

3D Gaussian Splatting을 위한 3단계 실시간 NPR 포스트 프로세싱 파이프라인

나민태^o 김선정*
한림대학교
{M24052, sunkim}@hallym.ac.kr

A 3-Stage Real-Time NPR Post-Processing Pipeline for 3D Gaussian Splatting

Mintae Na^o Sun-Jeong Kim*
Hallym University

요약

3D Gaussian Splatting(3DGS)은 메쉬 구조가 부재하고 고주파 노이즈가 내재되어 있어, 기존 비실사 렌더링(NPR) 기법을 직접 적용할 경우 외곽선 품질이 저하되는 문제가 있다. 본 논문은 이를 해결하기 위해 색상 추상화(Color Abstraction)와 구조 추출(Structure Extraction)을 분리한 3단계 실시간 NPR 포스트 프로세싱 파이프라인을 제안한다. 제안 기법은 Kuwahara 필터를 이용해 3DGS의 입자감을 회화적인 면으로 추상화하고, 가우시안 블러 기반 전처리를 통해 고주파 스플랫 노이즈를 억제한 뒤, Sobel 기반 외곽선 검출 결과를 합성하여 스타일화된 영상을 생성한다. 본 시스템은 Unity 6 URP RenderGraph 환경에서 구현되었으며, 실험 결과 3DGS의 아티팩트를 효과적으로 억제하면서도 고품질 NPR 결과와 실시간 성능을 동시에 확보함을 확인하였다.

Abstract

Applying non-photorealistic rendering (NPR) to 3D Gaussian Splatting (3DGS) is challenging due to its lack of explicit mesh geometry and inherent high-frequency noise, which degrade edge quality in direct screen-space stylization. We propose a 3-stage real-time NPR post-processing pipeline that decouples color abstraction from structure extraction. The method abstracts noisy 3DGS appearance into painterly surfaces using a Kuwahara filter, suppresses splat noise via Gaussian blur-based pre-filtering, and generates stylized results by compositing Sobel-based edges with the abstracted color image. Implemented in Unity 6 URP RenderGraph, the proposed method effectively suppresses 3DGS artifacts while maintaining high-quality stylization and real-time performance.

키워드: 3D 가우시안 스플래팅, 비실사 렌더링, 유니티, 구조 추출, 포스트 프로세싱, 쿠와하라 필터

Keywords: 3D Gaussian Splatting, NPR, Unity, Structure Extraction, Post-processing, Kuwahara Filter

1. 서론

최근 3D Gaussian Splatting(3DGS)[1]은 장면을 명시적인 3D Gaussian primitive 집합으로 표현함으로써, NeRF 계열 대비 높은 렌더링 효율과 실시간성을 제공하는 대표적인 radiance field 기법으로 주목받고 있다[1],[2]. 이러

한 특성은 가상현실, 디지털 트윈, 게임과 같은 실시간 실감형 콘텐츠 분야에서 3DGS의 활용 가능성을 확대하고 있다[2].

한편 비실사 렌더링(Non-Photorealistic Rendering, NPR)은 현실의 충실한 재현보다 추상화와 강조를 통해 시각적 전달력과 예술적 표현성을 높이는 데 초점을 둔다

*corresponding author: *correspondin author: Sun-Jeong Kim / Hallym University (sunkim@hallym.ac.kr)

[3],[4]. 따라서 3DGS의 사실적인 렌더링 결과를 painterly 또는 toon 스타일로 변환하려는 요구 역시 자연스럽게 증가하고 있다.

그러나 기존 NPR 기법들은 대체로 깊이, 노멀, 메쉬와 같은 기하학적 단서를 활용하여 외곽선을 검출하거나 형상 특징을 강조하므로[5], 명시적인 메쉬 구조가 없는 3DGS에는 직접 적용하기 어렵다[1],[6]. 특히 3DGS는 화면 공간에서 splat이 중첩되어 장면을 형성하므로, 시점 변화나 샘플링 비율 변화에 따라 고주파 노이즈와 불연속성이 발생할 수 있으며[7], 원본 결과에 직접 edge detection을 적용할 경우 splat 자국이 실제 외곽선으로 오인되는 문제가 발생한다. 이는 결과적으로 NPR의 심미성과 구조 전달력을 저해하는 주요 원인이 된다[6],[7]. 이러한 한계를 해결하기 위해 본 논문은 색상 추상화(Color Abstraction)와 구조 추출(Structure Extraction)을 분리한 3단계(3-Pass) 실시간 NPR 포스트 프로세싱 파이프라인을 제안한다. 먼저 Kuwahara 필터[4],[8]를 이용해 3DGS의 고주파 입자감을 회화적인 면으로 추상화하고, 이후 가우시안 블러 기반 전처리를 통해 구조 정보를 안정화한 뒤 강건한 외곽선을 추출한다. 마지막으로 추상화된 색상 정보와 구조적 외곽선을 합성하여, 3DGS의 노이즈 아티팩트를 억제하면서도 스타일화된 NPR 결과를 생성한다.

또한 본 연구는 이를 Unity 6 URP RenderGraph 환경에서 구현하여, 다단계 처리에 따른 메모리 오버헤드를 줄이면서 실시간 렌더링 성능을 유지한다. 결과적으로 본 연구는 별도의 재학습 없이 적용 가능한 zero-shot 후처리 관점에서, 3DGS 기반 인터랙티브 애플리케이션의 시각적 표현 범위를 확장하는 실용적 NPR 파이프라인을 제안한다.

본 연구가 목표로 하는 NPR 스타일은 photorealistic한 3DGS 영상을 실시간 toon/painterly hybrid 스타일로 변환하는 것으로, 표면의 미세한 노이즈를 억제하면서 주요 외곽선을 강조하는 데 초점을 둔다.

2. 관련 연구

2.1 비실사 렌더링과 이미지 기반 추상화

비실사 렌더링(NPR)은 현실의 물리적 재현보다 추상화와 강조를 통해 형상 이해와 시각적 전달력을 높이는 데 초점을 둔다. 전통적인 NPR 연구는 깊이, 노멀, 윤곽선과 같은 기하학적 단서를 활용하여 외곽선과 형상 특징을 강조하는 방향으로 발전해 왔다[5]. 이후 연구는 이미지 및 비디오 수준의 abstraction으로 확장되었으며, 실시간 환경에서도 painterly 또는 cartoon-like한 결과를 생성할 수 있음이 보고되었다[3],[4]. 그러나 이러한 방법들은 대체로 안정적인 기하 정보 또는 일반 영상 입력을 전제로

하므로, 3DGS와 같은 비정형 장면 표현에 직접 적용할 경우 한계가 있다.

2.2 3D Gaussian Splatting의 표현 특성과 한계

3D Gaussian Splatting(3DGS)[1]은 장면을 명시적인 3D Gaussian primitive 집합으로 표현하여, NeRF 계열 대비 높은 렌더링 효율과 실시간성을 제공하는 radiance field 기법이다[2]. 그러나 3DGS는 명시적인 메쉬 연결 구조가 없고, 화면 공간에서 splat이 중첩되어 장면을 형성하므로 기존 geometry-aware NPR 기법을 직접 적용하기 어렵다. 또한 시점 변화나 샘플링 비율 변화에 따라 aliasing과 high-frequency artifact가 발생할 수 있으며[7], 이러한 특성은 edge detection 과정에서 실제 구조와 splat 노이즈를 구분하기 어렵게 만든다.

2.3 뉴럴 렌더링 및 3DGS 스타일라이제이션 연구

3D 장면 스타일화 연구는 StylizedNeRF[9], ARF[10], StyleGaussian[11], ReGS[12], StylizedGS[13]와 같이 주로 학습 또는 최적화 기반 접근으로 발전해 왔다. 이러한 방법들은 3차원 일관성이 높다는 장점이 있으나, 새로운 스타일 적용 시 재학습 또는 추가 최적화가 필요하며 계산 비용이 증가할 수 있다[11],[12],[13]. 따라서 실시간 상호작용이 중요한 응용에서는 즉시 적용 가능한 경량 접근이 여전히 필요하다.

2.4 본 연구의 위치

본 연구는 위와 같은 한계를 고려하여, 3DGS의 렌더링 결과에 대해 별도의 재학습 없이 적용 가능한 screen-space 기반 NPR 포스트 프로세싱 파이프라인을 제안한다. 핵심 아이디어는 색상 추상화와 구조 추출을 분리하는 것으로, Kuwahara 기반 필터링을 통해 회화적 표면 추상화를 수행하고[4],[8], blur 기반 전처리 후 edge detection을 수행함으로써 splat noise에 강건한 외곽선을 생성한다. 즉, 본 연구는 학습 기반 스타일화와 달리 zero-shot으로 적용 가능하며, 실시간성과 적용 편의성을 중시하는 3DGS stylization 방법이라는 점에서 차별성을 가진다.

3. 제안하는 방법

3.1 시스템 개요

3D Gaussian Splatting(3DGS)은 수많은 Gaussian splat의 중첩을 통해 장면을 표현하므로, 렌더링 결과에 고주파 노이즈와 픽셀 단위 불연속성이 포함되기 쉽다. 이러한 특성은 NPR 스타일링 과정에서 색상 불연속성과 외곽선 검출의 불안정성을 유발하며, 원본 결과에 직접 edge detection을

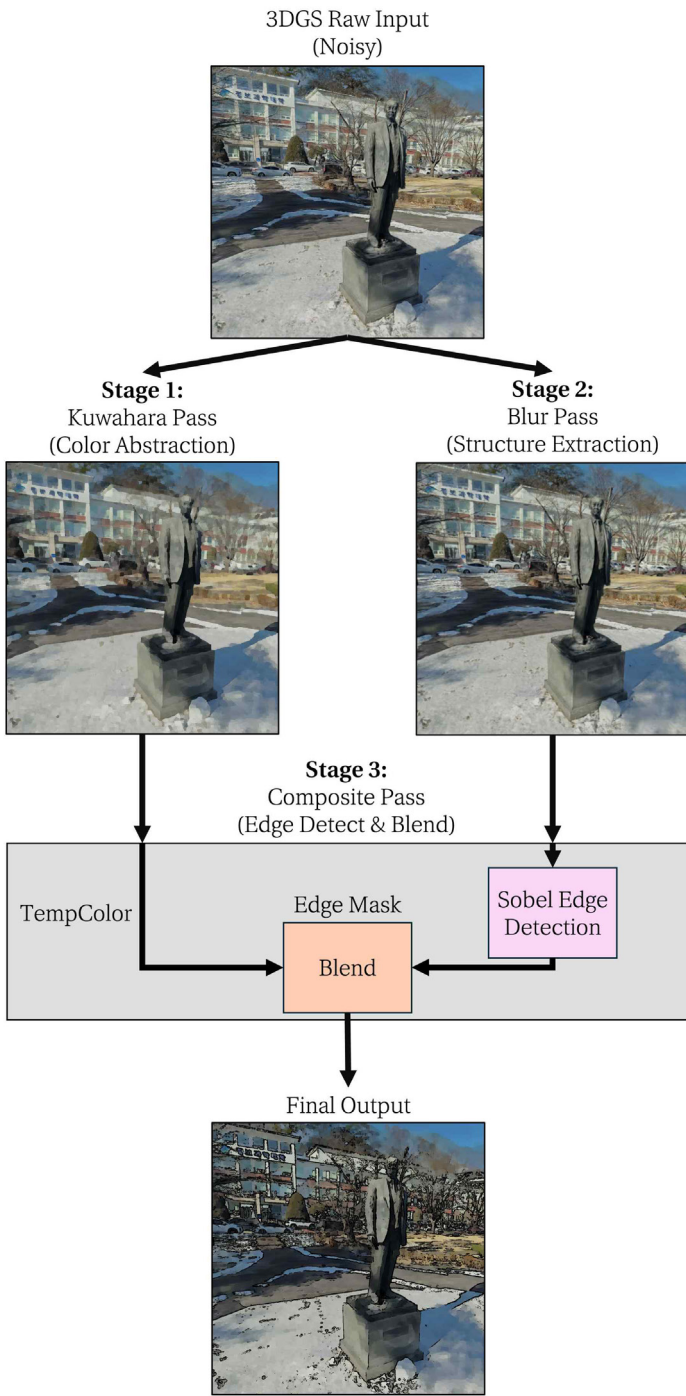


Figure 1. Overview of the proposed 3-stage real-time NPR pipeline. The pipeline decouples color abstraction (Stage 1) and structure extraction (Stage 2) to handle 3DGS artifacts effectively before the final composition (Stage 3)

적용할 경우 splat 자국이 실제 구조적 경계로 오인되는 문제를 초래한다. 이를 해결하기 위해 본 연구는 (1) 색상 추상화(Color Abstraction), (2) 구조 추출(Structure Extraction), (3) 합성(Composition)으로 구성된 3단계 실시간 포스트 프로세싱 파이프라인을 제안한다. 첫 번째 단계에서는 Kuwahara 필터를 이용하여 3DGS의 입자감을 회화적인 면으로 평

탄화하고, 두 번째 단계에서는 가우시안 블러 기반 전처리를 통해 구조 정보만을 안정적으로 추출한다. 마지막 단계에서는 전처리된 구조 텍스처에서 강건한 외곽선을 검출하고, 이를 추상화된 색상 정보와 결합하여 최종 NPR 결과를 생성한다. 본 시스템은 3DGS 데이터를 실시간으로 로드하고 렌더링하기 위한 오픈소스 기반 플러그인[14]을 확장하여 구현되었으며, RenderGraph를 활용하여 다단계 처리 과정의 리소스 생명주기와 메모리 할당을 효율적으로 관리한다. 제안하는 전체 파이프라인의 흐름은 Figure 1과 같다.

3.2 Stage 1:Kuwahara 필터를 이용한 표면 추상화

첫 번째 단계는 3DGS의 이산적인 입자감을 회화적인 면(Surface)으로 변환하는 과정이다. 3DGS의 원본 이미지는 픽셀 단위의 분산(Variance)이 매우 높기 때문에, 이를 평탄화하기 위해 Kuwahara 필터[8]를 적용한다. Kuwahara 필터는 픽셀 (x, y) 를 중심으로 하는 윈도우를 Figure 2와 같이

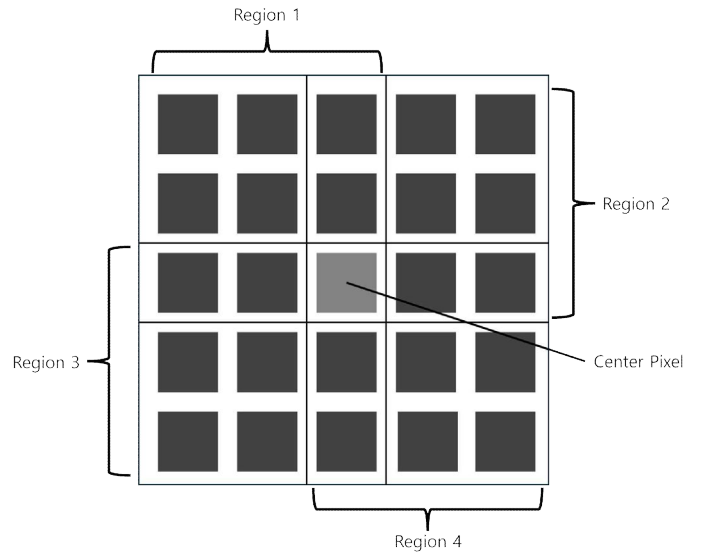


Figure 2. Kernel structure of the Kuwahara filter (5×5 window example). The filter divides the window into four overlapping subregions to calculate the mean and variance.

4개의 사분면 영역 Q_k ($k = 1, 2, 3, 4$)으로 분할한다. 각 영역 Q_k 에 포함된 픽셀 수 N 에 대하여, 평균 μ_k 와 분산 σ_k^2 은 식 (1)과 같이 정의된다.

$$\mu_k = \frac{1}{N} \sum_{(i,j) \in Q_k} C(i,j), \quad (1)$$

$$\sigma_k^2 = \frac{1}{N} \sum_{(i,j) \in Q_k} (C(i,j) - \mu_k)^2$$

여기서 $C(i, j)$ 는 픽셀의 색상 값을 의미한다. 필터는 4개의 영역 중 표준편차가 가장 작은 영역(Minimum Variance)을

선택하여, 해당 영역의 평균값을 최종 픽셀 색상 $C_{abstract}$ 로 결정한다.

$$k_{\min} = \arg \min_k (\sigma_k^2), \quad C_{abstract}(x, y) = \mu_{k_{\min}} \quad (2)$$

식 (2)의 과정을 통해 엣지(Edge) 정보가 있는 영역(분산이 큰 영역)은 배제되고 색상이 균일한 영역이 선택되므로, 3DGS 내부의 자잘한 스플랫 텍스처는 뭉개지고 주요 경계면은 보존된다. 본 실험에서는 필터의 윈도우 크기를 13×13 (반지름 $R = 6$)로 설정하여 3DGS의 입자감을 효과적으로 평탄화하였다. 이 결과물은 최종 단계에서 베이스 컬러(T_{color})로 사용된다.

3.3 Stage 2: 가우시안 블러를 이용한 구조 추출

본 연구의 핵심 제안인 두 번째 단계는 강건한(Robust) 외곽선 검출을 위한 전처리 과정이다. 3DGS 렌더링 결과물에 포함된 스플랫 노이즈는 고주파 성분(High-frequency)으로, 이는 엣지 검출 알고리즘이 실제 물체의 외곽선과 단순한 스플랫 자국을 구분하지 못하게 하는 주된 원인이 된다.

따라서 본 단계에서는 저주파 통과 필터(Low-pass Filter)인 가우시안 블러(Gaussian Blur)를 적용하여 구조 정보 텍스처($T_{structure}$)를 생성한다. 엄밀한 의미의 구조 추출(Structure Extraction)은 텍스처 질감만 제거하고 엣지는 보존하는 고비용의 연산(예: RTV 등)을 필요로 하나, 본 연구에서는 실시간 렌더링 성능 확보를 위해 가우시안 블러를 이용한 근사(Approximation) 방식을 채택하였다. 이 과정은 단순한 이미지 블러링이 아니라, 후속 단계인 Sobel 연산자가 미세한 스플랫 노이즈에는 반응하지 않고 객체의 거시적인 형태(Macro-structure) 변화에만 반응하도록 유도하는 전처리(Pre-filtering) 역할을 수행한다. 2차원 가우시안 분포 함수 $G(x, y)$ 는 식 (3)과 같다.

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2 + y^2}{2\sigma^2}} \quad (3)$$

이론적으로는 식 (3)을 따르지만, 본 연구에서는 실시간 렌더링 성능을 극대화하기 위해 이를 3×3 크기의 이산적인 이항 커널(Discrete Binomial Kernel)로 근사하여 구현하였다. 사용된 커널 K 는 식 (4)와 같이 중심 픽셀에 가장 높은 가중치(4)를 부여하고 거리에 따라 감소하는 고정 가중치(Fixed Weights)를 가진다.

$$K = \frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} \quad (4)$$

이 커널은 $\sigma \approx 1.0$ 인 가우시안 분포의 정수비 근사값에 해당하며, 부동소수점 연산 비용을 최소화하면서도 3DGS의 스플랫 노이즈(Splat Noise)를 효과적으로 평탄화(Smoothing)할 수 있다. 최종적으로 구조 텍스처 $T_{structure}$ 는 입력 이미지 I 와 커널 K 의 합성곱으로 생성된다.

$$T_{structure}(x, y) = K * I(x, y) \quad (5)$$

이 과정을 통해 3DGS의 미세한 입자 노이즈가 제거되고, 물체의 전체적인 형태(Low-frequency Structure) 정보만이 남게 된다. 이 텍스처는 사용자에게 직접 보이지 않으며, 오직 다음 단계의 엣지 검출 연산을 위한 참조 데이터로만 활용된다.

3.4 Stage 3: 특징 합성

마지막 단계에서는 앞서 생성된 두 텍스처를 결합하여 최종 결과를 생성한다. 먼저 Stage 2에서 생성된 구조 텍스처($T_{structure}$)를 입력으로 받아 Sobel 연산자[15]를 적용한다. 수평 방향(G_x)과 수직 방향(G_y)의 미분 근사값은 식 (6)와 같이 계산된다.

$$G_x = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix} * T_{structure}, \quad (6)$$

$$G_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ +1 & +2 & +1 \end{bmatrix} * T_{structure}$$

이때 원본 이미지가 아닌, 노이즈가 제거된 $T_{structure}$ 를 참조하므로 3DGS의 아티팩트에 영향을 받지 않는 깨끗한 그라디언트를 얻을 수 있다. 엣지의 강도(Magnitude) E 는 다음과 같다.

$$E = \sqrt{G_x^2 + G_y^2} \quad (7)$$

검출된 엣지 강도 E 는 임계값(Threshold) 함수를 거쳐 가중치 마스크 M_{edge} 로 변환된다. 이때 임계값 $T_{edge} = 0.25$ 를 적용하여, $E < T_{edge}$ 인 경우 노이즈로 간주하여 제거하고 그 이상인 경우 엣지로 판단하였다. 최종적으로 픽셀 셰이더는 식 (8)과 같이 Stage 1의 추상화된 색상(T_{color})과 엣지 색상(C_{edge} , 일반적으로 검은색)을 선형 보간(Linear Interpolation)하여 최종 NPR 이미지 C_{final} 을 생성한다.

$$C_{final} = (1 - M_{edge}) \cdot T_{color} + M_{edge} \cdot C_{edge} \quad (8)$$

RenderGraph는 이 합성이 완료되는 즉시 중간 텍스처인 T_{color} 와 $T_{structure}$ 의 메모리를 해제하여 시스템 리소스를 최적화한다.

3.5 RenderGraph 기반 리소스 관리 및 동기화

본 연구에서 제안하는 3-Pass 파이프라인은 다수의 렌더 타겟(Render Target)을 사용하므로, 비효율적인 메모리 관리는 성능 저하의 주된 원인이 될 수 있다. 이를 방지하기 위해 Unity RenderGraph의 자동화된 리소스 수명 관리(Transient Resource Lifespan Management)[16]를 적극 활용하였다.

중간 단계에서 생성되는 T_{color} 와 $T_{structure}$ 텍스처는 RenderGraph.CreateTexture를 통해 임시 리소스(Transient Resource)로 선언된다. 엔진은 프레임 그래프(Frame Graph)를 분석하여, Stage 3(합성)이 종료되는 시점에 해당 텍스처들이 더 이상 참조되지 않음을 감지하고 즉시 메모리를 반환하거나 다른 렌더 패스의 리소스로 재사용(Aliasing)한다. 이를 통해 개발자가 수동으로 Release를 호출할 때 발생할 수 있는 메모리 누수나 해제 지연 문제를 원천적으로 차단하였다.

또한, 각 패스(Pass) 간의 실행 순서와 데이터 의존성(Dependency)은 builder.UseTexture API를 통해 명시적으로 정의된다. RenderGraph 컴파일러는 이 정보를 바탕으로 GPU 파이프라인 상의 적절한 배리어(Barrier)와 동기화 지점을 자동으로 삽입하여, 불필요한 GPU 유휴 시간(Idle Time)을 최소화하고 병렬 처리가 가능한 구간을 최적화한다.

3.6 셰이더 레벨 최적화

반복 연산이 많은 이미지 처리 알고리즘의 실시간성을 보장하기 위해 HLSL 셰이더 레벨에서 다음과 같은 최적화 기법을 적용하였다.

첫째, 고정 커널(Fixed Kernel) 사용과 루프 제거이다. Stage 2의 가우시안 블러 패스에서는 런타임에 가중치를 계산하는 for 반복문 대신, 3×3 오프셋을 하드코딩(Hard-coding)하여 텍스처 샘플링을 수행하였다. 이는 동적 분기(Dynamic Branching)와 레지스터 할당 비용을 제거하여 GPU의 처리 효율을 높였다. 또한 Kuwahara 필터에서는 픽셀당 수십 회의 텍스처 샘플링을 수행하는 고비용 연산이다. 본 연구에서는 필터의 반경(Radius)을 상수로 고정하고 [unroll] 속성을 명시하여, 컴파일러가 반복문을 병렬 명령어로 전개하도록 유도하였다. 이는 동적 분기(Dynamic Branching) 비용을 제거하여 셰이더 실행 속도를 크게 향상시킨다.

둘째, LOD(Level of Detail) 연산 제거이다. 포스트 프로세싱 단계에서는 mip맵(Mipmap)이 필요하지 않으므로, SAMPLE_TEXTURE2D 대신 SAMPLE_TEXTURE2D_LOD(..., 0) 매크로를 사용하여 mip맵 레벨 계산에 소요되는 연산 부

하를 제거하였다.

셋째, 구조적 버퍼링(Structured Buffering)이다. 엣지 검출 단계에서 Sobel 연산 시 매번 9번의 텍스처 샘플링을 수행하는 대신, 3-Pass 구조를 통해 이미 전처리된(Pre-processed) 블러 텍스처를 1:1로 매핑하여 샘플링함으로써, 텍스처 캐시 적중률(Cache Hit Rate)을 높이고 메모리 대역폭 효율을 극대화하였다.

4. 실험 및 결과

4.1 실험 환경 및 데이터셋

본 제안 시스템의 유효성을 검증하기 위해 Mip-NeRF 360 데이터셋[17]과 자체 촬영한 데이터셋(대학교 캠퍼스 환경)을 사용하여 3DGS 씬을 구축하였다. 구현 및 테스트는 Unity 6 (URP) 환경에서 수행되었으며, 하드웨어 사양은 Intel i7-13700KF CPU, NVIDIA GeForce RTX 4090 GPU, 64GB RAM이다. 렌더링 해상도는 FHD(1920x1080)를 기준으로 하였으며, 4K(3840x2160) 해상도에서의 부하 테스트를 병행하였다. Figure 3은 본교 캠퍼스를 직접 촬영하여 3DGS로 학습시킨 후, 제안하는 NPR 파이프라인을 적용한 결과이다. 이를 통해 표준 벤치마크 데이터뿐만 아니라, 사용자가 직접 생성한 데이터에서도 안정적인 스타일라이징이 가능함을 확인하였다.



Figure 3. Qualitative results on the self-captured dataset (Campus). The proposed pipeline effectively stylizes real-world scenes captured with a smartphone.



(a)



(b)

Figure 4. Ablation study on Structure Extraction. (a) Edge detection on raw 3DGS input shows severe noise artifacts. (b) Proposed method using structure extraction generates clean and robust edges.

4.2 구조 추출 전처리의 유효성 검증

본 연구의 핵심 제안인 '구조 추출(Structure Extraction) 단계'의 필요성을 검증하기 위해 소거 연구(Ablation Study)를 수행하였다. Figure 4는 가우시안 블러(Gaussian Blur) 전처리 유무에 따른 엣지 검출 결과를 비교한다.

- (a) Raw Input + Sobel (1-Pass): 전처리 없이 원본 3DGS 이미지에 직접 Sobel 연산을 수행한 경우이다. 3DGS 특유의 타원체 스플랫 경계면이 모두 엣지로 오인(False Positive)되어, 화면 전체에 검은 점 형태의 고주파 노이즈가 발생함을 확인할 수 있다. 이는 심미성을 저해할 뿐만 아니라 물체의 형태를 인지하기 어렵게 만든다.
- (b) Blurred Input + Sobel (Ours, 3-Pass): 제안하는 방법으로, 구조 텍스처($T_{structure}$)를 경유하여 엣지를 검출한 결과이다. 블러링 과정을 통해 스플랫 단위의 미세 노이즈가 제거됨에 따라, 엣지 검출기가 스플랫 자국에 반응하지 않고 물체의 주요 기하학적 외곽선(Semantic Edge)만을 안정적으로 검출함을 확인하였다.

이를 통해 색상 처리와 구조 처리를 분리(Decoupling)하는 3단계 파이프라인이 3DGS의 아티팩트 문제를 해결하는 데 필수적임을 입증하였다.

4.3 엣지 검출 알고리즘 비교 분석

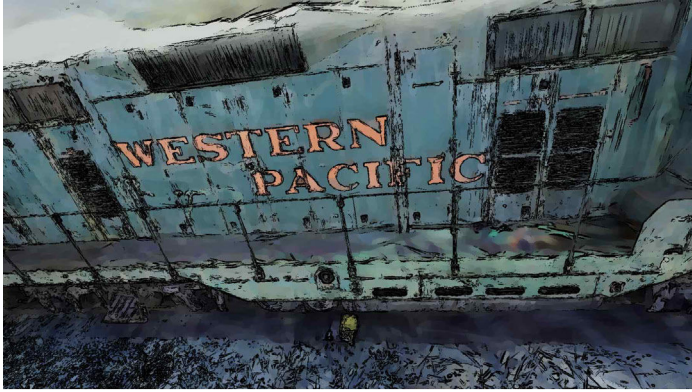
구조 추출 패스가 적용된 상태에서, 어떤 엣지 검출 알고리즘이 3DGS에 가장 적합한지 비교 분석하였다(Figure 5). Roberts 연산자[18]는 커널 크기가 작아 노이즈에 가장 민감하게 반응하여 엣지가 과도하게 검출되는 경향을 보였다. Canny 알고리즘[19]은 NMS(Non-Maximum Suppression) 과정을 포함하여 얇은 선을 검출하지만, 3DGS 데이터의 불연속성으로 인해 선이 끊기는(Disconnected) 현상이 발생하였다. 반면, 본 연구에서 채택한 Sobel 연산자는 중심 픽셀 주변의 3×3 영역에 대한 가중 평균(Weighted Average)을 포함하고 있어, 미세한 스플랫 노이즈에 대한 민감도를 낮추고 지역적인 변화율(Gradient)을 안정적으로 계산할 수 있다. 이러한 특성은 불연속적인 3DGS 데이터에서 가장 부드럽고 끊김 없는 외곽선을 생성하여, 만화적 스타일링에 가장 적합한 결과를 보였다.

4.4 파라미터 제어 및 시각적 변화

본 제안 기법은 사용자가 Kuwahara 필터의 윈도우 반경(Radius, R)을 조절하여 스타일의 추상화 정도를 실시간으로 제어할 수 있다. Figure 6는 R 값의 변화에 따른 시각적 결과를 비교한 것이다.

- Case (a) $R=3$: 필터링 범위가 좁아 3DGS 특유의 고주파 노이즈(High-frequency Artifacts)가 완전히 제거되지 않았으며, 회화적인 느낌보다는 단순한 노이즈 낀 이미지에 가깝다.
- Case (b) $R=6$ (Proposed): 본 연구에서 제안하는 최적 값으로, 3DGS의 스플랫 자국은 효과적으로 뭉개져(Clustering) 유화적인 질감을 형성하면서도, 자전거 프레임과 같은 주요 객체의 형태는 선명하게 유지됨을 확인할 수 있다.
- Case (c) $R=9$ 이상: 필터링 강도가 강해지면서 객체의 경계가 무너지는 현상이 발생한다. 특히 (d) $R=12$ 인 경우, 자전거 바퀴살(Spokes)과 같은 미세한 디테일이 배경과 혼합되어 소실되는 과도한 뭉개짐(Over-smoothing) 현상이 관찰되었다.

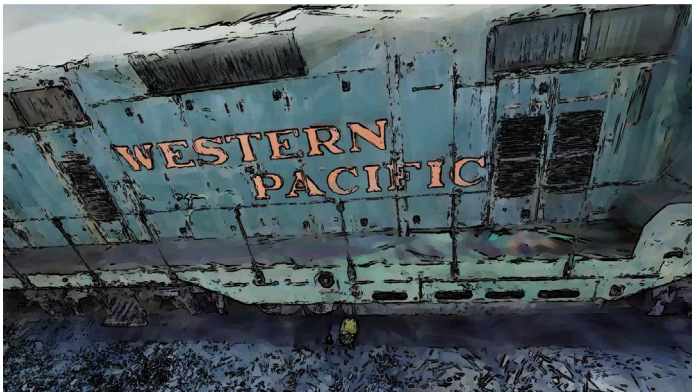
이러한 실험 결과는 사용자가 콘텐츠의 목적에 따라 '정밀한 묘사'와 '회화적 추상화' 사이의 균형점을 선택할 수 있음을 시사하며, 이는 인터랙티브 애플리케이션에서 중요한 시각적 조절 기능을 제공한다.



(a) Roberts



(b) Canny

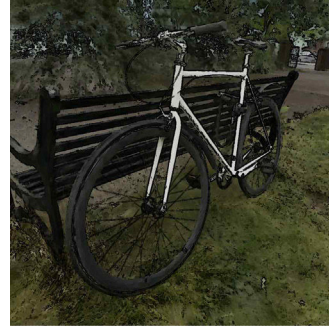


(c) Sobel (Ours)

Figure 5. Comparison of edge detection algorithms. Sobel operator (Ours) demonstrates the best trade-off between detail preservation and noise suppression compared to Roberts and Canny.

4.5 동적 시점 안정성

정지 영상뿐만 아니라 카메라가 이동하거나 줌인/아웃 (Zoom In/Out)하는 동적 상황에서의 안정성을 평가하였다 (Figure 7). 3DGS는 카메라가 물체에 근접(Zoom-in)할수록 스플랫 간의 간격이 벌어지는 특성이 있다. 실험 결과, 제안하는 Kuwahara 필터는 이러한 스플랫 사이의 빈 공간을 주변 색상으로 블렌딩(Blending)하여 매끄러운 효과를 보였으며, 블러 기반의 엣지 검출은 스플랫이 갈라지는 시점에서도 외



(a) R=3



(b) R=6



(c) R=9



(d) R=12

Figure 6. Comparison of stylization results according to Kuwahara filter radius (R).

곽선의 떨림(Flickering)을 최소화하여 시각적 일관성을 유지하였다. 이는 별도의 매쉬 복원 과정 없이도 3DGS 콘텐츠의 몰입감을 유지하는 데 기여한다.

4.6 성능 평가

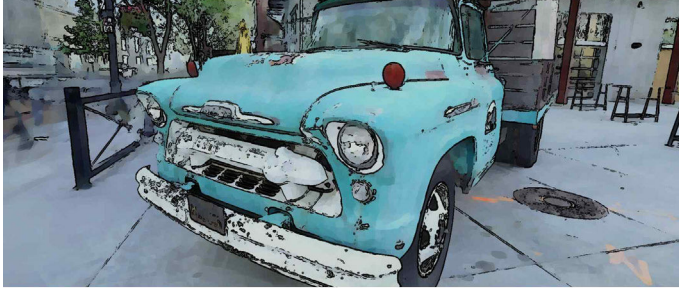
제안하는 3-Pass 파이프라인의 연산 부하를 측정하기 위해 FHD 및 4K 해상도에서 프레임 레이트(FPS)와 프레임 타임(Frame Time)을 측정하였다(Table 1). Raw 3DGS 방식 대비 텍스처 읽기/쓰기 횟수가 증가하여 프레임 타임은 소폭 증가(약 0.2ms~1.4ms)하였다. 4K 환경에서도 여전히 평균 100 FPS 이상의 성능을 기록하였다. 이는 RenderGraph의 최적화된 리소스 관리가 메모리 대역폭 병목을 효과적으로 완화했기 때문으로 분석된다. 결과적으로 본 시스템은 고품질의 스타일라이징을 수행하면서도 실시간성이 요구되는 인터랙티브 애플리케이션에 충분히 적용 가능하다.

Table 1. Performance comparison. Despite the multi-pass overhead, the proposed method maintains real-time performance.

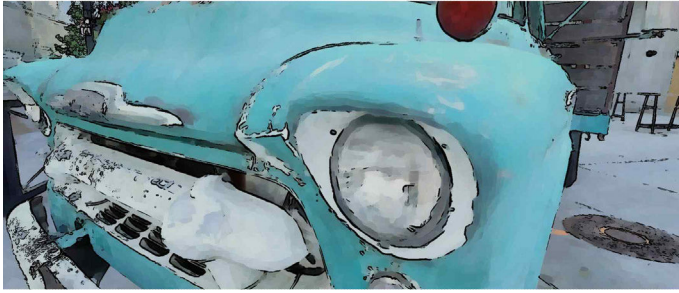
Resolution	Method	Avg FPS	Frame Time (ms)
1920x1080 (FHD)	Raw 3DGS	220	5.0
	Ours (3-Pass)	191	5.2
3840x2160 (4K)	Raw 3DGS	131	7.6
	Ours (3-Pass)	111	9.0



(a)



(b)



(c)

Figure 7. Dynamic view stability test. The proposed pipeline maintains visual coherence even when zooming in, effectively masking the separation of splats.

4.7 타 NPR 기법과의 비교

Table 2는 제안하는 기법과 3DGS 스타일라이징 연구들을 비교 분석한 것이다. StyleGaussian[11], ReGS[12], StylizedGS[13] 등 기존의 데이터 중심(Data-centric) 접근법들은 3D 가우시안의 속성(Feature, Geometry)을 직접 학습하거나 최적화하므로 3차원적 일관성(Consistency)은 뛰어나지만, 사전 학습(Pre-processing)이 필수적이거나 높은 메모리 비용(VRAM Usage)이 발생한다는 한계가 있다.

반면, 제안하는 기법(Ours)은 스크린 공간 처리 방식의 특성상 시간적 일관성에는 물리적 한계가 존재하나, 1) 별도의 학습 과정이 전혀 없는 제로 샷(Zero-shot) 방식이며, 2) RenderGraph의 임시 리소스(Transient Resource)만을 활용하여 메모리 오버헤드를 최소화했다는 점에서, 즉각적인 상호작용이 필요한 실시간 애플리케이션에 가장 적합한 특성을 보인다.

5. 토의 및 한계점

본 접근은 3D 장면 표현 자체를 수정하지 않고 최종 렌더링 결과의 화면 공간(screen space)에서 필터링과 외곽선 추출을 수행한다는 점에서, 실시간성과 적용 편의성 측면의 장점을 가지는 동시에 몇 가지 구조적 한계를 가진다. 우선 각 프레임을 독립적으로 처리하므로 카메라가 빠르게 이동하거나 회전할 때 temporal consistency가 저하될 수 있으며, 이에 따라 shower-door effect나 boiling artifact가 발생할 가능성이 있다. 또한 3DGS의 view-dependent color 특성으로 인해 시점 변화에 따라 입력 색상 그래디언트가 달라질 경우, 동일한 기하 구조에 대해서도 외곽선 안정성이 영향을 받을 수 있다. 더불어 본 연구는 noisy depth를 직접

Table 2 Comparison with State-of-the-Art 3DGS Stylization Methods

Feature	Ours	StyleGaussian[11]	ReGS[12]	StylizedGS[13]
Core Method	Screen-space Post-processing	3D Feature Embedding & Decoding	Reference-based Optimization	Stroke-based Optimization
Space	2D Screen Space	3D Latent Space	3D Geometry & Color	3D Color & Opacity
Pre-processing	None (Zero-shot)	Required (Feature Distillation)	Required (Scene Optimization)	Required (Optimization)
Style Switching	Instant (Real-time)	Instant (Inference)	Slow (Retraining)	Slow (Retraining)
Consistency	Low (Screen-space limit)	High (3D Consistent)	High (Geometry-aware)	High
Memory Cost	Very Low (Transient)	High (High-dim Features)	Moderate	Moderate

활용하지 않고 색상 기반 구조 추출 전략을 채택하였기 때문에, 배경과 객체 사이의 색상 대비가 약한 장면에서는 경계 검출 성능이 제한될 수 있다.

이러한 한계는 제안 기법이 3차원 일관성보다 screen-space 기반의 즉시성과 경량성을 우선하는 설계를 취했기 때문에 발생한다. 즉, 본 방법은 별도의 재학습 없이 실시간으로 적용 가능하다는 점에서 실용성이 높지만, 구조적 일관성 측면에서는 화면 공간 후처리 방식의 물리적 제약을 함께 가진다. 따라서 본 연구의 기여는 모든 상황에서 완전한 NPR 해법을 제시하는 데 있다기보다, 3DGS에 적합한 실시간 스타일라이징의 가능성을 실용적인 수준에서 보여주었다는 데 있다.

향후 연구에서는 모션 벡터 기반 reprojection과 temporal accumulation을 도입하여 시간적 안정성을 높이고, view-independent diffuse 성분이나 스무딩된 depth map, semantic segmentation 정보를 결합하여 구조 추출의 강건성을 향상시킬 필요가 있다. 또한 외곽선 두께, 색상 양자화, 음영 단순화와 같은 표현 파라미터를 확장하면 toon, painterly, illustrative 스타일을 보다 다양하게 제어할 수 있을 것이다. 이러한 점에서 본 연구는 완결된 최종 해법이라기보다, 3DGS에 적합한 실시간 NPR 후처리의 가능성과 확장 방향을 제시하는 출발점으로 볼 수 있다.

6. 결론

본 논문에서는 3D Gaussian Splatting(3DGS)의 고주파 노이즈와 구조적 불연속성으로 인해 기존 screen-space 기반 비실사 렌더링(NPR) 기법을 직접 적용하기 어렵다는 문제를 해결하기 위해, 색상 추상화(Color Abstraction)와 구조 추출(Structure Extraction)을 분리한 3단계 실시간 NPR 포스트 프로세싱 파이프라인을 제안하였다.

제안 기법은 먼저 Kuwahara 필터를 이용하여 3DGS의 입자감을 회화적인 면으로 추상화하고, 이후 가우시안 블러 기반 구조 추출 단계를 통해 고주파 스플랫 노이즈를 억제한 뒤, Sobel 기반 외곽선 검출 결과를 추상화된 색상 정보와 합성함으로써 최종 NPR 영상을 생성한다. 이를 통해 원본 3DGS 결과에 직접 외곽선 검출을 적용할 때 발생하는 splat 기반 노이즈 아티팩트를 줄이고, 보다 안정적인 스타일라이징 결과를 얻을 수 있었다.

실험 결과, 제안 기법은 3DGS 특유의 자글자글한 노이즈를 효과적으로 완화하면서도 주요 객체의 형태를 유지하는 외곽선을 생성하였으며, 파라미터 조절을 통해 정밀한 묘사와 회화적 추상화 사이의 균형을 제어할 수 있음을 확인하였다. 또한 FHD 및 4K 해상도에서 모두 실시간 성능을 유지함으로써, 제안하는 다단계 포스트 프로세싱 구조가 인터랙티브 응용 환경에서도 실용적으로 활용 가능함을 보였다.

본 연구의 의의는 장면 자체를 재학습하거나 최적화하지 않

고도, 렌더링 결과 단계에서 즉시 적용 가능한 zero-shot NPR 파이프라인을 제시했다는 데 있다. 기존의 데이터 중심 스타일라이징 기법들이 높은 3차원 일관성을 제공하는 대신 추가적인 학습 또는 최적화 비용을 요구하는 것과 달리, 제안 기법은 상대적으로 낮은 메모리 오버헤드와 즉시성을 바탕으로 게임 엔진 환경, 디지털 트윈 시각화, 실시간 콘텐츠 제작과 같은 응용 분야에 적합한 특성을 가진다. 또한 Unity 6 URP RenderGraph 기반 구현을 통해 다단계 처리 과정의 중간 리소스를 효율적으로 관리함으로써, 품질과 성능을 함께 확보하였다.

종합하면, 본 연구는 3DGS의 데이터적 특성을 고려한 실시간 NPR 포스트 프로세싱 파이프라인을 제안하고, 색상 추상화와 구조 추출의 분리를 통해 3DGS 스타일라이징의 실용적 가능성을 확장하였다. 이러한 접근은 실시간성과 적용 편의성이 중요한 다양한 인터랙티브 시나리오에서 유용한 기반 기법으로 활용될 수 있을 것으로 기대된다.

감사의 글

본 과제(결과물)는 2025년도 교육부 및 강원특별자치도의 재원으로 강원RISE센터의 지원을 받아 수행된 지역혁신중심 대학지원체계(RISE)의 결과입니다.(2025-RISE-10-009)

References

- [1] B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis, "3D Gaussian Splatting for Real-Time Radiance Field Rendering," *ACM Transactions on Graphics*, vol. 42, no. 4, pp. 139:1-139:14, 2023.
- [2] G. Chen and W. Wang, "A Survey on 3D Gaussian Splatting," *arXiv preprint arXiv:2401.03890*, 2024.
- [3] H. Winnemöller, S. C. Olsen, and B. Gooch, "Real-time Video Abstraction," *ACM Transactions on Graphics*, vol. 25, no. 3, pp. 1221-1226, 2006.
- [4] J. E. Kyprianidis, H. Kang, and J. Döllner, "Image and Video Abstraction by Anisotropic Kuwahara Filtering," *Computer Graphics Forum*, vol. 28, no. 7, pp. 1955-1963, 2009.
- [5] T. Saito and T. Takahashi, "Comprehensible Rendering of 3-D Shapes," *Computer Graphics*, vol. 24, no. 4, pp. 197-206, 1990.
- [6] O. Wegen, W. Scheibel, M. Trapp, R. Richter, and J. Döllner, "A Survey on Non-Photorealistic Rendering Approaches for Point Cloud Visualization," *IEEE Transactions on Visualization and Computer Graphics*, vol. 31, no. 9, pp. 4511-4533, 2025.
- [7] Z. Yu, A. Chen, B. Huang, T. Sattler, and A. Geiger, "Mip-Splatting: Alias-free 3D Gaussian Splatting," *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 19447-19456, 2024.
- [8] M. Kuwahara, K. Hachimura, S. Eiho, and M. Kinoshita, "Processing of RI-Angiocardiographic Images," in *Digital Processing of Biomedical Images*, pp. 187-202, 1976.

- [9] Y.-H. Huang, Y. He, Y.-J. Yuan, Y.-K. Lai, and L. Gao, "StylizedNeRF: Consistent 3D Scene Stylization as Stylized NeRF via 2D-3D Mutual Learning," Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 18342-18352, 2022.
- [10] K. Zhang, N. Kolkin, S. Bi, F. Luan, Z. Xu, E. Shechtman, and N. Snavely, "ARF: Artistic Radiance Fields," Proceedings of the European Conference on Computer Vision - ECCV 2022, pp. 717-733, 2022.
- [11] K. Liu, F. Zhan, M. Xu, C. Theobalt, L. Shao, and S. Lu, "StyleGaussian: Instant 3D Style Transfer with Gaussian Splatting," in SIGGRAPH Asia Technical Communications 2024, pp. 21:1-21:4, 2024.
- [12] Y. Mei, J. Xu, and V. M. Patel, "ReGS: Reference-based Controllable Scene Stylization with Gaussian Splatting," Advances in Neural Information Processing Systems, vol. 37, pp. 4035-4061, 2024.
- [13] D. Zhang, Y.-J. Yuan, Z. Chen, F.-L. Zhang, Z. He, S. Shan, and L. Gao, "StylizedGS: Controllable Stylization for 3D Gaussian Splatting," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 47, no. 12, pp. 11961-11973, 2025.
- [14] Y. Wu, "gsplat-unity," GitHub repository, 2025. Available: <https://github.com/wuyize25/gsplat-unity>
- [15] I. Sobel and G. Feldman, "A 3x3 Isotropic Gradient Operator for Image Processing," Stanford Artificial Intelligence Project, 1968.
- [16] Unity Technologies, "Universal Render Pipeline Documentation," Unity Manual. Available: <https://docs.unity3d.com/Packages/com.unity.render-pipelines.universal@17.0/manual/index.html>
- [17] J. T. Barron, B. Mildenhall, D. Verbin, P. P. Srinivasan, and P. Hedman, "Mip-NeRF 360: Unbounded Anti-Aliased Neural Radiance Fields," Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 5470-5479, 2022.
- [18] L. G. Roberts, "Machine Perception of Three-Dimensional Solids", Ph.D. thesis, Massachusetts Institute of Technology, 1963.
- [19] J. Canny, "A Computational Approach to Edge Detection," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 8, no. 6, pp. 679-698, 1986.

< 저자 소개 >

나민태



- 2021 한림대학교 융합소프트웨어학 학사
- 2024~현재 한림대학교 컴퓨터공학 석사
- 관심분야: 컴퓨터그래픽스, NPR
- <https://orcid.org/0009-0006-5369-1038>

김선정



- 1996년 고려대학교 컴퓨터학과 학사
- 1998년 고려대학교 컴퓨터학과 석사
- 2003년 고려대학교 컴퓨터학과 박사
- 2000년 이스라엘 Tel-Aviv 대학교 연구원
- 2004년 독일 RWTH-Aachen 대학교 박사후연구원
- 2005년~현재 한림대학교 컴퓨터공학과 교수
- 관심분야: 컴퓨터그래픽스, 가상/증강현실, 모바일게임 및 앱 개발
- <https://orcid.org/0000-0002-8663-4578>