Journal of Next-generation Convergence Information Services Technology Vol.15, Suppl.1, November (2026) pp.25-38
ISSN 2384-101X(p) 2672-1163(e) *http://dx.doi.org/10.29056/jncist.2026.11.03*

독립형 Windows DICOM CT 뷰어의 단계별 성능 평가와 동일 마스크 기반 지연 전처리 효과 분석

Stepwise Performance Evaluation and Same-Mask Analysis of Lazy Preprocessing in a Standalone Windows DICOM CT Viewer

권나현[1], 정다엘[2], 이규만[3]*

Nahyeon Kwon[1], Dael Jeong[2], Kyuman Lee[3]*

요 약

본 연구는 WPF 기반 사용자 인터페이스, fo-dicom 기반 DICOM 처리, 그리고 ActiViz.NET을 통한 VTK 볼륨 렌더링으로 구성된 독립형 Windows DICOM CT 뷰어를 구현하고, DICOM 폴더 입력부터 2D MPR 준비, VTK 초기화, 프리셋 전환, FPS·프레임 시간 및 메모리까지의 처리 파이프라인을 단계별로 평가하였다. 공개 TCIA CT 시리즈 3개에 대해 Release|x64, 웜 캐시(warm-cache), 교차 데이터셋 순서로 초기화(initialization) 측정 10회와 품질별 전체 실행 10회를 수행하였다. DICOM 폴더 입력부터 2D MPR 준비까지의 시간과 VTK 창 초기화 시간 표본 중앙값은 small 533/848 ms, medium 848/2,153 ms, large 2,345/5,310 ms였다. Balanced 프리셋을 고정한 Fast 모드의 FPS 표본 중앙값은 각각 24.21, 13.43, 4.93이었으며, 모든 품질 모드별 실행에서 원시 프레임 시간 100개를 보존하였다. 동일 최적화 신체 마스크와 볼륨(volume) 생성기를 사용한 합성 데이터 기반 제거 실험(n=30)에서 지연 초기화 시간은 즉시 전처리 대비 50.00 – 53.85% 감소하였다. 본 연구의 기여는 두 상위 시간 측정 지표의 경계를 분리하고, 최초 사용과 캐시 재사용 시의 상호작용 및 원시 프레임(raw-frame) 기반 꼬리 지연시간(tail latency)을 재현 가능한 결과 패키지로 연결한 것이다.

핵심어 : DICOM CT 뷰어, VTK, 볼륨 렌더링, 지연 전처리, 성능 평가

Abstract

This study implements a standalone Windows DICOM CT viewer using WPF for the user interface, fo-dicom for DICOM processing, and VTK through ActiViz.NET for volume rendering, and evaluates its pipeline from DICOM-folder input to 2D-MPR readiness, VTK initialization, preset switching, FPS/frame time, and memory usage. For three public TCIA CT series, ten initialization runs and ten full runs per quality mode were executed using a Release|x64 build, a declared warm-cache policy, and interleaved dataset order. Sample medians for folder-to-2D-MPR-ready / VTK-window initialization were 533/848 ms

1 Department of Bio Medical Devices, Gachon University, Seongnam, Korea [Graduate Student]
e-mail: knh3144@gachon.ac.kr
2 Department of Bio Medical Devices, Gachon University, Seongnam, Korea [Graduate Student]
e-mail: dael0207@gachon.ac.kr
3 Department of Bio Medical Devices, Gachon University, Seongnam, Korea [Professor]
e-mail: manleekyu@gachon.ac.kr (Corresponding author)

Received(September 19, 2026), Review Result(1st: October 13, 2026), Accepted(November 12, 2026), Published(November 30, 2026)

© 2026 The Authors. Published by NCISS.
This is an open access article licensed under the Creative Commons Attribution-NonCommercial 4.0 International License. To view a copy of this license, visit http://creativecommons.org/licenses/by-nc/4.0/.

(small), 848/2,153 ms (medium), and 2,345/5,310 ms (large). Fast-mode sample-median FPS values with the Balanced preset fixed were 24.21, 13.43, and 4.93, and every quality run retained 100 raw frame-time samples. In a same-mask synthetic ablation (n=30), lazy initialization reduced time by 50.00–53.85% relative to eager initialization. The contribution is a traceable benchmark framework that separates parent timing boundaries, first-use versus cached interaction, and raw-frame tail latency.

Keyword : DICOM CT Viewer, VTK, Volume Rendering, Lazy Preprocessing, Performance Evaluation

1. 서론

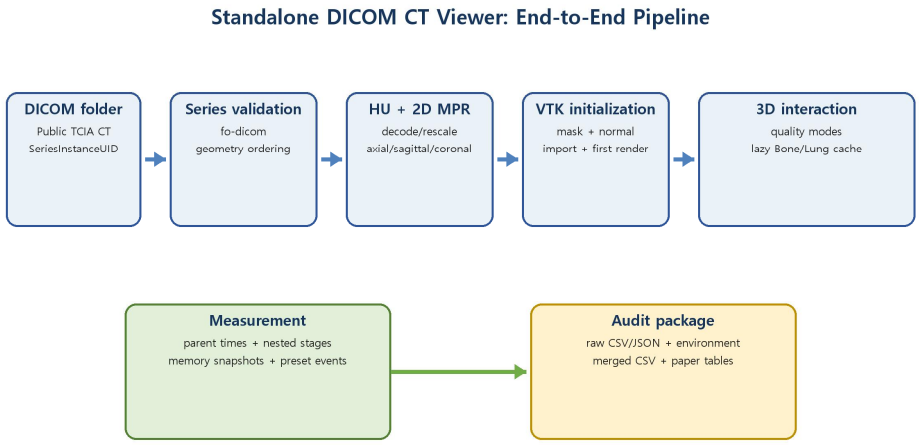
CT DICOM 뷰어(viewer)는 의료 영상 확인, 수술 계획, 교육, 연구 데이터 검수 과정에서 중요한 역할을 한다. CT 영상은 여러 장의 2D 슬라이스(slice)가 하나의 3D 볼륨을 구성하는 형태로 저장되며, DICOM 파일에는 픽셀 데이터(pixel data)뿐 아니라 시리즈(series) 정보, 간격 정보(spacing), 촬영 매개변수(acquisition parameter), 환자 및 기관 메타데이터(metadata) 등이 함께 포함된다 [1]. 연구 및 교육 환경에서는 이러한 DICOM 시리즈를 불러와 슬라이스 구성을 확인하고, HU 기반으로 조직별 구조를 시각화하며, 필요한 경우 3D 볼륨 렌더링을 통해 해부학적 구조(anatomical structure)를 직관적으로 확인할 수 있는 뷰어가 필요하다.

최근 DICOM 뷰어 연구는 XR·웹(Web)·GPU 기반 볼륨 렌더링과 다양한 배포 환경을 다룬다. 선행 연구에서는 HoloLens 2, DirectX 11 및 OpenXR 기반 XR 프로토타입을 제시했다 [2]. 웹 기반 DICOM 뷰어 관련 연구에서는 후보 16개 중 12개의 기능을 조사하고 온라인 시연이 가능한 7개 뷰어의 성능을 비교하여 브라우저(browser)와 하드웨어(hardware)가 결과 해석에 중요함을 보였다 [3]. 또 다른 연구에서는 무료 웹 뷰어 13개를 식별한 뒤 6개를 평가했다 [4]. BlueLight는 JavaScript 기반 클라이언트 측(client-side) 저비용 계산 경로를 통해 계산·메모리 부담을 줄이는 접근을 제시했다 [5]. 본 연구는 이들과 달리 WPF/ActiViz.NET/VTK 기반 단일 Windows 데스크톱 처리 파이프라인(pipeline)의 내부 단계와 캐시 경계를 분석하며, 서로 다른 하드웨어·렌더러(renderer)·데이터셋의 FPS를 직접 비교하여 우열을 판단하지 않는다.

기존 연구와 구현 사례는 주로 렌더링 품질, FPS, GPU 가속(acceleration) 및 상호작용(interaction) 방식에 집중하는 경우가 많다. 그러나 실제 사용자는 DICOM 폴더 선택 이후 파일 탐색(file discovery), 메타데이터 읽기, 시리즈 그룹화(grouping), 슬라이스 정렬(sorting), 픽셀 디코딩(pixel decoding), HU 볼륨 생성, 전처리(preprocessing), VTK 파이프라인 구성, 최초 렌더링 및 프리셋 전환까지의 지연을 연속적으로 경험한다. 이에 본 연구는 렌더링 FPS뿐 아니라 초기화 파이프라인과 상호작용 지표를 함께 계측한다.

따라서 연구문제는 (RQ1) DICOM 폴더 입력부터 2D-MPR 준비까지의 시간과 VTK 초기화 시간의 병목이 볼륨 크기에 따라 어떻게 변하는가, (RQ2) 품질 모드(quality mode)가 고정 프리셋의 FPS와 꼬리 프레임 시간(tail frame-time)에 어떠한 상충관계(trade-off)를 만드는가, (RQ3) 동일 마스

크·자료형·볼륨 생성기 조건에서 즉시·지연 생성 시점만 바꾸면 초기 전처리 비용이 얼마나 달라지는가로 정의하였다. [그림 1]은 전체 시스템 경계와 결과 산출 흐름을 나타낸다.



[그림 1] 시스템 아키텍처 및 CT 렌더링 파이프라인

[Fig. 1] System architecture and CT rendering pipeline

2. 이론적 배경

2.1 DICOM CT 볼륨 렌더링

DICOM CT 시리즈는 다수의 슬라이스를 하나의 3D 볼륨으로 재구성하여 시각화할 수 있다. CT 픽셀 값은 RescaleSlope와 RescaleIntercept를 적용하여 HU 값으로 변환되며, HU 값은 뼈(bone), 연부조직(soft tissue), 폐(lung) 등 조직별 시각화 프리셋을 구성하는 기준으로 활용된다.

2.2 VTK 광선 투사 파이프라인

VTK의 광선 투사 렌더링(Ray Casting rendering)은 카메라에서 발사된 광선이 볼륨을 통과하면서 각 샘플링 지점의 HU 값을 전달 함수(transfer function)와 불투명도 함수(opacity function)에 적용하여 최종 픽셀 색상을 계산하는 방식이다. SampleDistance는 광선 샘플링 간격을 의미하며, 값이 작을수록 렌더링 품질은 높아질 수 있으나 계산 비용이 증가한다.

2.3 지연 전처리 및 프리셋 캐시

즉시 전처리(eager preprocessing)는 초기화 단계에서 일반(normal)·뼈·폐 볼륨을 모두 생성한다. 지

연 전처리(lazy preprocessing)는 일반 볼륨만 초기 생성하고, 뼈 및 폐 프리셋에 필요한 볼륨은 최초 사용 시 생성하여 캐시에 저장한다. 본 연구는 [그림 3]의 동일 마스크(same-mask) 기반 제거 실험(ablation)에서 BuildBodyMaskOptimized와 볼륨 생성기(builder)를 고정하고 생성 시점만 비교하며, 실제 DICOM 실행 결과와 합성 데이터 기반 전처리(synthetic preprocessing) 결과를 구분한다.

3. 시스템 및 계측 방법

3.1 구현 시스템

구현 시스템은 WPF 기반 Windows 데스크톱 애플리케이션(application)으로 구성되며 [6], DICOM 시리즈 불러오기, 축상면(Axial)·시상면(Sagittal)·관상면(Coronal) 2D MPR 뷰어, DICOM 메타데이터 확인 및 HU 기반 VTK 3D 볼륨 렌더링 프리셋을 제공한다 [7]. DICOM 불러오기 단계에서는 fo-dicom을 사용하여 사용자가 선택한 단일 파일 또는 폴더를 기준으로 동일한 SeriesInstanceUID를 가진 슬라이스를 하나의 시리즈로 구성한다 [8]. 슬라이스 정렬은 ImageOrientationPatient와 ImagePositionPatient로부터 계산한 슬라이스 법선 방향 위치를 우선 사용하고, 기하 정보가 없을 때 ImagePositionPatient의 Z값, InstanceNumber, 파일명 순으로 대체 기준(fallback)을 적용한다. PixelSpacing, SpacingBetweenSlices 및 SliceThickness를 이용하여 3D 볼륨의 물리적 간격을 추정한다.

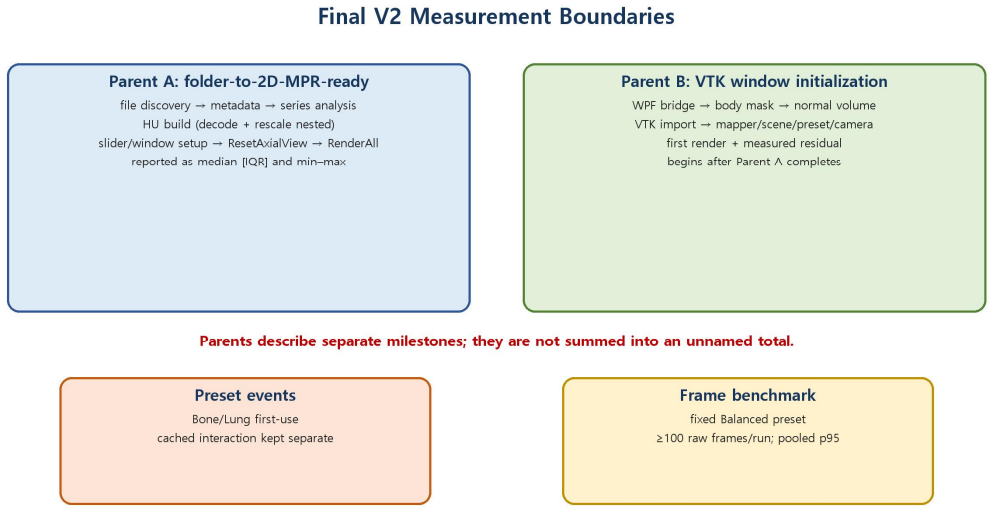
최종 GUI 벤치마크는 .NET Framework 4.8, fo-dicom.Desktop 4.0.8, ActiViz.NET.x64 5.8.0 및 Kitware.VTK 5.8.0.607 기반 Release|x64 빌드에서 수행하였다. 실험에 사용한 소스코드는 Git 커밋 c50644e(태그: paper-benchmark-v2-final-c50644e)으로 고정하였으며, 실행 파일의 SHA-256 해시값을 기록하였다. 동일 마스크 기반 합성 데이터 비교 실험의 재실행 코드와 공개 결과 패키지는 별도의 Git 커밋 57039de(태그: paper-reproducibility-v3.1-20260818)으로 고정하였다. 전체 커밋 해시와 실행 파일의 SHA-256 해시값을 포함한 재현성 정보는 공개 저장소에서 확인할 수 있다 [7]. 환경은 Windows 11 Home(빌드 26200), Intel Core i5-13600KF, RAM 32 GB, NVIDIA GeForce GTX 1660 SUPER(드라이버 버전 32.0.15.9186), 화면 해상도 1920×1080, 주사율 60 Hz 및 화면 배율 100%(96 DPI)였다.

각 슬라이스의 픽셀 데이터는 8비트 또는 16비트 입력을 대상으로 읽어 들이며, RescaleSlope와 RescaleIntercept를 적용하여 HU 기반 정수 볼륨으로 변환한다. 3D 렌더링 단계에서는 ActiViz.NET을 통해 VTK 볼륨 렌더링 파이프라인을 구성하고 [9][10], 전달 함수와 불투명도 함수를 이용하여 CT 조직별 시각화 프리셋을 제공한다.

DICOM 폴더 입력부터 2D MPR 준비까지의 시간에는 순수 입출력뿐 아니라 파일 탐색, 메타데이터 및 시리즈 분석, HU 볼륨 생성, 슬라이더 설정, 2D 창 초기화, ResetAxialView 및 RenderAll 실행이 포함된다. 따라서 해당 지표를 ‘DICOM 폴더 입력부터 2D MPR 준비까지의 상위 측정 시

간'으로 정의하였다. VTK 창 초기화 시간은 이 측정 구간이 종료된 뒤 시작하며, 앞선 DICOM 측정 구간을 포함하지 않는다. 두 지표는 구분되는 상위 측정 시간(parent metric)이므로 합산하지 않는다. [그림 2]는 이러한 측정 경계를, [표 1]은 단계별 성능 지표와 단위를 나타낸다.

작업 집합(Working Set)과 전용 메모리(Private Memory)는 VTK 초기화 직후 고정된 시점에서 측정하였으며, 실행 전체의 최대 메모리 사용량을 의미하지 않는다.



[그림 2] V2 벤치마크 측정 범위

[Fig. 2] V2 benchmark measurement boundaries

[표 1] 벤치마크 성능 지표 및 측정 단계

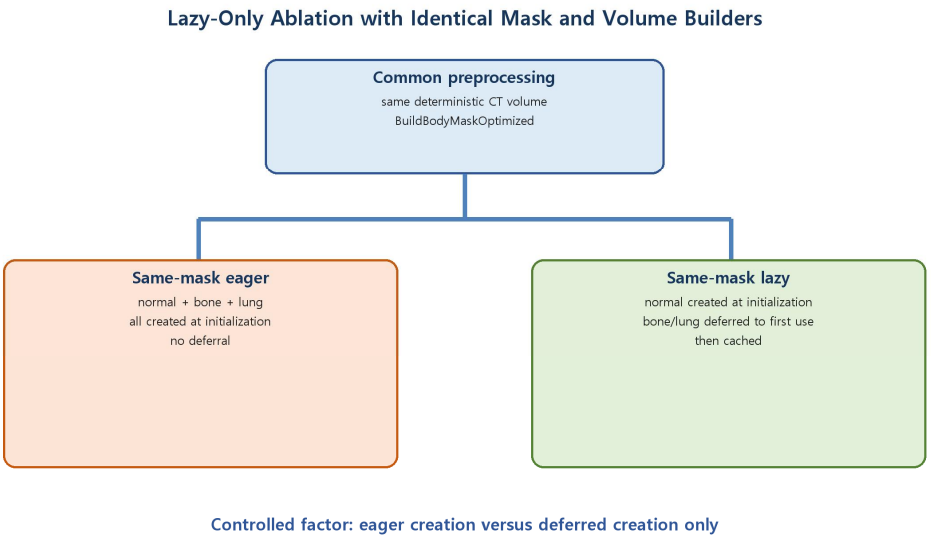
[Table 1] Benchmark metrics and measurement stages.

지표	설명	단위	측정 단계
DICOM 폴더 입력부터 2D MPR 준비까지의 상위 지표	파일 탐색부터 HU 볼륨 생성, 슬라이더·창·화면 설정 및 최초 2D RenderAll 실행까지	ms	상위 지표 A
파일 탐색 시간	DICOM 후보 파일 탐색 시간	ms	불러오기
메타데이터 읽기 시간	DICOM 태그 및 시리즈 메타데이터 읽기 시간	ms	불러오기
시리즈 분석 및 슬라이스 정렬 시간	시리즈 그룹화 및 슬라이스 정렬 시간	ms	불러오기
픽셀 디코딩 및 값 변환 시간	픽셀 데이터 디코딩 및 RescaleSlope·RescaleIntercept 적용 시간	ms	HU 볼륨 생성
HU 볼륨 생성 시간	상위 지표 A 내부의 HU 볼륨 생성 세부 단계	ms	DICOM 내부 세부 단계

신체 마스크 전처리 시간	신체 마스크 생성 및 전처리 시간	ms	전처리
일반·뼈·폐 볼륨 전처리 시간	프리셋별 볼륨 전처리 시간	ms	전처리
VTK 데이터 가져오기 시간	HU 볼륨을 VTK 파이프라인으로 가져오는 시간	ms	렌더링 설정
최초 렌더링 시간	최초 볼륨 렌더링 완료까지의 시간	ms	렌더링
VTK 창 초기화 상위 지표	상위 지표 A 완료 후 VTK 연동부터 최초 렌더링까지	ms	상위 지표 B
프리셋 최초 사용 시 전환 시간	뼈·폐 최초 선택 시 지연 볼륨 생성, 프리셋 적용 및 렌더링 포함	ms	상호작용
캐시 재사용 시 프리셋 전환 시간	생성된 캐시 재사용 시 프리셋 상호작용 지연시간	ms	상호작용
샘플링 간격	품질 모드별 광선 샘플링 간격 설정값	value	렌더링 설정
평균 FPS	Balanced 프리셋 고정 조건에서 프레임 표본 기반 FPS	FPS	렌더링
프레임 시간 중앙값 및 통합 p95	표본 중앙값, 최근접 순위법 기반 Q1·Q3 및 원시 프레임 통합 p95	ms	렌더링
VTK 초기화 직후 작업 집합	VTK 초기화 직후 고정 시점의 작업 집합 측정값	MB	메모리
동일 마스크 기반 즉시 전처리 구조적 메모리	최적화된 ushort 기반 BFS와 VTK 가져오기·복사 버퍼를 포함한 구조적 추정치	MB	합성 데이터 기반 추정
동일 마스크 기반 지연 전처리 구조적 메모리	최적화된 ushort 기반 BFS와 VTK 가져오기·복사 버퍼를 포함한 초기 구조적 추정치	MB	합성 데이터 기반 추정
VTK 초기화 직후 전용 메모리	VTK 초기화 직후 고정 시점의 전용 메모리 측정값	MB	메모리

3.2 동일 마스크 기반 즉시·지연 전처리 조건

지연 전처리 단독 제거 실험에서는 입력 볼륨, BuildBodyMaskOptimized 함수, 일반·뼈·폐 볼륨 생성기 및 자료형을 두 조건에서 동일하게 유지하였다. 동일 마스크 기반 즉시 전처리 조건은 초기화 시 일반·뼈·폐 볼륨을 모두 생성하고, 동일 마스크 기반 지연 전처리 조건은 일반 볼륨만 생성한 뒤 뼈/폐 프리셋을 최초 선택할 때 해당 볼륨을 생성하도록 정의하였다. 두 조건에서 볼륨 생성 시점만 변경하는 비교 실험 구조를 [그림 3]에 제시하였으며, 실제 DICOM 실행 조건과 즉시·지연 전처리 조건의 차이를 [표 2]에 정리하였다. 기존 마스크 기준 조건(Legacy-mask baseline)과 최적화된 마스크 처리 경로(optimized-mask path)의 비교는 복합적인 공학적 비교(engineering comparison)로만 활용하며, 지연 전처리 단독 효과의 근거로 사용하지 않는다.



[그림 3] 동일 마스크 기반 즉시·지연 전처리 제거 실험 구조

[Fig. 3] Same-mask eager and lazy preprocessing ablation structure

[표 2] 동일 마스크 제거 실험 및 실제 DICOM 실행 조건

[Table 2] Same-mask ablation and real-DICOM runtime conditions

조건	초기 전처리	프리셋 생성	캐시	논문에서의 활용
실제 DICOM GUI	최적화된 실행 환경	뼈·폐 최초 사용 시 생성	최초 사용 이후	TCIA 시간·FPS·메모리 측정
동일 마스크 기반 즉시 전처리	최적화된 마스크 + 일반·뼈·폐	모든 볼륨 즉시 생성	해당 없음	지연 전처리 단독 제거 실험의 대조 조건
동일 마스크 기반 지연 전처리	최적화된 마스크 + 일반 볼륨	뼈·폐 최초 사용 시 생성	최초 사용 이후	지연 전처리 단독 제거 실험의 실험 조건

4. 실험

4.1 공개 DICOM GUI 벤치마크 실행 조건

공개 DICOM GUI 벤치마크는 각 데이터셋과 반복 실행의 조합마다 초기화 전용 프로세스(initialization-only process) 1개와 Fast/Balanced/High 품질 모드별 독립적인 전체 실행 프로세스(process)를 사용하였다. 최종 분석에는 데이터셋당 10회(n=10)의 측정 결과만 포함하였으며, 이전 5회(n=5) 측정 결과는 제외하였다. 각 프로세스 실행 직전에는 시리즈 파일을 순차적으로 읽는 웜

캐시 정책을 적용하였다. 데이터셋은 반복 실행별 순환 교차 배치(cyclic interleave)하고, 품질 모드는 시드(seed) 20260818로 재현 가능한 순서를 사용하였다. 워밍업(Warm-up) 실행을 별도로 제외하지 않았으며, 구조적으로 유효한 10개 관측을 모두 포함하고 이상치(outlier)를 제거하지 않았다. 각 품질 모드별 실행은 최소 3초이면서 원시 프레임 100개를 확보할 때까지 측정하되 최대 120초로 제한하였다. 표본 중앙값은 관측값 개수가 짝수일 때 두 중앙 관측값의 평균으로 계산하고, Q1·Q3는 최근접 순위법(nearest-rank)을 사용하였다. 모든 결과에 최솟값(min)과 최댓값(max)을 제시하고, 원시 프레임 9,000개를 이용하여 통합 95백분위 프레임 시간(Pooled p95)을 재계산하였다.

[표 3]은 실험에 사용한 공개 CT DICOM 데이터셋의 크기와 복셀 수를 정리한 것이다. 세 시리즈는 TCIA 운영 및 데이터 이용 정책을 참고하고 [11], 각 데이터셋의 인용 문헌(citation)을 참고하여 사용하였다 [12-14]. SeriesInstanceUID, 간격 정보, 영상 크기(dimensions), 복셀 수 및 전송 구문(transfer syntax)은 로컬 목록 파일(local manifest)과 DICOM 헤더(header)를 대조하였다. RIDER 및 Interobserver 메타데이터의 재현성 관련 정보와 LIDC-IDRI 컬렉션의 특성은 선행 연구를 참고하였다 [15][16].

[표 3] 공개 CT DICOM 데이터셋 특성

[Table 3] Public CT DICOM dataset characteristics.

크기	출처	데이터셋	불러온 볼륨	복셀 수
Small	TCIA	NSCLC-RADIOMICS-INTEROBSERVER1 CT	512×512×154	40,370,176
Medium	TCIA	RIDER-LUNG-CT	512×512×257	67,371,008
Large	TCIA	LIDC-IDRI CT	512×512×764	200,278,016

품질 모드별 정지 상태(still), 상호작용 상태(interactive) 및 슬라이더 조작 상태(slider)의 SampleDistance는 Fast 1.20/2.50/3.00, Balanced 0.50/1.50/2.00, High 0.35/1.00/1.50으로 고정하였다. 프레임 벤치마크의 활성 프리셋은 모든 품질 모드에서 Balanced로 고정하여 품질 설정 매개변수(quality parameter) 외의 프리셋 차이를 배제하였다.

통합 95백분위 프레임 시간은 조건별 원시 프레임 N개를 오름차순 정렬한 뒤 올림 함수 ceil(0.95N)번째 관측값을 선택하는 최근접 순위법으로 정의하였다. 95% 신뢰구간은 독립 단위인 10개 실행을 복원추출하고 선택된 실행의 원시 프레임을 합쳐 통합 95백분위값을 다시 계산하는 실행 단위 군집 부트스트랩 1,000회로 산출하였다. 기본 난수 시드 20260818에 데이터셋과 품질 모드의 조합별 결정적 오프셋을 더하였으며, 조건별 난수 시드 값은 공개 결과 패키지에 기록하였다 [7].

4.2 통제된 합성 벤치마크

통제된 합성 벤치마크(controlled synthetic benchmark)는 결정적(deterministic)으로 생성한 볼륨

128×128×96, 192×192×128 및 256×256×160을 사용하였다. Release 빌드에서 워밍업 3회 후 조건별로 30회 반복하였다. VTK/GPU 렌더링은 제외하고 전처리 시간만 측정했으며, 동일 마스크 기반 즉시·지연 전처리 조건은 생성 시점 외의 마스크 생성 방식, 자료형 및 볼륨 생성기를 동일하게 유지하였다.

5. 결과 및 고찰

5.1 실제 DICOM 최적화 실행 결과

[표 4]는 공개 결과 패키지의 초기화 요약 데이터와 실행 결과 통합 데이터만을 사용하여 작성하였다 [7].

[표 4] 웜 캐시 V2 초기화의 상위 메트릭 및 고정 메모리 스냅샷

[Table 4] Warm-cache V2 initialization parent metrics and fixed-memory snapshots (median [IQR] / min – max, n=10)

데이터셋	DICOM 폴더 입력부터 2D MPR 준비까지의 시간(ms) 중앙값[사분위범위] / 최솟값~최댓값	VTK 초기화 시간(ms) 중앙값[사분위범위] / 최솟값~최댓값	작업 집합(MB) 중앙값[사분위범위] / 최솟값~최댓값	전용 메모리(MB) 중앙값[사분위범위] / 최솟값~최댓값	n
Small	533 [526 – 538] 524 – 569	848 [829 – 853] 813 – 876	762 [761 – 768] 758 – 772	825 [825 – 832] 821 – 838	10
Medium	848 [841 – 871] 831 – 904	2153 [2124 – 2212] 2110 – 2724	1098 [1095 – 1228] 1090 – 1230	1178 [1173 – 1307] 1171 – 1311	10
Large	2345 [2290 – 2409] 2268 – 2505	5310 [5118 – 5518] 4892 – 6185	3144 [3120 – 3146] 3113 – 3248	3230 [3210 – 3236] 3195 – 3324	10

웜 캐시 재측정에서 DICOM 폴더 입력부터 2D MPR 준비까지의 시간과 VTK 초기화 시간의 표본 중앙값[사분위범위(IQR)] 및 min – max는 Small에서 각각 533 [526 – 538], 524 – 569 / 848 [829 – 853], 813 – 876 ms였다. Medium에서는 848 [841 – 871], 831 – 904 / 2,153 [2,124 – 2,212], 2,110 – 2,724 ms였으며, Large에서는 각각 2,345 [2,290 – 2,409], 2,268 – 2,505 / 5,310 [5,118 – 5,518], 4,892 – 6,185 ms였다. 측정 범위는 IQR 밖의 느린 실행도 포함하였으며, 관찰된 변동 원인을 운영체제 스케줄러(OS scheduler) 또는 백그라운드 부하(background load) 중 하나로 단정하지 않았다.

5.2 품질 모드별 렌더링 성능

[표 5]에서 Fast FPS 표본 중앙값은 Small 24.21, Medium 13.43, Large 4.93이었고, 통합 95백분위 프레임 시간은 각각 52, 92, 253 ms였다. Large 데이터셋의 High 모드에서는 통합 95백분위 프레임

시간이 522 ms였으며 실행별 95백분위값의 범위는 423 – 698 ms였다. 기존 3초 고정 측정에서 95백분위값이 7 – 10 프레임의 사실상 최대값에 가까워지는 문제를 피하기 위해 모든 실행에서 원시 프레임 100개를 확보하였다. FPS는 각 실행에서 산출한 평균 FPS의 표본 중앙값으로 제시하였으며, 프레임 시간은 실행별 프레임 시간 통계량의 표본 중앙값으로 제시하였다. 두 값은 서로 다른 통계량이므로 FPS 중앙값의 역수와 프레임 시간 중앙값이 반드시 일치하지는 않는다.

[표 5] Balanced 프리셋 고정 조건의 프레임 성능 및 원시 프레임 통합 p95 성능

[Table 5] Fixed-Balanced-preset frame performance with raw-frame pooled p95 (n=10; 100 frames/run)

데이터셋	품질 모드	n	FPS 중앙값[사분위범위] / 최솟값~최댓값	프레임 시간(ms) 중앙값[사분위범위] / 최솟값~최댓값	95백분위 프레임 시간(ms) 통합값 / 실행별 중앙값[사분위범위] / 실행별 최솟값~최댓값 / 부트스트랩 신뢰구간	실행당 프레임 표본 수 중앙값[사분위범위] / 최솟값~최댓값
Small	Fast	10	24.21 [22.57 – 25.21] 21.52 – 26.48	37 [35 – 45] 34 – 49	pooled 52 run52[51 – 53] run50 – 55 CI52 – 53	100 [100 – 100] 100 – 100
Small	Balanced	10	13.05 [12.92 – 13.40] 11.97 – 13.49	77 [75 – 79] 73 – 82	pooled 89 run86[85 – 87] run84 – 103 CI87 – 91	100 [100 – 100] 100 – 100
Small	High	10	10.05 [9.55 – 10.51] 9.14 – 10.62	99 [92 – 104] 91 – 108	pooled 118 run113[109 – 119] run104 – 126 CI116 – 120	100 [100 – 100] 100 – 100
Medium	Fast	10	13.43 [12.38 – 14.06] 12.08 – 15.08	73 [67 – 80] 61 – 83	pooled 92 run89[86 – 93] run82 – 124 CI90 – 93	100 [100 – 100] 100 – 100
Medium	Balanced	10	8.79 [8.41 – 9.14] 7.38 – 9.41	110 [105 – 119] 101 – 131	pooled 142 run130[125 – 144] run123 – 180 CI138 – 147	100 [100 – 100] 100 – 100
Medium	High	10	7.13 [6.74 – 7.40] 6.53 – 7.56	139 [131 – 145] 128 – 150	pooled 170 run160[154 – 179] run149 – 188 CI166 – 175	100 [100 – 100] 100 – 100
Large	Fast	10	4.93 [4.62 – 5.21] 4.12 – 5.38	202 [188 – 211] 182 – 215	pooled 253 run232[217 – 260] run202 – 424 CI245 – 261	100 [100 – 100] 100 – 100

Large	Balanced	10	2.79 [2.75 – 2.87] 2.66 – 2.98	349 [343 – 357] 307 – 376	pooled 418 run413[385 – 429] run371 – 491 CI407 – 429	100 [100 – 100] 100 – 100
Large	High	10	2.45 [2.29 – 2.52] 2.08 – 2.55	400 [390 – 432] 380 – 454	pooled 522 run482[441 – 537] run423 – 698 CI501 – 538	100 [100 – 100] 100 – 100

5.3 프리셋 지연 처리 및 캐시 적용 방식의 전환 비용

[표 6]은 공개 결과 패키지의 프리셋 전환 이벤트 데이터 중 Fast 품질 모드의 측정 결과만을 사용하여 작성하였다 [7]. Bone 프리셋의 최초 사용 시 지연시간 표본 중앙값은 Small, Medium, Large에서 각각 175ms, 296ms, 770ms였으며, 캐시 재사용 시에는 각각 25ms, 30ms, 28ms로 측정되었다. 최초 사용 시 지연시간에는 지연 볼륨 생성과 프리셋 적용, 볼륨 전환 및 렌더링이 포함되므로, 최초 사용과 캐시 재사용 간의 차이를 순수한 전처리 시간의 차이로 해석할 수는 없다.

[표 6] Fast 모드 최초 사용 및 캐시 프리셋 전환 지연시간

[Table 6] Fast-mode first-use and cached preset-switch latency (median [IQR] and min – max, n=10).

데이터셋	프리셋	상태	n	중앙값(ms)[사분위범위]	최솟값-최댓값(ms)
Small	Bone	최초 사용	10	175 [173 – 178]	170 – 185
Small	Bone	캐시 재사용	10	25 [23 – 28]	23 – 29
Small	Lung	최초 사용	10	220 [211 – 232]	204 – 258
Small	Lung	캐시 재사용	10	33 [27 – 37]	27 – 38
Medium	Bone	최초 사용	10	296 [293 – 301]	287 – 317
Medium	Bone	캐시 재사용	10	30 [27 – 32]	26 – 36
Medium	Lung	최초 사용	10	337 [330 – 347]	325 – 354
Medium	Lung	캐시 재사용	10	47 [43 – 55]	40 – 58
Large	Bone	최초 사용	10	770 [766 – 816]	755 – 861
Large	Bone	캐시 재사용	10	28 [27 – 29]	25 – 34
Large	Lung	최초 사용	10	943 [921 – 955]	885 – 999
Large	Lung	캐시 재사용	10	94 [89 – 103]	85 – 110

5.4 동일 마스크 기반 지연 전처리 제거 실험

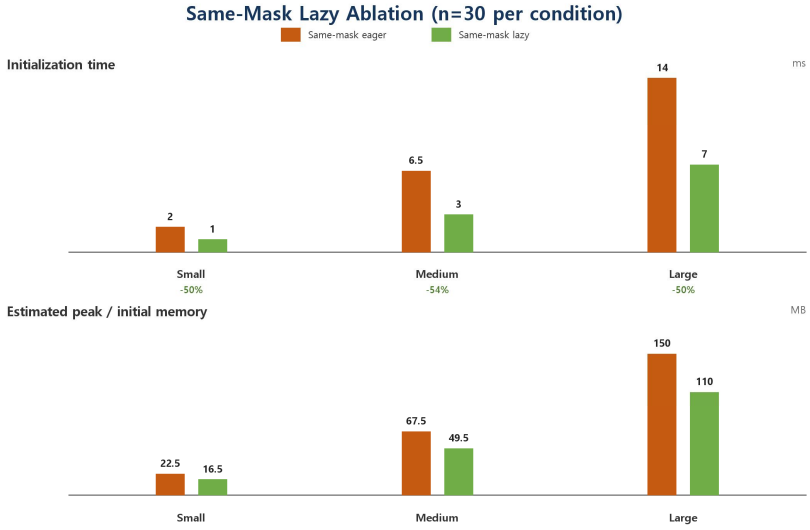
동일 마스크 기반 즉시·지연 전처리의 초기화 시간, 시간 감소율 및 구조적 메모리 추정치를 [표

7]에 제시하였다.

[표 7] 동일 마스크 기반 즉시·지연 전처리 제거 실험 및 VTK 가져오기·복사 버퍼를 포함한 구조적 메모리 추정
[Table 7] Same-mask eager versus lazy preprocessing ablation and structural memory estimate including the VTK import-copy buffer (warm-up 3; n=30 per condition).

데이터셋	동일 마스크 기반 즉시 전처리 시간(ms) 중앙값[사분위범위] / 최솟값~최댓값	동일 마스크 기반 지연 전처리 시간(ms) 중앙값[사분위범위] / 최솟값~최댓값	초기화 시간 감소율	즉시 전처리 구조적 메모리(MB)	지연 전처리 구조적 메모리(MB)
synthetic-small	2 [2 – 4] 2 – 16	1 [1 – 2] 1 – 4	50.00%	22.5	16.5
synthetic-medium	6.5 [6 – 8] 5 – 9	3 [3 – 4] 2 – 6	53.85%	67.5	49.5
synthetic-large	14 [13 – 15] 12 – 22	7 [7 – 8] 6 – 14	50.00%	150.0	110.0

동일 마스크 기반 제거 실험에서 지연 초기화 시간은 즉시 전처리 대비 합성 소형 데이터(synthetic-small)에서 2→1 ms(50.00%), 합성 중형 데이터(synthetic-medium)에서 6.5→3 ms(53.85%), 합성 대형 데이터(synthetic-large)에서 14→7 ms(50.00%)로 감소하였다. 두 전처리 방식의 초기화 시간과 구조적 메모리 차이를 [그림 4]에 시각적으로 나타내었다.



[그림 4] 동일 마스크 기반 즉시·지연 전처리의 초기화 시간 및 구조적 메모리 비교
[Fig. 4] Comparison of Initialization Time and Structural Memory Between Same-Mask Eager and Lazy Preprocessing

두 조건은 BuildBodyMaskOptimized, ushort[] 기반 너비 우선 탐색(BFS) 상태, 자료형 및 볼륨 생성기를 동일하게 유지하고 초기 뼈·폐 볼륨의 생성 시점만 다르므로, 해당 감소율은 합성 전처리 범위에서 지연 생성 전략의 효과로 해석할 수 있다. 메모리 값은 입력 데이터·마스크·볼륨 및 VTK 가져오기·복사 버퍼(import-copy buffer)를 포함한 구조적 추정치로, 실제 프로세스의 작업 집합을 의미하지 않는다. 밀리초 해상도 스톱워치와 합성 볼륨을 사용했으므로 이 결과를 실제 DICOM GUI 전체 초기화 시간의 개선율로 일반화할 수 없다.

6. 결론

본 연구는 WPF/ActiViz.NET/VTK 기반 독립형 DICOM CT 뷰어의 성능을 DICOM 폴더 입력부터 2D MPR 준비까지의 시간, VTK 초기화, 최초 사용 및 캐시 재사용 시 프리셋 전환 지연시간, 원시 프레임 기반 렌더링 성능으로 분해하여 평가하였다. 실험에 사용한 Git 커밋과 실행 파일의 SHA-256을 고정한 Release|x64 실험에서 초기화 기록 30개, 프레임 요약 90개, 프리셋 이벤트 360개 및 원시 프레임 기록 9,000개가 무결성 오류 없이 병합되었다.

실험 결과, 첫째, 데이터셋 크기가 증가함에 따라 DICOM 폴더 입력부터 2D MPR 준비까지의 시간과 VTK 초기화 시간이 모두 증가하였으며, 특히 VTK 초기화 시간의 증가 폭이 크게 나타났다. 둘째, 품질 모드에 따라 FPS와 꼬리 지연시간에 차이가 발생하였으며, Large 데이터셋에서 통합 95백분위 프레임 시간은 Fast 모드 253ms에서 High 모드 522ms로 증가하였다. 셋째, 동일 마스크 기반 합성 비교 실험에서 지연 전처리는 즉시 전처리 대비 초기화 시간을 50.00~53.85% 감소시켜 지연 생성 전략의 효과를 확인하였다.

다만 웜 캐시 조건의 측정 결과는 특정 콜드 스타트 환경을 대표하지 않으며, 본 연구의 관찰 범위와 하드웨어에 한정된다. 동일 마스크 합성 비교 실험은 지연 생성 시점의 효과를 분리한 것으로, 측정된 감소율이 실제 DICOM GUI 전체 초기화 시간의 개선율을 의미하지 않는다.

References

[1] National Electrical Manufacturers Association, "Digital Imaging and Communications in Medicine (DICOM), PS3.3 Information Object Definitions and PS3.5 Data Structures and Encoding," *dicomstandard.org*, https://www.dicomstandard.org/current/ (accessed Aug. 18, 2026).

[2] Menghe Zhang, Weichen Liu, Nadir Weibel, and Jürgen P. Schulze, "A DirectX-Based DICOM Viewer for Multi-User Surgical Planning in Augmented Reality," in *Advances in Visual Computing*, San Diego, USA, Oct. 3-5, 2022, pp. 217-228, doi: 10.1007/978-3-031-20716-7_17.

[3] Hugo Pereira, Luis Romero, and Pedro Miguel Faria, "Web-Based DICOM Viewers: A Survey and a Performance Classification," *Journal of Imaging Informatics in Medicine*, vol. 38, no. 3, pp. 1304-1322, Jun.

2025, doi: 10.1007/s10278-024-01216-5.

[4] Jagjot Singh Wadali, Sanjay P. Sood, Rajesh Kaushish, Shabbir Syed-Abdul, Praveen K. Khosla, and Madhu Bhatia, "Evaluation of Free, Open-source, Web-based DICOM Viewers for the Indian National Telemedicine Service (eSanjeevani)," *Journal of Digital Imaging*, vol. 33, no. 6, pp. 1499-1513, Dec. 2020, doi: 10.1007/s10278-020-00368-4.

[5] Tseng-Tse Chen, Ying-Chou Sun, Woei-Chyn Chu, and Chung-Yueh Lien, "BlueLight: An Open Source DICOM Viewer Using Low-Cost Computation Algorithm Implemented with JavaScript Using Advanced Medical Imaging Visualization," *Journal of Digital Imaging*, vol. 36, no. 2, pp. 753-763, Apr. 2023, doi: 10.1007/s10278-022-00746-0.

[6] Microsoft, "Windows Presentation Foundation documentation," *learn.microsoft.com*, https://learn.microsoft.com/dotnet/desktop/wpf/ (accessed Aug. 18, 2026).

[7] Nahyeon Kwon, "DICOM Viewer and Anonymization Quality Verification Tool: paper reproducibility packag e V3, commit 57039de5437f224ca22e0ec360ee4d894770cddf," *github.com*, https://github.com/Kwon-nahyeon/He llo/tree/paper-reproducibility-v3.1-20260818 (accessed Aug. 18, 2026).

[8] Fellow Oak DICOM, "fo-dicom.Desktop 4.0.8," *nuget.org*, https://www.nuget.org/packages/fo-dicom.Desktop/4.0.8 (accessed Aug. 18, 2026).

[9] Will Schroeder, Ken Martin, and Bill Lorensen, *The Visualization Toolkit: An Object-Oriented Approach to 3D Graphics*, 4th ed., Kitware, Inc., 2006.

[10] Kitware, "ActiViz.NET.x64 5.8.0," *nuget.org*, https://www.nuget.org/packages/ActiViz.NET.x64/5.8.0 (accessed Aug. 18, 2026).

[11] Kenneth Clark et al., "The Cancer Imaging Archive (TCIA): Maintaining and Operating a Public Information Repository," *Journal of Digital Imaging*, vol. 26, no. 6, pp. 1045-1057, Dec. 2013, doi: 10.1007/s10278-013-9622-7.

[12] Leonard Wee, Hugo J. W. L. Aerts, Petros Kalendralis, and Andre Dekker, "Data from NSCLC-Radiomics-Interobserver1," *cancerimagingarchive.net*, https://doi.org/10.7937/TCIA.2019.CWVLPD26 (accessed Aug. 18, 2026).

[13] Binsheng Zhao, Lawrence H. Schwartz, Mark G. Kris, and Gregory J. Riely, "Coffee-break lung CT collection with scan images reconstructed at multiple imaging parameters, Version 3," *cancerimagingarchive.net*, https://doi.org/10.7937/K9/TCIA.2015.U1X8A5NR (accessed Aug. 18, 2026).

[14] Samuel G. Armato III et al., "Data From LIDC-IDRI," *cancerimagingarchive.net*, https://doi.org/10.7937/K9/TCIA.2015.LO9QL9SX (accessed Aug. 18, 2026).

[15] Petros Kalendralis et al., "FAIR-compliant clinical, radiomics and DICOM metadata of RIDER, interobserver, Lung1 and head-Neck1 TCIA collections," *Medical Physics*, vol. 47, no. 11, pp. 5931-5940, Nov. 2020, doi: 10.1002/mp.14322.

[16] Samuel G. Armato III et al., "The Lung Image Database Consortium (LIDC) and Image Database Resource Initiative (IDRI): A Completed Reference Database of Lung Nodules on CT Scans," *Medical Physics*, vol. 38, no. 2, pp. 915-931, Feb. 2011, doi: 10.1118/1.3528204.