

3D 그래픽스, 셰이더, OpenGL

3D Graphics Programming with OpenGL Shaders

biztripcru@gmail.com

© 2022–2024. biztripcru@gmail.com. All rights reserved.
모든 저작권은 biztripcru@gmail.com 에게 있습니다.

그래픽스 시스템

Graphics System

Contents 내용

- 컴퓨터 그래픽스의 정의
- 이미지 생성
- 합성 카메라 모델
- 래스터 시스템
- 그래픽스 아키텍처

컴퓨터 그래픽스의 정의

Definition of Computer Graphics

컴퓨터 그래픽스 Computer Graphics

- 모호하게 ambiguously 사용될 수 있음
- from Wikipedia,
 - (컴퓨터를 이용한) 그래픽 디자인 (또는 결과물)
 - ▶ 포토샵, 일러스트레이터 사용법 강의
 - a.k.a CG → 컴퓨터 애니메이션 (또는 결과물)
 - ▶ 3D Studio Max, Maya 사용법 강의
 - **컴퓨터 과학의 컴퓨터 그래픽스 분야**
 - ▶ 컴퓨터 프로그래밍 강의

https://commons.wikimedia.org/wiki/File:Photo_of_Peter_M_Lawrence_MS_medical_illustrator_and_researcher_at_Barrow_Neurological_Institute_2019.jpg



<https://pixabay.com/photos/toy-story-%EC%95%84%EC%9D%B4%EB%93%A4%EC%9D%9B-%ED%8C%8C%ED%8B%B0-2375242/>



컴퓨터 그래픽스 Computer Graphics

- 컴퓨터 과학 computer science **에서의 정의** definition
 - 컴퓨터를 이용해서 이미지/애니메이션을 만들어내는 **모든 기술**
 - **all technologies** related to producing **images** and **animations** (or sequences of images) using a **computer**
 - ▶ modified from "Interactive Computer Graphics", by Edward Angel

컴퓨터 그래픽스, 궁극의 목표

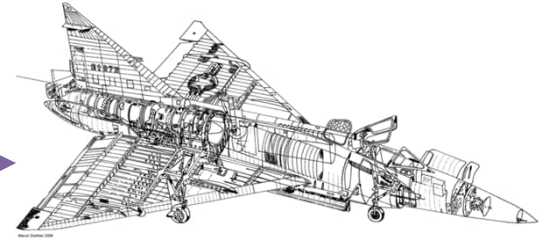
- **현실 세계** real world **의 시뮬레이션** simulation

- 실제/가상 물체의 모델링
- 현실 세계의 빛 – 반사, 굴절, 회절
- 물리학/광학의 시뮬레이션
- 인간의 시각/카메라 시뮬레이션

- **어떻게 시뮬레이션** simulation **할 것인가?**

- 물리학/광학/수학 방정식 equation
- 또는 근사 approximation 기법들
 - ▶ 예를 들어, 텍스처 매핑 texture mapping

CC BY 2.5, 2022.01.01
https://commons.wikimedia.org/wiki/File:F-102_Delta_Dagger_wireframe.png



슈퍼컴퓨터

CC BY-SA 2.0
https://commons.wikimedia.org/wiki/File:IBM_Blue_Gene_P_supercomputer.jpg

public domain, 2022.01.01
https://en.wikipedia.org/wiki/Convair_F-102_Delta_Dagger#/media/File:Convair_YF-102_FC-782.jpg



이미지 생성

Image Synthesis

이미지 생성

- **현실 세계** real world **의 시뮬레이션** simulation

- 실제/가상 물체의 모델링
- 물리학/광학의 시뮬레이션
- 인간의 시각/카메라 시뮬레이션

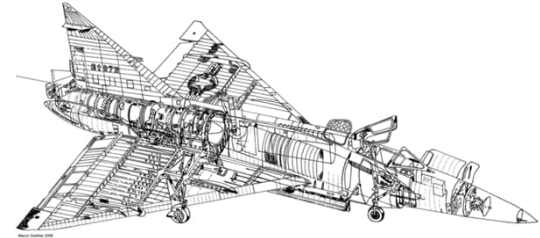
- **필요한 이론 ?**

- 기하학 geometry, 삼각측량학 trigonometry

- **시뮬레이션 대응관계**

- 실세계 real world ⇄ **3차원 공간** 3D space
- 실제 물체 objects ⇄ **오브젝트** objects
- 카메라 camera ⇄ **뷰어** viewer

CC BY 2.5, 2022.01.01
https://commons.wikimedia.org/wiki/File:F-102_Delta_Dagger_wireframe.png



슈퍼컴퓨터

CC BY-SA 2.0
https://commons.wikimedia.org/wiki/File:IBM_Blue_Gene_P_supercomputer.jpg

public domain, 2022.01.01
https://en.wikipedia.org/wiki/Convair_F-102_Delta_Dagger#/media/File:Convair_YF-102_FC-782.jpg

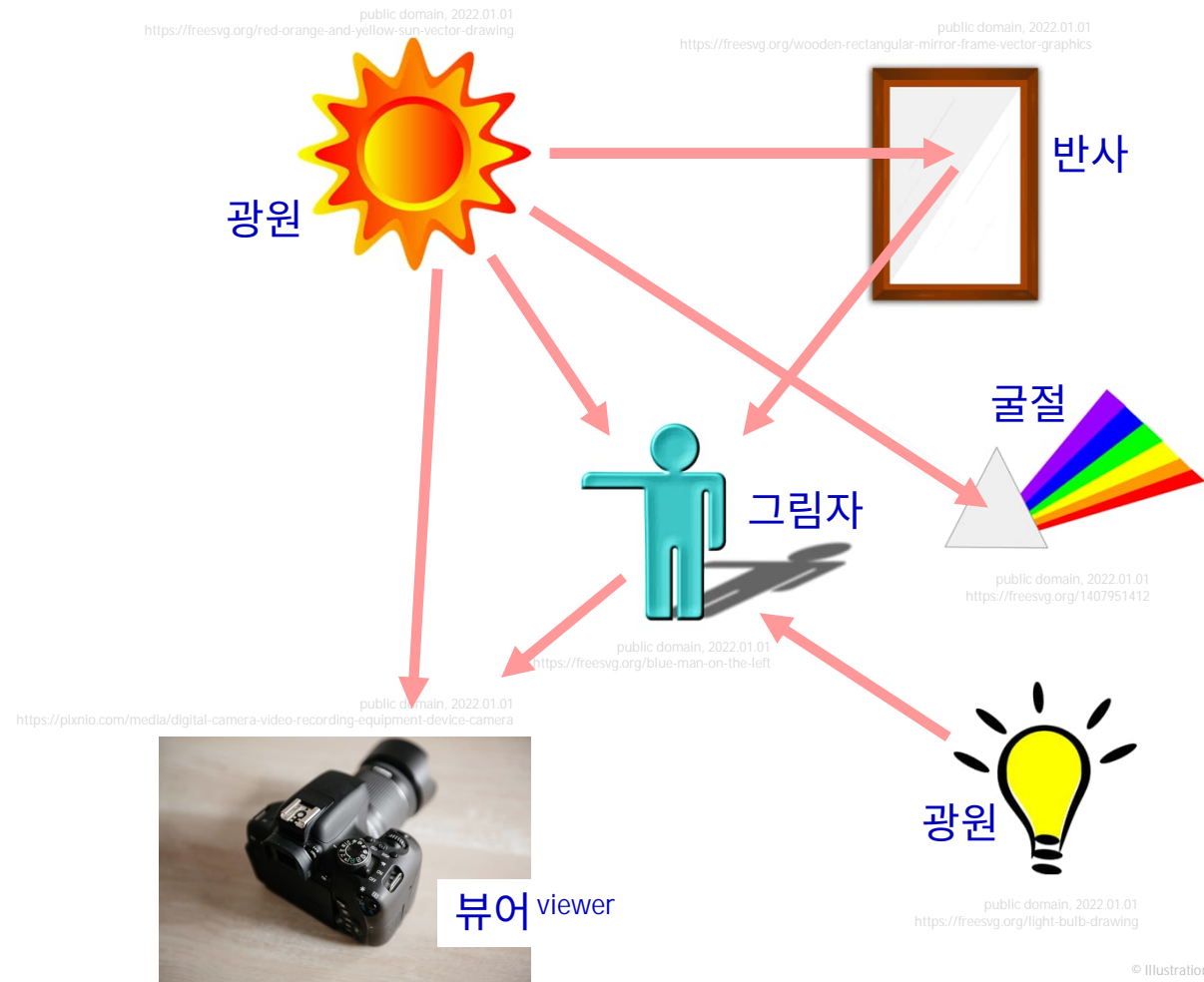


Illustration by biztripcru@gmail.com

광학 시뮬레이션

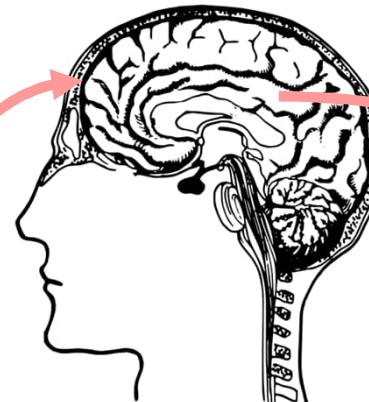
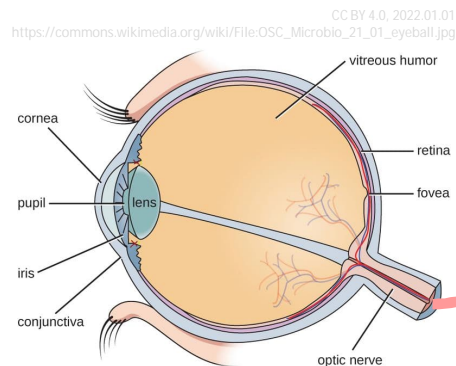
- (고전) 광학 classic optics

- 광원 light source
- 빛의 반사 reflection
- 굴절 refraction
- 회절 diffraction
- . . .



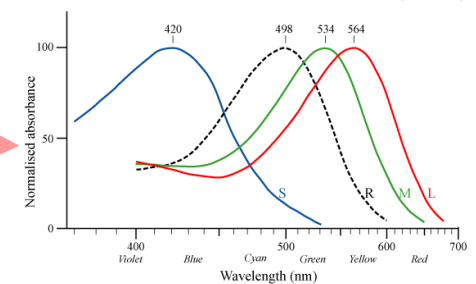
인간 시각 시스템 human visual system

- 망막 : 24 FPS frame per second **정도의 성능**
 - 간상 세포 rod cell : 야간에 민감, 흑백
 - 원추 세포 cone cell : 주간에 민감, 색상 구별
- 시신경 + 뇌 : 후처리로 고화질 영상 획득
 - 고성능 인공지능 필요! → 시뮬레이션 불가능 ...
- 대안 → **카메라를 시뮬레이션!**



CC0 1.0, 2022.01.01
https://freesvg.org/human-brain-diagram-vector-image

CC BY-SA 3.0, 2022.01.01
https://ko.wikipedia.org/wiki/%EC%9B%90%EC%B6%94%EC%84%B8%ED%8F%AC#/media/%ED%8C%8C%EC%9D%BC-Cone-response-en.png



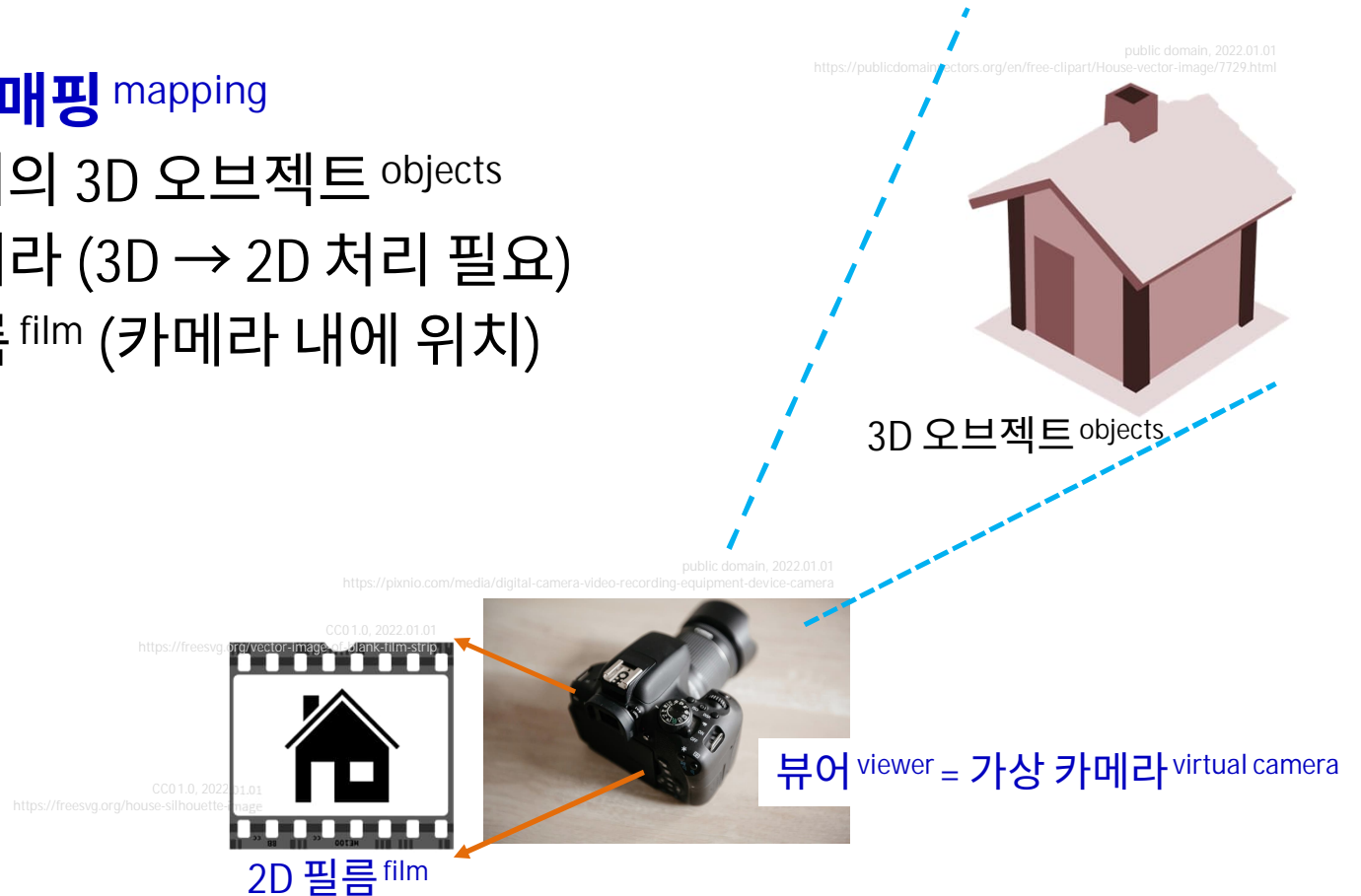
반응 그래프

© Illustration by biztripcru@gmail.com

카메라 모델

- 카메라 모델 : 3D → 2D 매핑 mapping

- 3차원 공간 3D space 내의 3D 오브젝트 objects
- 뷰어 viewer : 가상 카메라 (3D → 2D 처리 필요)
- 이미지 image : 2D 필름 film (카메라 내에 위치)

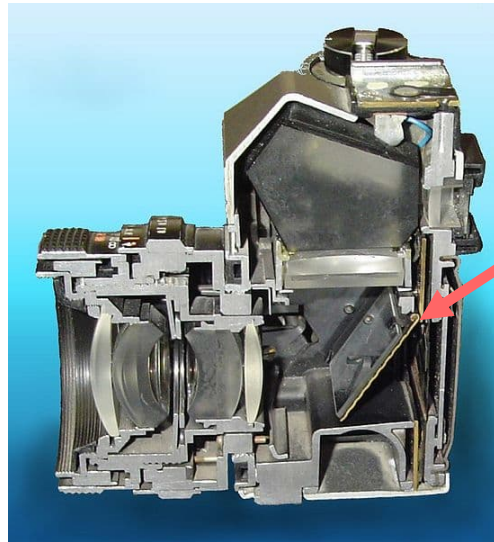


SLR, DSLR 카메라

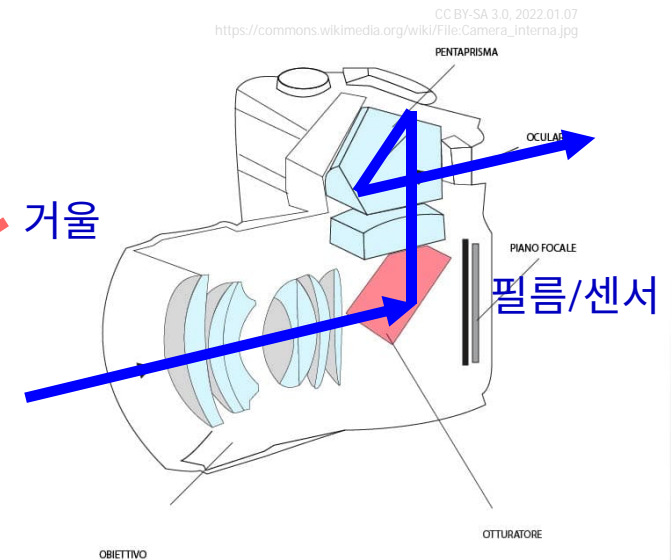
- SLR = single lens (assembly) reflex (camera)
 - reflex : 빛의 반사^{reflection} 구조, 거울이 필요 → 거울을 닫으면^{shut} 필름에 촬영
- DSLR = digital SLR (camera)



pixabay license, 2022.01.07
<https://pixabay.com/ko/photos/%eb%94%94%ec%a7%80%ed%84%b8-%ec%b9%b4%eb%a9%94%eb%9d%bc-%ec%a0%95%ea%b2%bd-%ec%97%90%ec%98%a4%ec%8a%a4-4134671/>



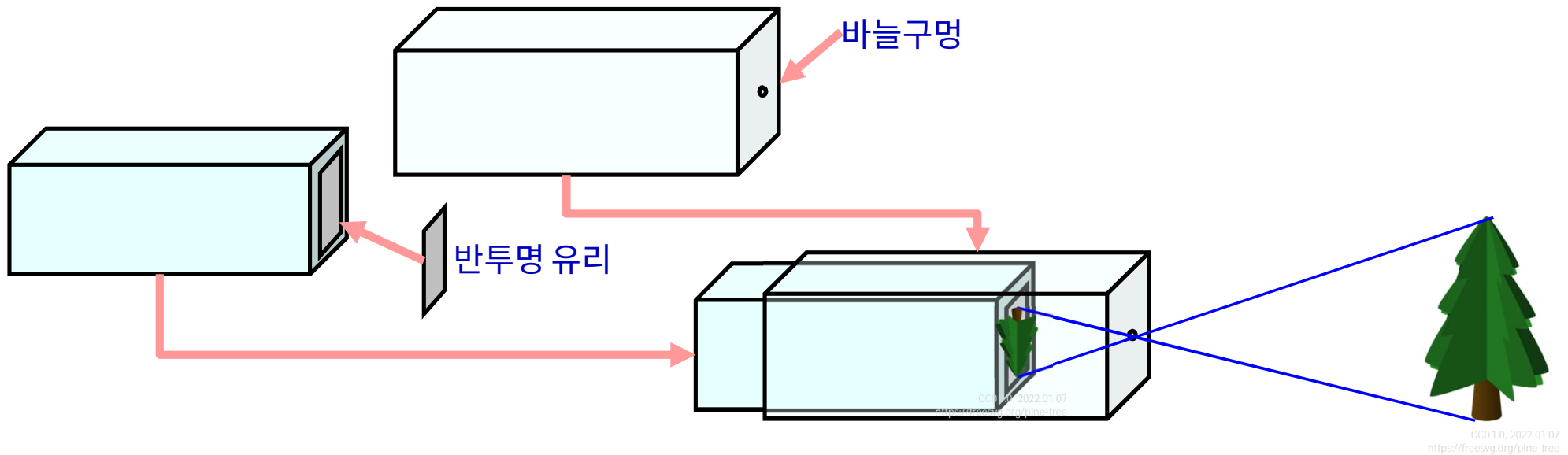
GNU free document license, 2022.01.07
<https://commons.wikimedia.org/wiki/File:Pentaprisma-kol.jpg>



- 장점: 고품질 이미지 가능
- 단점: 렌즈는 어떻게 시뮬레이션???

바늘구멍 사진기 pinhole camera

- **핀홀** pinhole **카메라** : 가장 간단한 구조
 - 단점: 이미지가 뒤집어져서 보인다



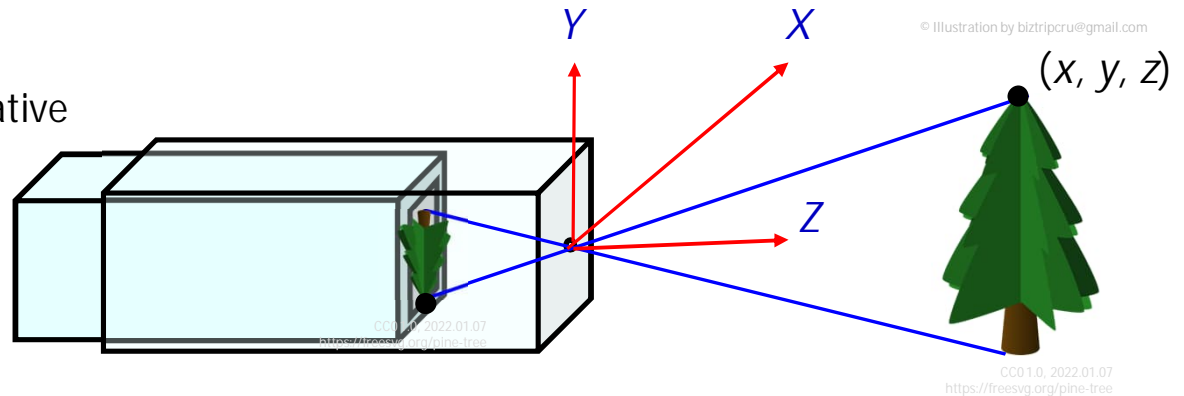
합성 카메라 모델

Synthetic Camera Model

바늘구멍 카메라의 수학적 해석

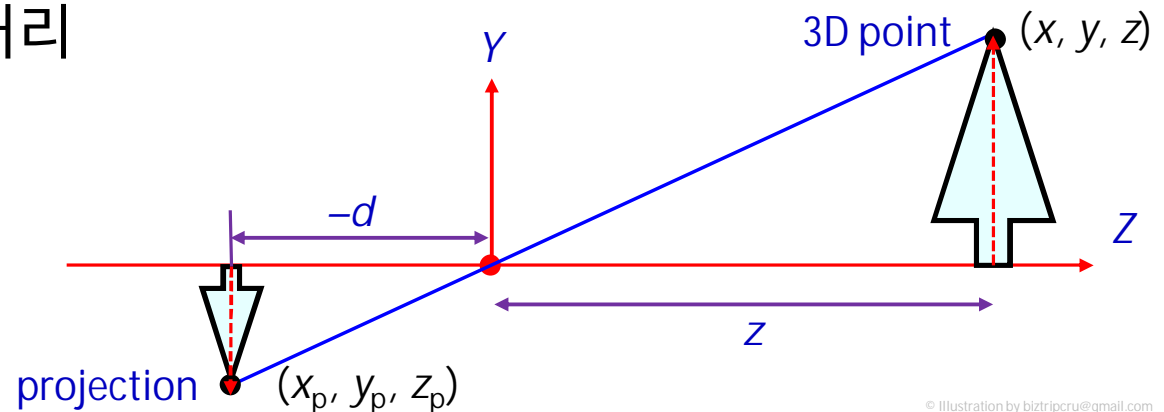
- 가장 간단한 카메라 모델

- 인간 시각 시스템의 대안 alternative



- 비례식 해석

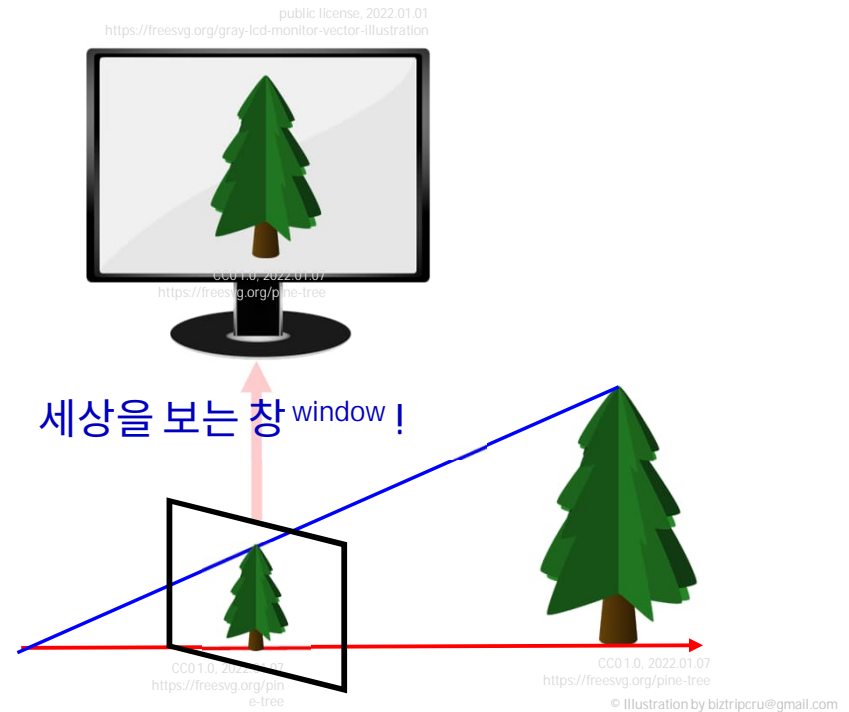
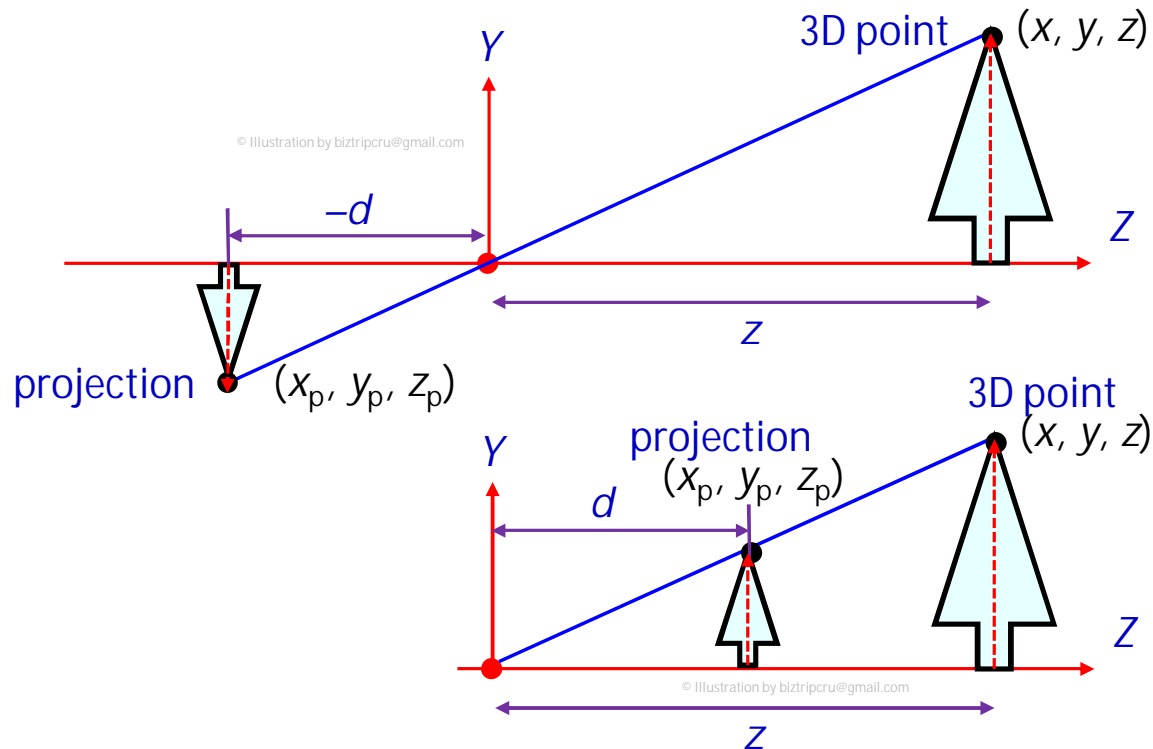
- z = 원점에서 물체까지의 거리
- $z_p = -d$: 원점에서 필름까지의 거리
- $z_p : z = x_p : x \rightarrow x_p = -\frac{x}{z/d}$
- $z_p : z = y_p : y \rightarrow y_p = -\frac{y}{z/d}$



컴퓨터 그래픽스 카메라 모델

● 바늘구멍 카메라 모델의 변형

- 필름이 원점 앞으로 이동 → 컴퓨터 스크린, 윈도우!

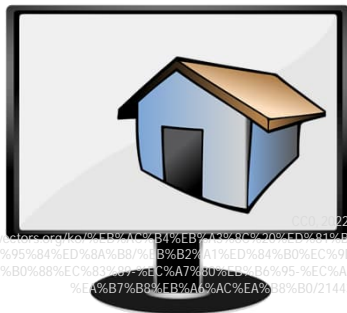


컴퓨터 그래픽스 카메라 모델

- **프로젝션** projection = 투사, 투영
 - 3차원 물체 → 2차원 화면

your screen

public license, 2022.01.01
<https://freemvg.org/gray-lcd-monitor-vector-illustration>



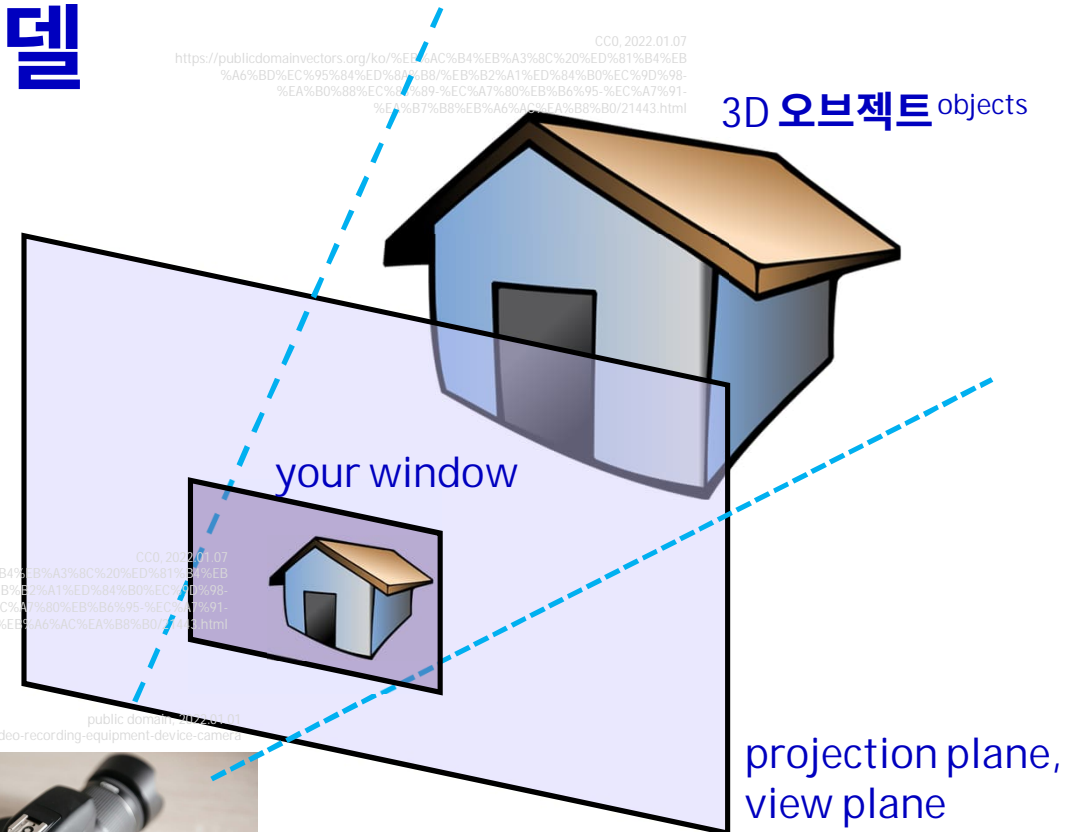
CC0, 2022.01.07
<https://publicdomainvectors.org/ko/%EB%A6%BD%EC%95%84%ED%8A%B8/%EB%A1%ED%84%B0%EC%9D%98-%EA%B0%88%EC%83%B9-%EC%A7%B0%EB%B6%95-%EC%A7%91-%EA%B7%B8%EB%A6%AC%EA%B8%B0/21443.html>

center of projection



뷰어 viewer = 가상 카메라 virtual camera

3D 오브젝트 objects



CC0, 2022.01.07
<https://publicdomainvectors.org/ko/%EB%A6%BD%EC%95%84%ED%8A%B8/%EB%A1%ED%84%B0%EC%9D%98-%EA%B0%88%EC%83%B9-%EC%A7%B0%EB%B6%95-%EC%A7%91-%EA%B7%B8%EB%A6%AC%EA%B8%B0/21443.html>

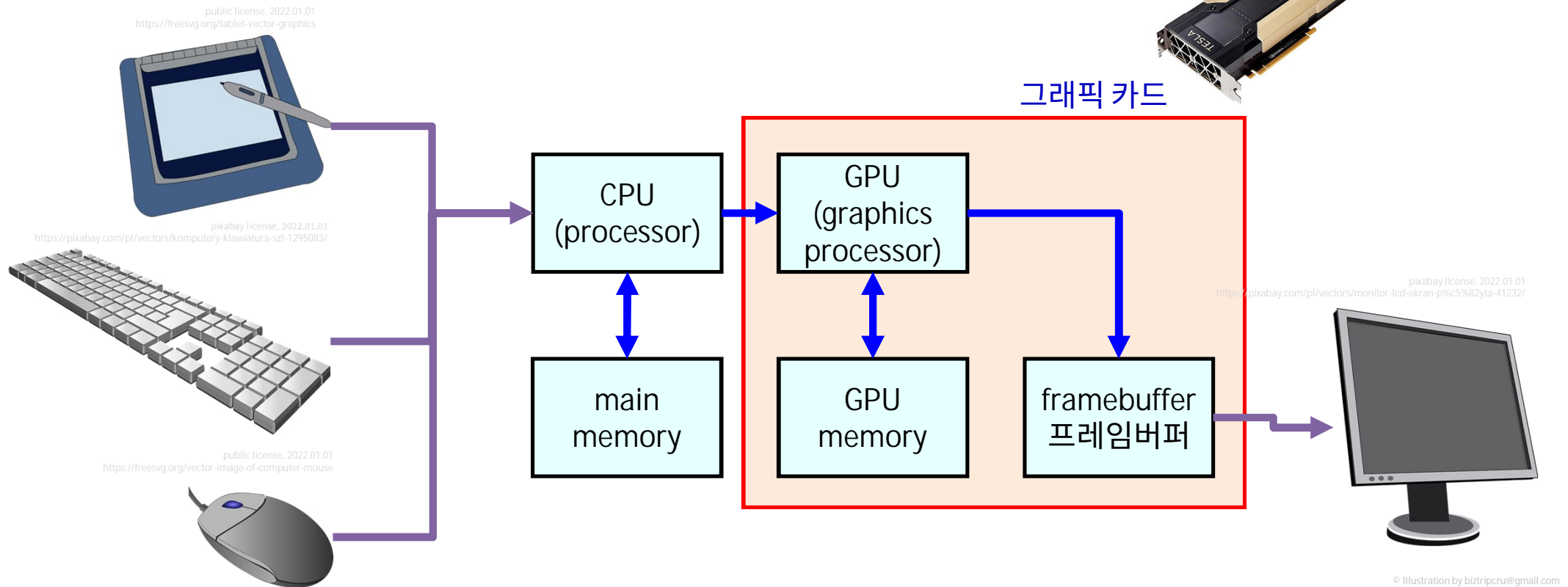
래스터 시스템

Raster System

컴퓨터에 이미지를 어떻게 저장할 것인가?

그래픽스 시스템

- 이미지 image 는 프레임버퍼 framebuffer 에 저장



이미지^{image} 저장 방법?

- 래스터^{raster} 방식

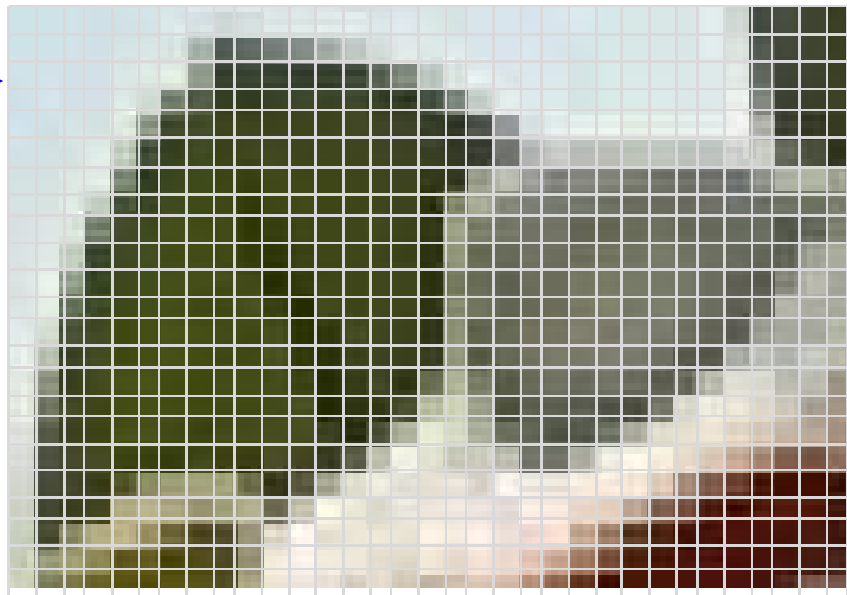
- 이미지 → (색깔 있는) 사각형의 2차원 배열 = 2D array of **colored rectangles**
- **pixel** 픽셀 = picture element = a single colored rectangle in the image

© photograph by biztripcru@gmail.com

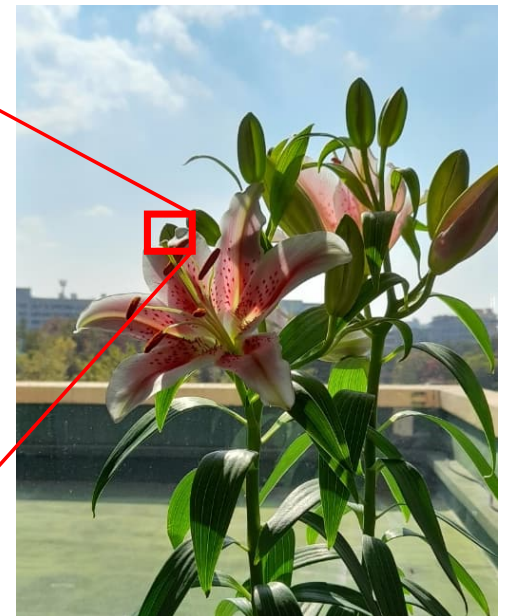


(실생활 속의) 픽셀 방식
(엘리베이터 표시 화면)

© photograph by biztripcru@gmail.com



© photograph by biztripcru@gmail.com



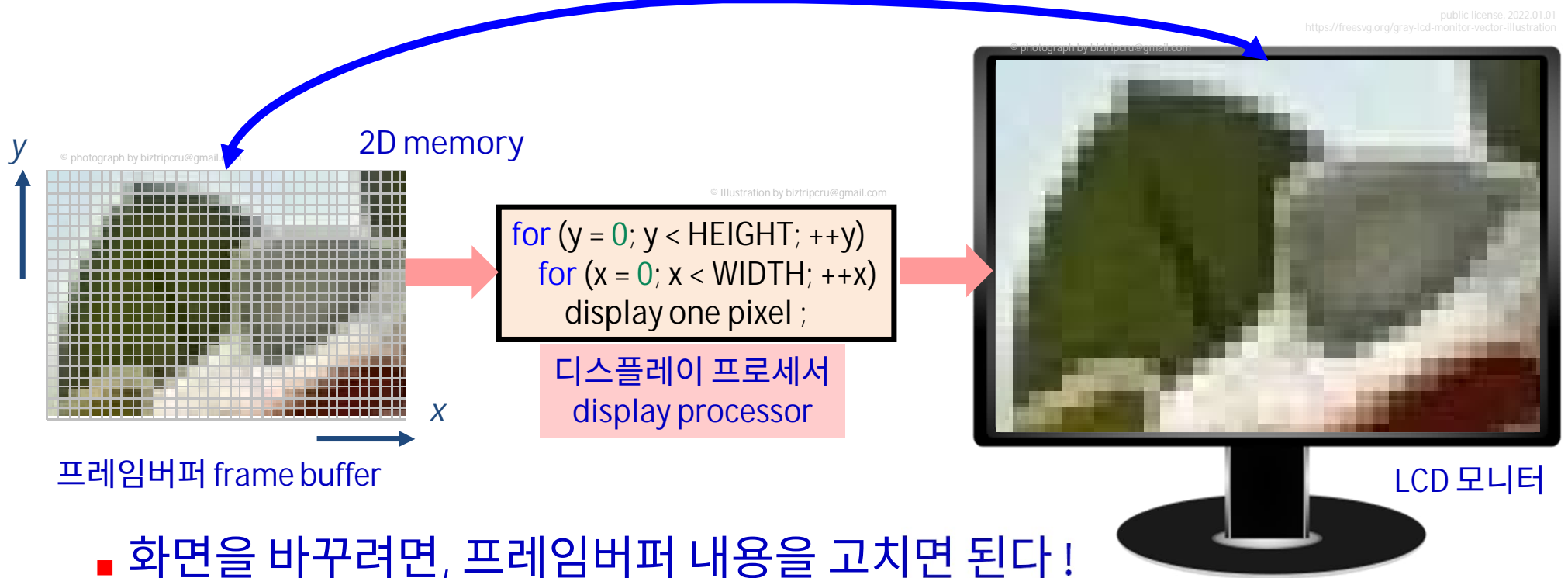
래스터 시스템 용어들

- pixel 픽셀 = picture element
- raster 래스터 = 픽셀의 2차원 배열 a 2D array of pixels
- framebuffer 프레임버퍼 = 픽셀을 저장한 메모리 영역 (2D) memory area for pixel storage



프레임버퍼의 기능

- 프레임버퍼 → 화면 으로 1:1 대응 관계



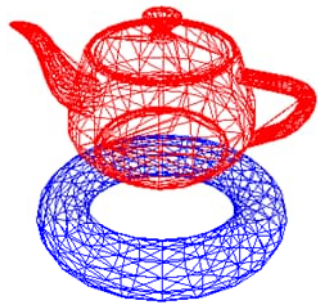
그래픽스 아키텍처

Graphics Architecture

모델링-렌더링 패러다임

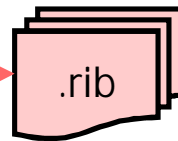
- 모델링 modeling_ 렌더링 rendering 패러다임 paradigm
 - 가장 간단한 그래픽스 아키텍처, 가장 오래된 역사
 - 서로 독립적인 프로그램 → **파일 포맷**으로 자료 교환
 - 모델링 프로그램 / 렌더링 프로그램이 서로 독립적
 - 적용 예 : Pixar 픽사 RenderMan 렌더맨 Interface (.rib)

CC BY-SA 4.0, 2022.01.01
https://commons.wikimedia.org/wiki/File:Toru_and_utah_teapot.png

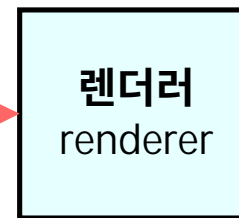


AutoCAD, 3DS, MAYA, ...

© Illustration by biztripcru@gmail.com



limited use, 2022.01.01
<https://namu.wiki/jump/sanyRplpvZ2z41K9YAaODtvgUjlcVrEk1fHG2WPPFrGtZwSP28fGvWHmXb9WGGa>
RENDERMAN.



RenderMan 프로그램

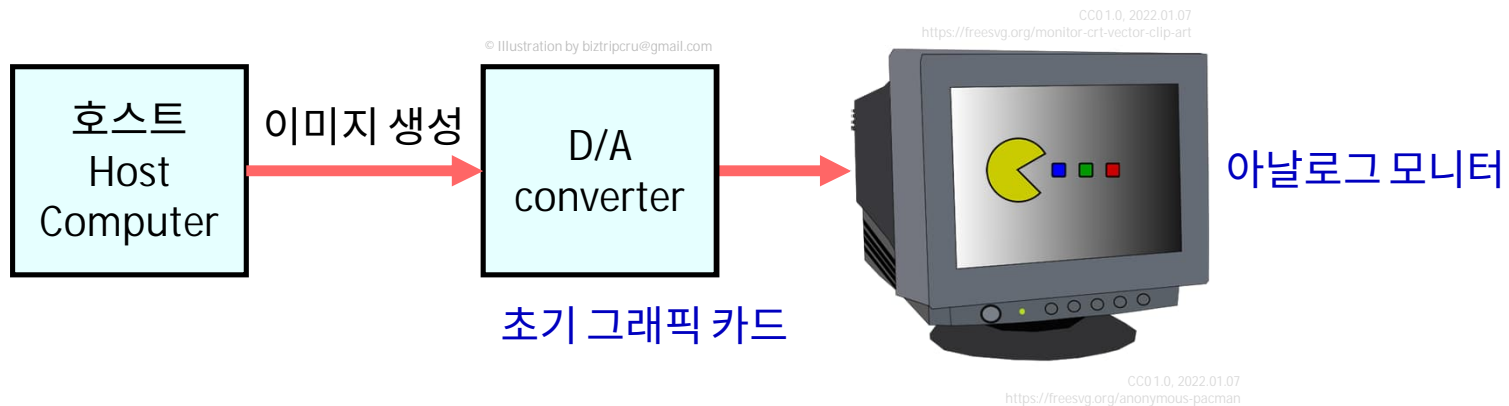
CC BY 2.0, 2022.01.01
<https://creativecommons.org/licenses/by/2.0/deed.en>



© Illustration by biztripcru@gmail.com

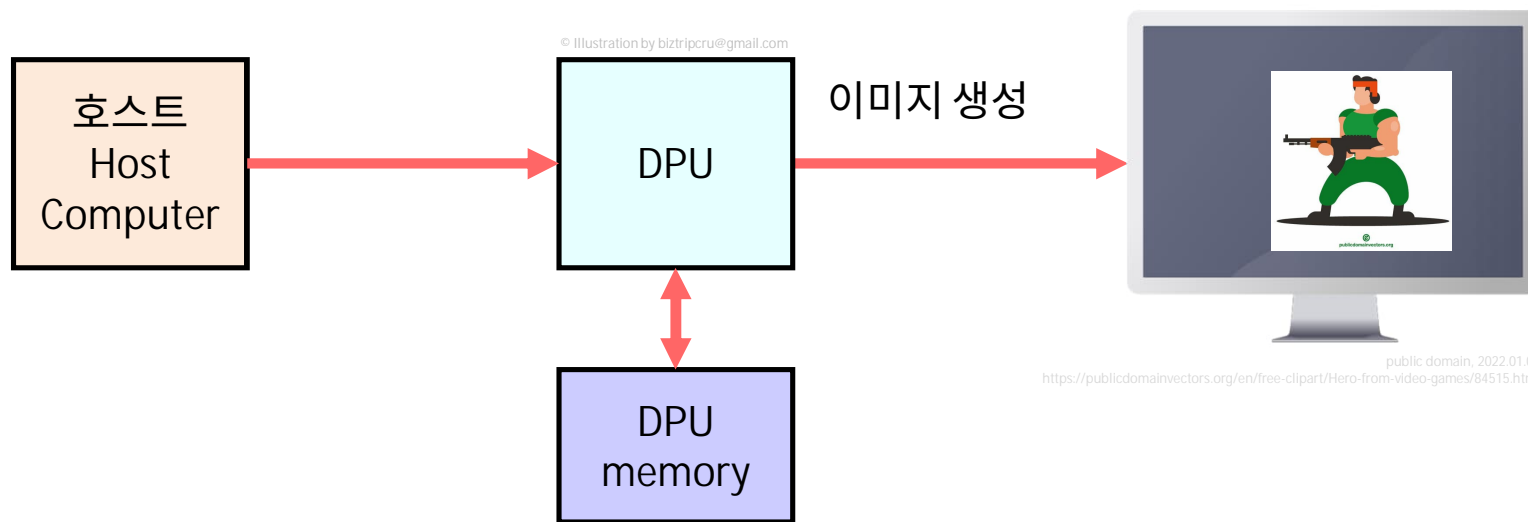
CPU 단독 처리 모델

- single CPU model
 - CPU 에서 모두 처리하자
 - ▶ Intel x86 : SSE (streaming SIMD extension)
 - 문제점: CPU가 담당하는 계산량이 너무 많다



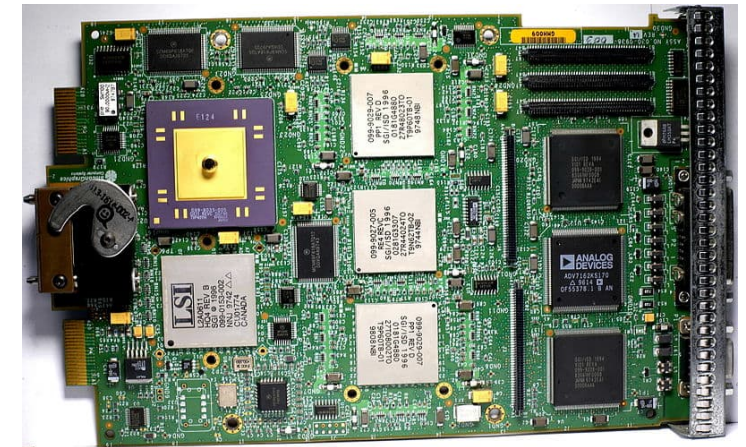
DPU 모델

- DPU = display processing unit, **display processor**
 - 그래픽스 처리만 전담하는, 특별한 목적의 프로세서
 - 파이프라인 모델로 가는 중간 단계

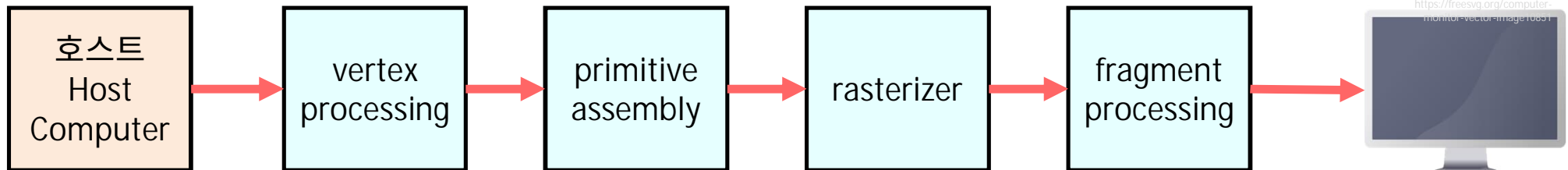


파이프라인 모델

- 전용 그래픽 카드 방식 모델
 - 그래픽스 처리 과정을 단계별로 VLSI 칩으로 처리
 - 실리콘 그래픽스 Silicon Graphics 가 도입
 - ATI, NVIDIA 에서 적극 활용



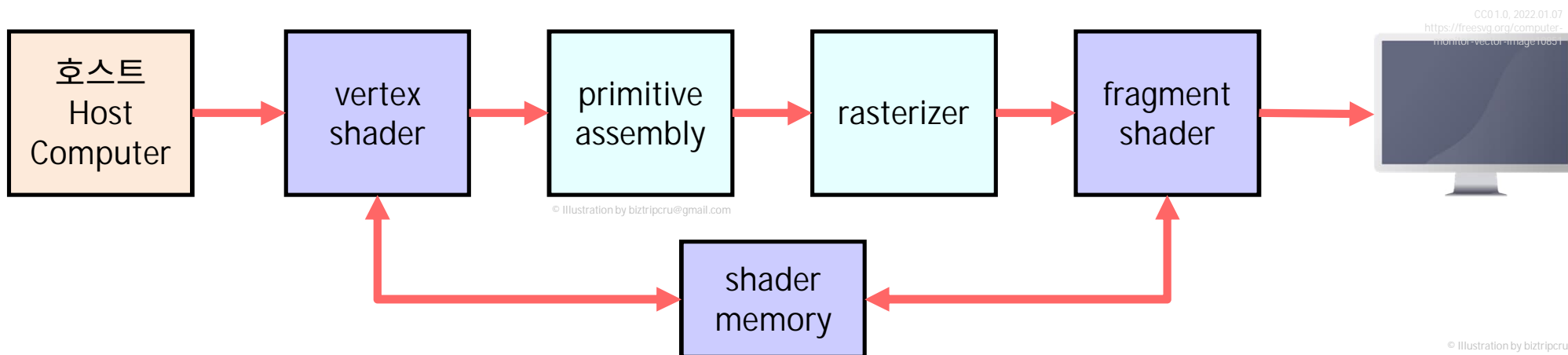
Silicon Graphics Octane 그래픽스 보드



© Illustration by biztripcru@gmail.com

프로그래머블 programmable 파이프라인 모델

- 고정된 VLSI 칩 대신, 프로그래밍 가능한 processor를 사용
 - 더 효율적이고, 다용도로 사용 가능
 - vertex shader 버텍스 셰이더 : programmable vertex processor
 - fragment shader 프래그먼트 셰이더 : programmable fragment processor
 - shading language 셰이더 랭귀지 : assembly language for shader coding



CC01.0, 2022.01.07
<https://freemsg.org/computer-monitor-vector-image10331>

© Illustration by biztripcru@gmail.com

GPU 지피유

- GPU = graphics processing unit
 - 프로그래머블 파이프라인에 사용되는 그래픽스 전용 프로세서
 - 빠른 처리가 필요 → floating point 연산, 4×4 행렬 연산에 최적화
- GPGPU : general purpose GPU
 - GPU를 일반적 목적 (general purpose) 으로 사용
 - 즉, GPU를 CPU 대신 계산에 사용
 - 대규모 병렬 컴퓨팅 MPC *massively parallel computing* 의 시작

Contents **내용**

- 컴퓨터 그래픽스의 정의
- 이미지 생성
- 합성 카메라 모델
- 래스터 시스템
- 그래픽스 아키텍처

그래픽스 시스템

Graphics System

본고딕 Noto Sans KR

Source Sans Pro

Source Serif Pro

정참판 양반댁 규수 큰 교자 타고 혼례 치른 날

The quick brown fox jumps over the lazy dog

Mathematical Notations $O(n \log n)$