 어렴풋이 감지하지만, 그 위치에 정확하고 날카로운 엣지를 생성하지는 못함을 보여준다.

4.3.2 방법론별 비교 분석 (실패 사례 및 개선 과정 포함)

4.3.2.1L1과 LPIPS 손실의 상호작용과 한계: 본 연구에서 사용한 L1과 LPIPS 손실의 조합은 모델의 독특한 행동 양상을 유도한 것으로 분석된다. L1 손실은 픽셀 값의 평균적인 차이를 줄여 전반적인 구조를 맞추도록 하지만, 동시에 이미지를 부드럽게 만드는 경향이 있다. 반면, LPIPS 손실은 인간의 시지각과 유사하게 강한 특징에 더 높은 가중치를 둔다. 모델이 이 두 손실을 동시에 최소화하는 과정에서, L1 손실로 인한 블러 효과를 상쇄하고 LPIPS 손실을 낮추기 위해 질감의 특징, 특히 Contrast를 실제보다 더 강하게 생성하는 방향으로 학습되었을 가능성이 있다. 이는 결과적으로 통계적 프로파일은 유사하지만 그 강도가 과장된 결과로 이어진 것으로 보인다.

4.3.2.2인코더의 특징 추출 능력: EfficientNet-B7은 깊은 인코더로 복잡한 시각적 특징을 효과적으로 추출할 수 있다. 모델은 이 능력을 바탕으로 질감의 종류를 파악하고 그에 맞는 통계적 패턴을 생성하는 데는 성공했으나, 디코더와 손실 함수가 이를 최종적으로 정밀하게 복원하는 데는 한계를 보였다.

4.3.3 모델의 한계점 및 추가 논의

본 실험을 통해 확인된 모델의 명백한 한계점은 다음과 같다.

4.3.3.1질감 특징의 과장: 모델이 질감을 단순히 재현하는 것을 넘어 대비를 크게 증폭시키는 현상은, 촉감의 강도를 정확하게 제어해야 하는 응용 분야(예: 의료 시뮬레이션, 정밀 햅틱 렌더링)에 부적합할 수 있다. 생성된 촉감이 실제와 동일하지 않으며 더 강하게 느껴질 것이다.

4.3.3.2기하학적 부정확성: Original_SSIM과 Edge_F1 점수에서 나타나듯, 모델은 질감의 정확한 형태나 경계선을 생성하는 데 어려움을 겪는다. 이는 생성된 하이트맵을 3D 모델의 텍스처 맵이나 정밀한 가공을 위한 데이터로 사용하기 어렵게 만드는 핵심적인 한계이다.

4.3.3.3일반화 성능의 한계: 테스트 세트에 대한 보통 수준의 SSIM 및 LPIPS 점수는 모델이 학습 데이터셋의 특성에 다소 과적합되었을 수 있으며, 완전히 새로운 질감에 대한 일반화 성능이 더 개선될 필요가 있음을 시사한다.

5 결론 및 향후 연구 방향

5.1 결론

본 연구는 생성형 AI 기술을 기반으로, 시각적 질감 이미지로부터 해당 표면의 3차원적 굴곡 정보를 담은 하이트맵을 생성하는 알고리즘 개발을 목표로 하였다. 이를 위해 Stanford-Gelsight 데이터셋의 동영상 프레임으로부터 (입력 RGB 이미지, 정답 하이트맵) 데이터 쌍을 구축하는 효과적인 파이프라인을 설계하였으며, 사전 학습된 인코더를 사용하는 U-Net 아키텍처를 생성 모델로 채택하였다. 모델 학습에는 픽셀 단위의 구조적 유사성을 위한 L1 손실과 인간의 시지각적 특성을 반영하기 위한 LPIPS 손실을 조합하여

사용하였다.

정량적 평가 결과, 제안된 모델은 질감의 통계적 특성(GLCM)을 매우 유사하게 재현하는 데 성공했음을 확인하였다. 하지만 생성된 하이트맵은 원본에 비해 대비(Contrast)가 과장되는 경향을 보였으며, 전체적인 구조적, 시지각적 유사도(SSIM, LPIPS)는 보통 수준에 머물렀다. 특히, 엣지(경계선)의 픽셀 단위 정확도를 나타내는 F1 점수가 매우 낮게 나타나, 정밀한 경계선 생성에는 명확한 한계가 있음을 확인하였다.

결론적으로 본 연구는 Gelsight 데이터를 활용한 하이트맵 생성 파이프라인을 구축하고, U-Net 기반 모델과 L1/LPIPS 손실 조합의 성능 특성을 심도 있게 분석했다는 점에서 의의를 가진다. 또한, 통계적 질감 재현이라는 강점과 함께 특징 과장 및 엣지 표현력 부족이라는 구체적인 한계점을 정량적으로 규명하여 향후 연구의 명확한 기반을 마련하였다.

5.2 향후 연구 방향

본 연구에서 확인된 한계점들을 극복하고 모델의 성능을 향상시키기 위해 다음과 같은 후속 연구를 제안한다.

5.2.1 모델 성능 개선

5.2.1.1 손실 함수 고도화: 현재 모델의 '질감 특징 과장'과 '엣지 표현력 부족' 문제를 해결하기 위해 손실 함수를 개선할 필요가 있다.

- **첫째**, 생성된 하이트맵의 GLCM Contrast와 같은 특정 통계 값이 원본과 크게 벗어나지 않도록 직접 제어하는 손실 항을 추가하여 특징의 과장을 억제할 수 있다.
- **둘째**, 생성된 하이트맵과 원본의 그래디언트 맵 간의 차이를 최소화하는 손실을 추가하여 엣지의 선명도를 높이는 방안을 모색한다.

5.2.1.2 아키텍처 탐색: U-Net 외에 더 날카로운 이미지 생성에 강점이 있는 것으로 알려진 GAN(Generative Adversarial Network)과 같은 다른 생성 모델 아키텍처를 도입하여 엣지 생성 능력의 근본적인 개선을 시도해 볼 수 있다.

5.2.2 데이터셋 확장: 현재 모델의 일반화 성능을 높이고 특정 데이터셋에 대한 과적합을 방지하기 위해, 자체 Gelsight 장비를 활용하여 더 다양한 종류와 고해상도의 질감 데이터를 직접 수집하고 데이터셋을 확장하는 노력이 필요하다.

5.2.3 실시간성 확보 및 시스템 통합: 생성된 하이트맵을 실제 햅틱 장치에서 즉각적으로 렌더링하기 위해, 모델 경량화 및 최적화를 통해 실시간 추론 성능을 확보하는 연구를 진행한다. 이를 통해 사용자에게 즉각적인 촉감 피드백을 제공하는 통합 시스템을 구축할 수 있다.

5.2.4 다중 모달리티 정보 활용: 시각 정보뿐만 아니라, 질감 표면을 탐색할 때 발생하는 소리(청각)나 압력 변화(물리)와 같은 다른 감각 정보를 함께 모델의 입력으로 사용하여, 더욱 풍부하고 정확한 촉감 생성을 시도해 볼 수 있다.

5.2.5 정성적 사용자 평가 확대: 현재의 정량적 평가를 넘어, 생성된 촉감을 사용자가 실제로 어떻게 인지하는지에 대한 정성적 사용자 평가를 수행해야 한다. 이를 통해 알고리즘의 실효성을 검증하고, 사용자의 피드백을 바탕으로 실질적인 개선 방향을 도출할 수 있을 것이다.

본 연구를 통해 개발된 생성형 AI 기반 촉감 생성 알고리즘은 향후 이러한 후속 연구들을 통해 발전하여, 가상현실(VR), 증강현실(AR), 원격 조작, 디지털 디자인 등 다양한 분야에서 사용자 경험을 획기적으로 향상시키는 데 기여할 수 있을 것으로 기대된다.

6 참고문헌

6.1 **GelSight, Inc. GelSight GitHub:** https://github.com/gelsightinc

6.2 **GelSight GitHub - gsrobotics:** https://github.com/gelsightinc/gsrobotics

6.3 **GelSight Mini Product Sheet:** https://www.gelsight.com/wp-

content/uploads/productsheet/Mini/GS_Mini_Product_Sheet_10.07.24.pdf

6.4 GelSight Mini: https://www.gelsight.com/gelsightmini/

6.5 J. Fan, et al. "Haptic Texture Generation - Dataset". Stanford University: https://sites.google.com/stanford.edu/haptic-texture-generation/dataset?authuser=0

6.6 O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation," in Medical Image Computing and Computer-Assisted Intervention (MICCAI), 2015, pp. 234-241: https://arxiv.org/pdf/1505.04597

6.7 GRASP Laboratory, "The Penn Haptic Texture Toolkit," University of Pennsylvania: https://www.grasp.upenn.edu/projects/the-penn-haptic-texture-toolkit/

6.8 Artificial Neural Tactile Sensing System https://koasas.kaist.ac.kr/bitstream/10203/305329/3/r21_sub06_fulltext_eng.pdf

6.9 P. Iakubovskii. "Segmentation Models PyTorch". GitHub: https://github.com/qubvel-org/segmentation_models.pytorch

6.10 K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in CVPR, 2016: https://arxiv.org/pdf/1512.03385

6.11 K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," in ICLR, 2015: https://arxiv.org/pdf/1409.1556

6.12 R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The Unreasonable Effectiveness of Deep Features as a Perceptual Metric," in CVPR, 2018: https://arxiv.org/pdf/1801.03924
D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," in *ICLR*, 2015: https://arxiv.org/pdf/1412.6980

6.13 OpenCV Team. "OpenCV (Open Source Computer Vision Library)": https://opencv.org

6.14 M. Tan and Q. V. Le, "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks," in ICML, 2019: https://arxiv.org/pdf/1905.11946

7 부록

7.1 데이터셋 예시

본 연구에서 사용된 데이터셋의 예시는 다음과 같다.

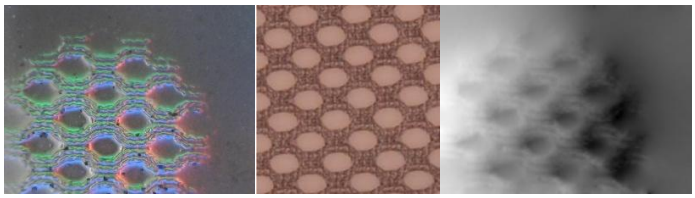


그림 7.1: 원본프레임 | 그림 7.2: 입력이미지 | 그림 7.3: 정답하이트맵

7.2 모델 구조

7.2.1 전체 구조: U-Net (인코더-디코더 대칭 구조)

7.2.2 인코더: EfficientNet-B7 (ImageNet사전학습)

역할: 입력 이미지(256x256x3)에서 점차 작은 스케일의 특징들을 추출

7.2.3 디코더 (Decoder): U-Net 기본 디코더 블록

역할: 인코더가 추출한 특징들을 다시 원본 크기로 키우면서 하이트맵(256x256x1)을 생성

7.2.4 핵심 기술: 스킵 연결 (Skip Connection)

역할: 인코더의 각 단계에서 추출된 특징(디테일 정보)을 디코더의 해당 단계로 직접 전달하여, 이미지의 세밀한 정보가 사라지지 않게 함