

웹 도색 시뮬레이터

2025-1 **AJOU**
SOFTCON

팀 명 프라브러쉬

팀 원 정승환

지도교수 고옥

멘 토 권민성

개발동기 및 목적

타미야 레드



모델 조립 취미에서 도색은 매우 중요한 과정이지만, 몇가지 문제점이 있다.

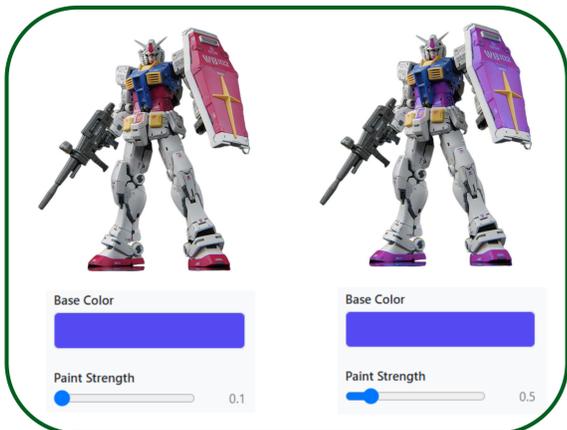
도료 회사가 제공하는 샘플 사진이나 RGB 값만으로는 도료의 실제 결과를 예측하기 어렵고, 값비싼 도료로 시행착오를 겪으면서 재료 낭비와 부품 손상이 발생한다. 포토샵 같은 기존 프로그램은 구독료 부담과 복잡한 사용법이 장벽이 된다.

이 문제를 해결하기 위해 신현준 교수님의 “계층적 표현을 활용한 간결한 팔레트 및 색상 모델 생성과 조작” 논문을 기반으로 도색 결과를 실제와 가깝고, 도료 별 특성을 고려하여 시뮬레이션 할 수 있는 시스템을 구축하였다.

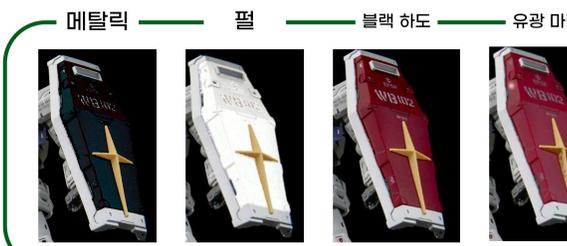
개발내용



- 원본 이미지의 빛, 디테일을 보존
- 도료 색상표와 실제 도색한 표면 차이 구현
- 정확한 픽셀 단위
- 원본 색상이 같은부품 판별



- 사용자가 원하는 색상 직접 입력 가능
- 모델의 표면 색상을 밑바탕색으로 감안하여 처리
- 검칠 횟수, 도료 별 차폐력 차이 구현



- 메탈릭, 펄 등 특수 도료
- 하도(서페이스) 강도, 색상에 따른 시각적 차이
- 유광, 무광 마감

오픈소스 URL

<https://github.com/fetooths/pramodelPainting>

참고 문헌 1) 신현준 외, 2019, Succinct Palette and Color Model Generation and Manipulation Using Hierarchical Representation, Pacific Graphics 2019.
2) Tominaga, S., & Yong, R. K. F. 2022, Appearance estimation and reconstruction of glossy object surfaces based on the dichromatic reflection model., Color Research & Application vol(48).

주요기술

Hierarchical Representation



1. 순수 색상 픽셀 분리 (Pure Color Sample Selection)
CIE Lab*색 공간에서 주변 색상과 분산이 낮은 순수 픽셀을 분리

2. 로컬 인스턴트 생성 (Local Instance Generation)
SLIC 알고리즘과 밀도기반 클러스터링을 통해 이미지를 공간적 덩어리로 구분하고, 대표 색상과 분포를 계산

3. 계층적 색상 모델 구축 (Hierarchical Color Model Construction)
로컬 인스턴트를 리프 노드로 사용하여 색상을 병합한 계층적 색상 모델을 구축

4. 대화형 팔레트/색상 모델 생성 및 조작(Interactive Palette/Color Model Generation and Manipulation)
분산 임계값 또는 구성 요소 수를 기준으로 특정 수준에서 계층 구조를 잘라 팔레트를 추출하거나 사용자가 지정하여 병합 및 분할 작업 수행 후, 결과 색상 모델을 사용하여 색상 변경 실행



해당 기술을 사용함으로써 반사영역, 그림자에 혼합된 색상까지 동일 비율로 처리 가능함
또한 동일 색상 런너에서 비롯된 부품 일체로 제어 가능



메탈릭 : OpenCV의 가우시안 랜덤 노이즈와 블러링을 사용하여 금속 표면을 구현

펄 : 다중 계층 노이즈 시스템을 사용, 큰 스케일의 펄 패턴으로 입자 위치를 결정하고, 미세한 스케일 패턴으로 입자 방향 변화를 표현하여 이산화티타늄 코팅 운모의 특수한 반사 효과를 시뮬레이션

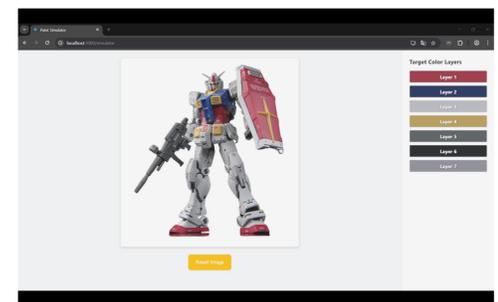
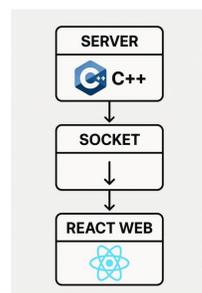


유광 : 노멀 맵을 활용, 집중 정반사 맵으로 하이라이트 중심을 식별하고 가우시안 분포 하이라이트를 생성. 총 빛 에너지의 85%는 하이라이트에, 15%는 인근 산란광 표현에 사용

무광 : 가우시안 블러를 통해 빛 에너지를 넓은 영역으로 분산

이색 반사 모델 기반 표면: 광택 물체의 외관 추정 및 복원 논문을 참고

결과 및 분석



이미지 처리 서버 : cuda, opencv contrib 연동한 c++ 네이티브 HTTP 소켓 서버, Base64 인코딩, json, http 모듈 연동을 이용한 이미지 전송 및 비동기화 처리 가능함. 추후 cdn 사용하여 안정적인 성능 구현 목표

웹 프론트 엔드 : react 웹 프론트 엔드 동적 구현 완료. 사용자가 원하는 색상을 원하는 부품에 시뮬레이션 하고, 그 결과를 실시간으로 확인할 수 있음.

