computer vision

Web as Medium

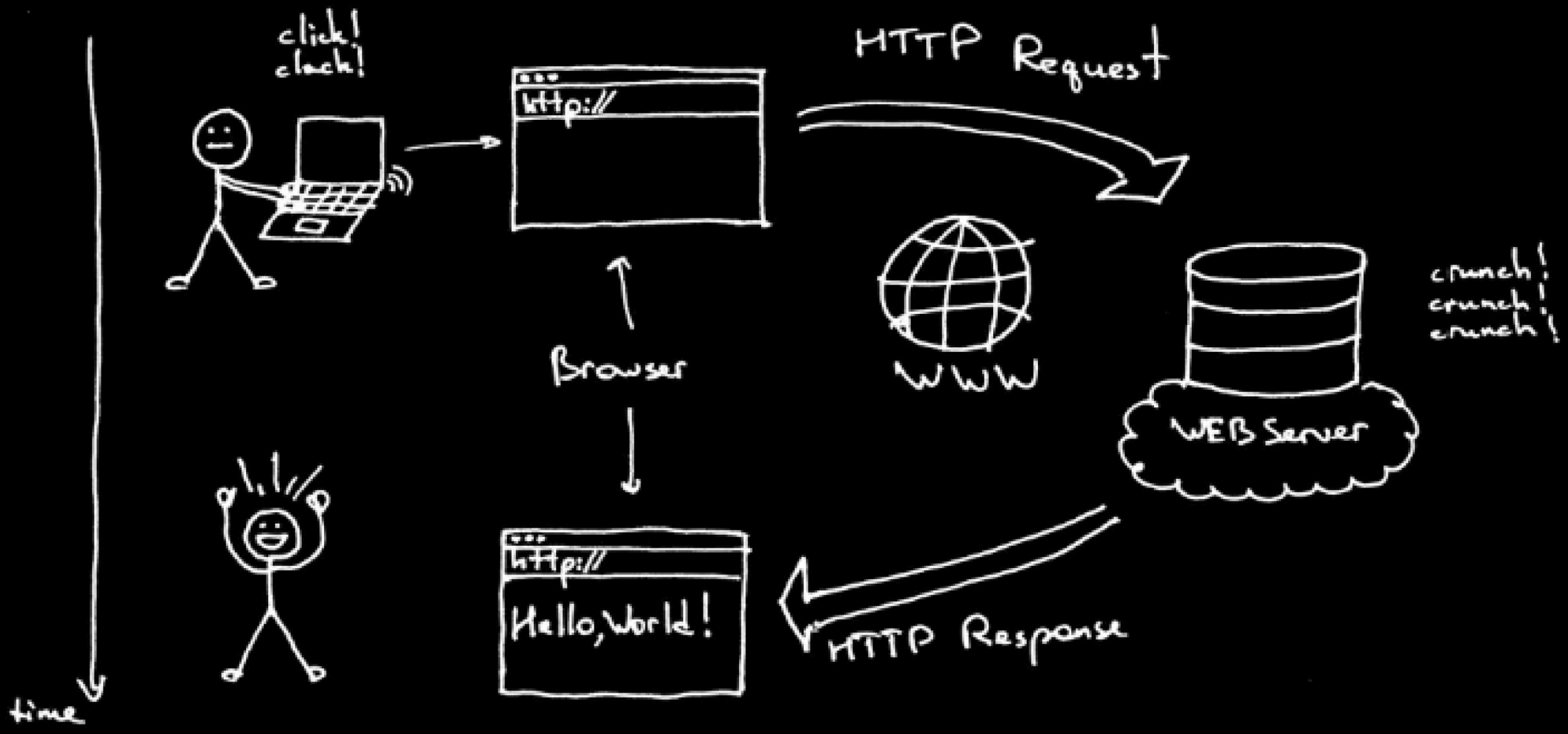
Hailey and Faith

04.10.2025

Browser Display + representation

Browser Display

Browsers are responsible for retrieving and displaying web content to users. When a user enters a URL or clicks on a link, the browser initiates a complex series of actions to retrieve the web content from a server and display it on the user’s device.



A web browser is a piece of software that loads files from a remote server (or perhaps a local disk) and displays them to you — allowing for user interaction.

Browser Rendering

The rendering process begins when you open a web page in your browser. This process involves several steps that work together to transform the page's code into a visual display on your screen. The steps involved are:

1. Parsing HTML
2. Parsing CSS
3. Constructing the Rendering Tree
4. Layout
5. Painting
6. Compositing

1. Parsing HTML - raw bytes of HTML to DOM

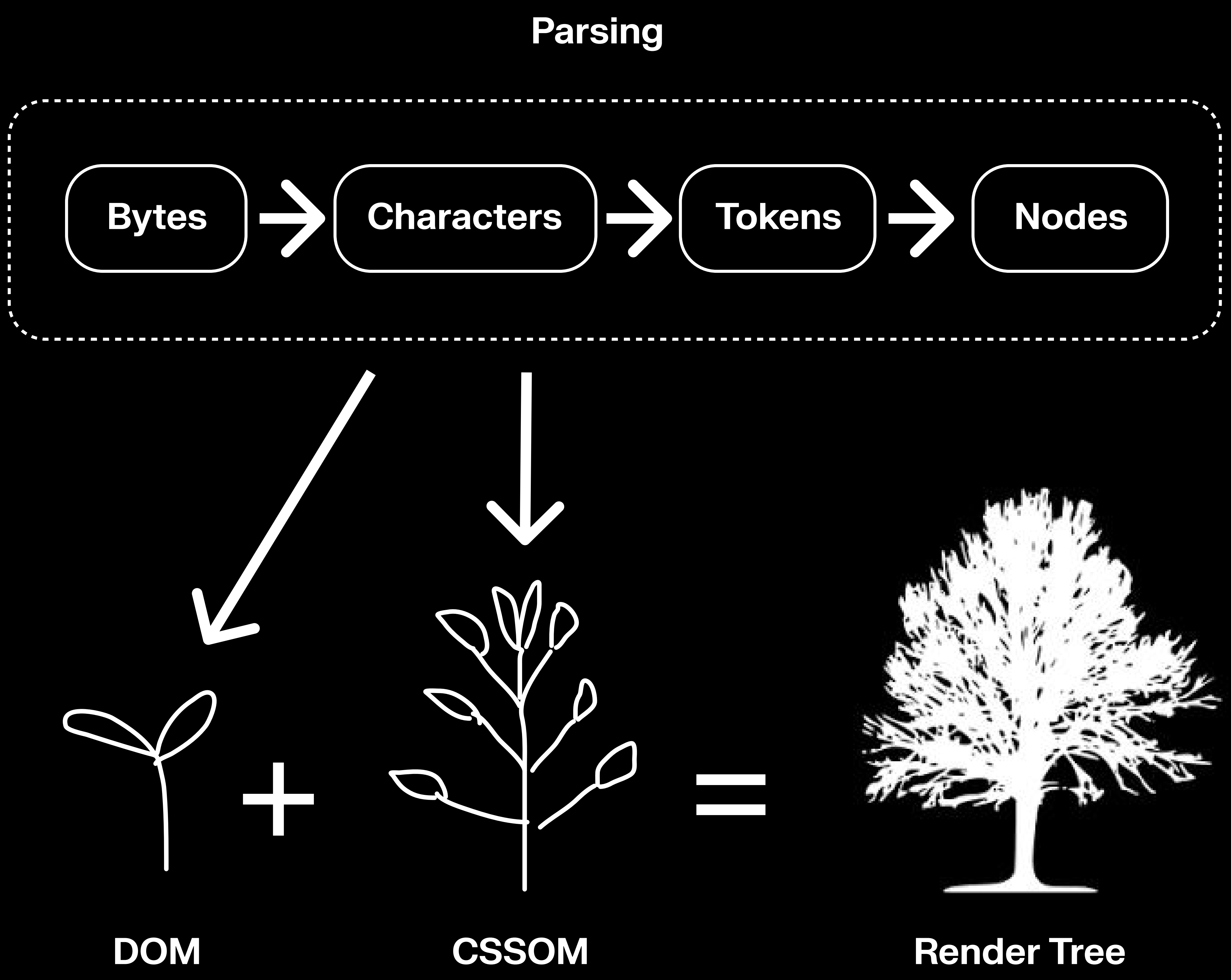
When the browser receives an HTML document, it begins by parsing the code. It parses the HTML code to create a tree-like structure called the Document Object Model (DOM). The DOM represents the web page's structure and includes all of the elements and attributes defined in the HTML code.

2. Parsing CSS

The browser takes the CSS code associated with the page and creates a separate tree-like structure known as the CSS Object Model (CSSOM). The CSSOM contains all of the style rules defined in the CSS code.

3. Constructing the Rendering Tree

After the browser has created the DOM and CSSOM, it combines the two to create the Rendering Tree. The Rendering Tree combines the DOM and CSSOM, representing the elements on the page and their associated styles.



4. Layout

The browser calculates the position and size of each element in the Rendering Tree. This is based on the styles defined in the CSS code and the content of the DOM. The Layout step is **important for determining the placement of each element on the screen and ensuring that the page displays correctly.**

5. Painting

Once the Layout has been calculated, the browser moves on to the Painting step. Painting involves filling in each element with color and creating an image of the page to be displayed on the screen. The browser uses the styles and layout information to paint each element in the Rendering Tree.

6. Compositing

The browser combines all of the painted elements into a final image. This image is then displayed on the screen, completing the rendering process.

1. Parsing HTML

When the browser receives an HTML document, it begins by parsing the code. It parses the HTML code to create a tree-like structure called the Document Object Model (DOM). The DOM represents the web page's structure and includes all of the elements and attributes defined in the HTML code.

2. Parsing CSS

The browser takes the CSS code associated with the page and creates a separate tree-like structure known as the CSS Object Model (CSSOM). The CSSOM contains all of the style rules defined in the CSS code.

3. Constructing the Rendering Tree

After the browser has created the DOM and CSSOM, it combines the two to create the Rendering Tree. The Rendering Tree combines the DOM and CSSOM, representing the elements on the page and their associated styles.

1. Issues with Parsing HTML - invalid HTML code

If the HTML code is not well-formed or contains errors, the browser may be unable to create a complete DOM tree. Results in missing or misplaced elements on the page or errors in the display of the page. JavaScript - can modify the content and structure of the page after it has been parsed.

2. Issues with Parsing CSS - invalid CSS code

If the CSS code contains errors, the browser may be unable to create a complete CSSOM tree. This can result in missing or incorrect styles on the page or errors in the display of the page.

3. Issues with the Rendering Tree - conflicting styles

The browser must determine which rule takes precedence if multiple style rules apply to the same element. This can lead to unexpected behavior or an incorrect layout on the page. Another issue is using complex or inefficient CSS selectors, which can slow the rendering process and affect page performance.

4. Layout
The browser calculates the position and size of each element in the Rendering Tree. This is based on the styles defined in the CSS code and the content of the DOM. The Layout step is **important for determining the placement of each element on the screen and ensuring that the page displays correctly.**

5. Painting
Once the Layout has been calculated, the browser moves on to the Painting step. Painting involves filling in each element with color and creating an image of the page to be displayed on the screen. The browser uses the styles and layout information to paint each element in the Rendering Tree.

6. Compositing
The browser combines all of the painted elements into a final image. This image is then displayed on the screen, completing the rendering process.

5. Issue with painting and compositing
complex or inefficient CSS styles can slow the rendering process and affect page performance. Another issue is using large or uncompressed images, which can take a long time to load and display on the page.

so.....
computer vision?

What is computer vision?

Computer vision refers to the ability of machines to identify patterns within visual data and glean meaningful insights. Depending on the context, machines can use cameras, sensors, smartphones and other devices to compile data for training and analysis. They can then perform tasks like reading written text, recognizing specific faces in images and locating particular objects in a video feed.

Teaching computers how to make sense of visual information.

Basic browser rendering = layout and display logic. No computer vision.
Computer vision = perception and interpretation of visual content.

Libraries/frameworks: OpenCV, TensorFlow, MediaPipe, PyTorch, Detectron2
Hardware: Cameras, LiDAR, infrared sensors
Algorithms: Convolutional Neural Networks (CNNs), edge detection, feature matching

Task	Example
Object detection	Finding and labeling objects in an image (e.g. "cat", "tree")
Facial Recognition	Identifying or verifying who someone is
Pose estimation	Understanding how someone is standing, sitting, moving
Image segmentation	Separating parts of an image (like sky vs. ground)
Optical character recognition (OCR)	Reading text from images (like scanned documents)
Scene understanding	Recognizing what's happening in a space
Depth estimation	Figuring out how far away things are (used in AR/VR)

Detect objects
Recognize patterns
Measure distances
Analyze movement
Classify scenes
Segment images (semantic and instance segmentation)
Track objects across frames
Detect and track faces
Estimate human poses
Recognize gestures
Detect emotions
Analyze gait
Read text in images (OCR)
Detect logos or brands
Infer depth from images
Reconstruct 3D environments
Perform SLAM (Simultaneous Localization and Mapping)
Detect motion or changes over time
Estimate optical flow
Apply style transfer
Inpaint missing parts of images
Enhance resolution (super-resolution)
Generate new images (e.g. with GANs)
Match features across images
Align or register images
Perform visual search
Answer visual questions (VQA)
Detect anomalies
Recognize actions or behaviors
Analyze environmental conditions (e.g. fog, light)
Estimate crowd density
Measure biometrics (e.g. heart rate from video)
Detect and count objects
Track environmental shifts (e.g. plant growth)
Detect shadows and lighting changes
Identify materials or textures
Generate 3D point clouds from images
Estimate age or gender
Segment and analyze documents
Visualize flow or direction (e.g. in fluids, traffic)
Understand scene geometry
Build heatmaps of attention or focus
Predict future frames in a video
Simulate vision in different spectra (e.g. IR, UV)

Challenges the screen as just a flat surface — making it perceptual, dynamic, alive

technical concepts

Photogrammetry
Motion Tracking
Place Illusion
Gamification
three.js

Photogrammetry

Photogrammetry is the process of taking reliable measurements from photographs, and using photographs to create 3D models of objects, structures, or spaces. It’s used in many fields, including construction, engineering, archaeology, manufacturing, public safety, and gaming.

How it works

1. Image Acquisition

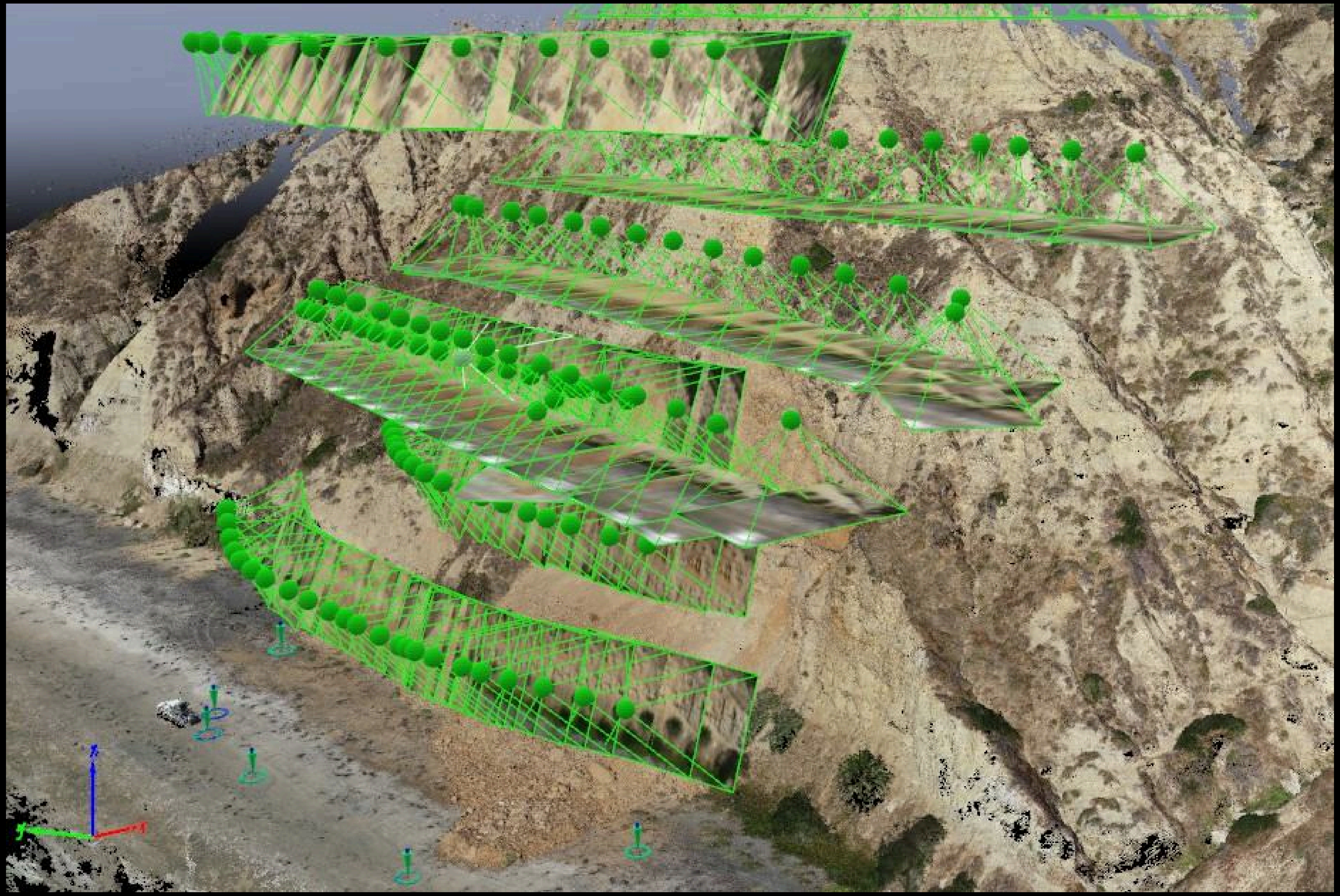
Multiple photographs are taken of the same object or area from different angles and positions.

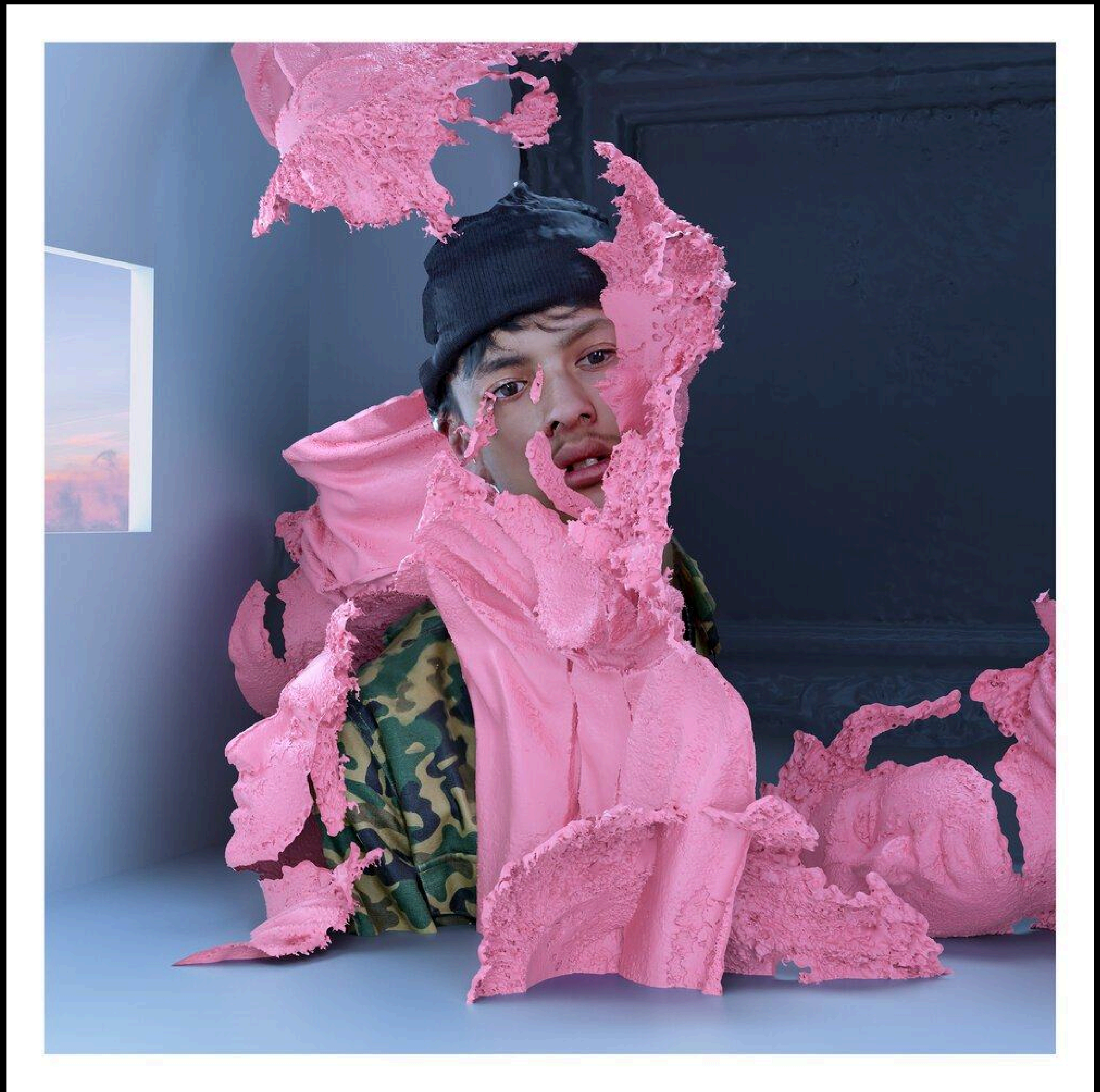
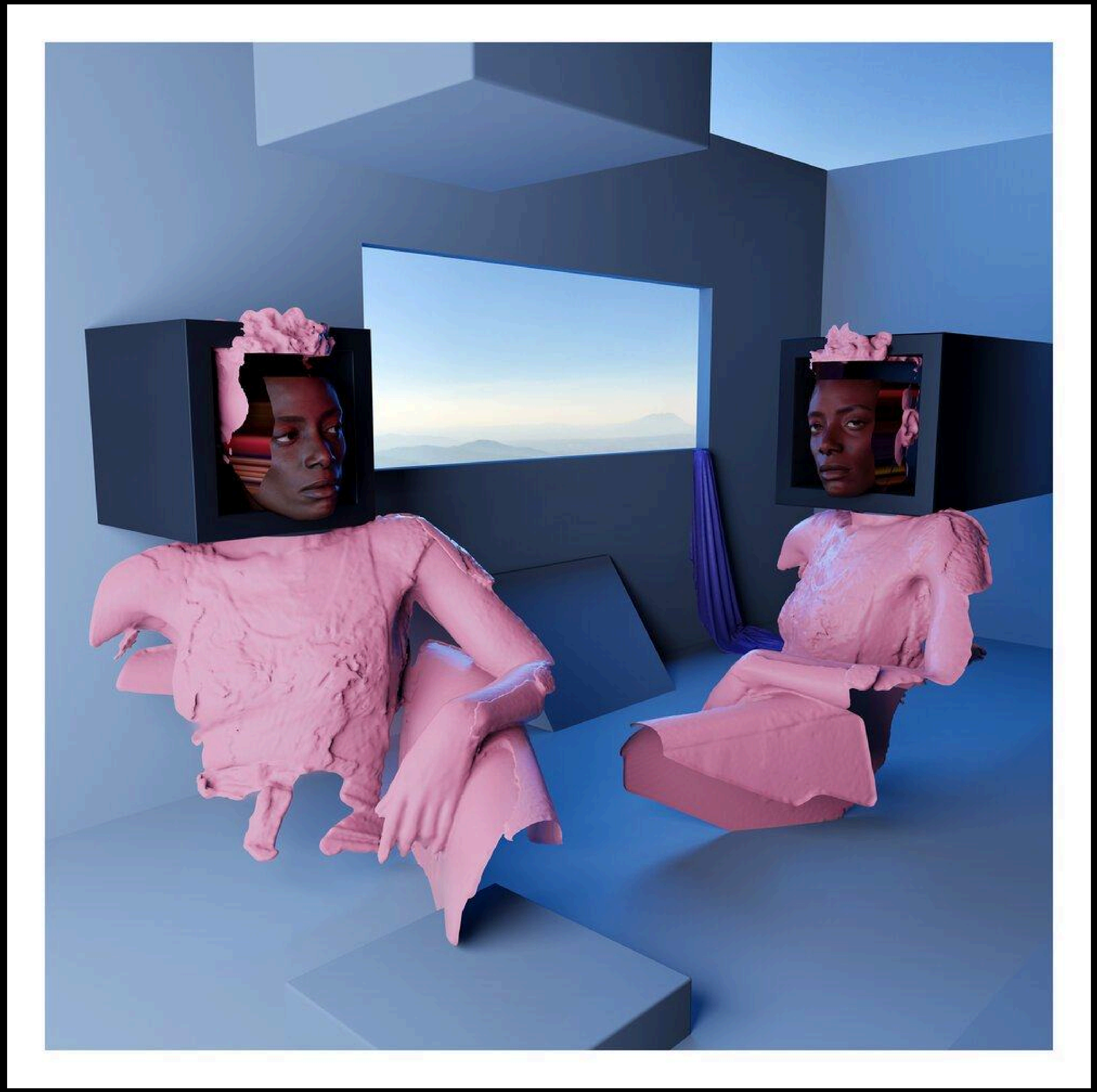
2. Image Processing

Software analyzes these images, identifying corresponding points (landmarks) and their spatial relationships.

3. 3D Reconstruction

Based on the identified landmarks and their spatial relationships, the software creates a 3D model, often in the form of a point cloud or mesh.



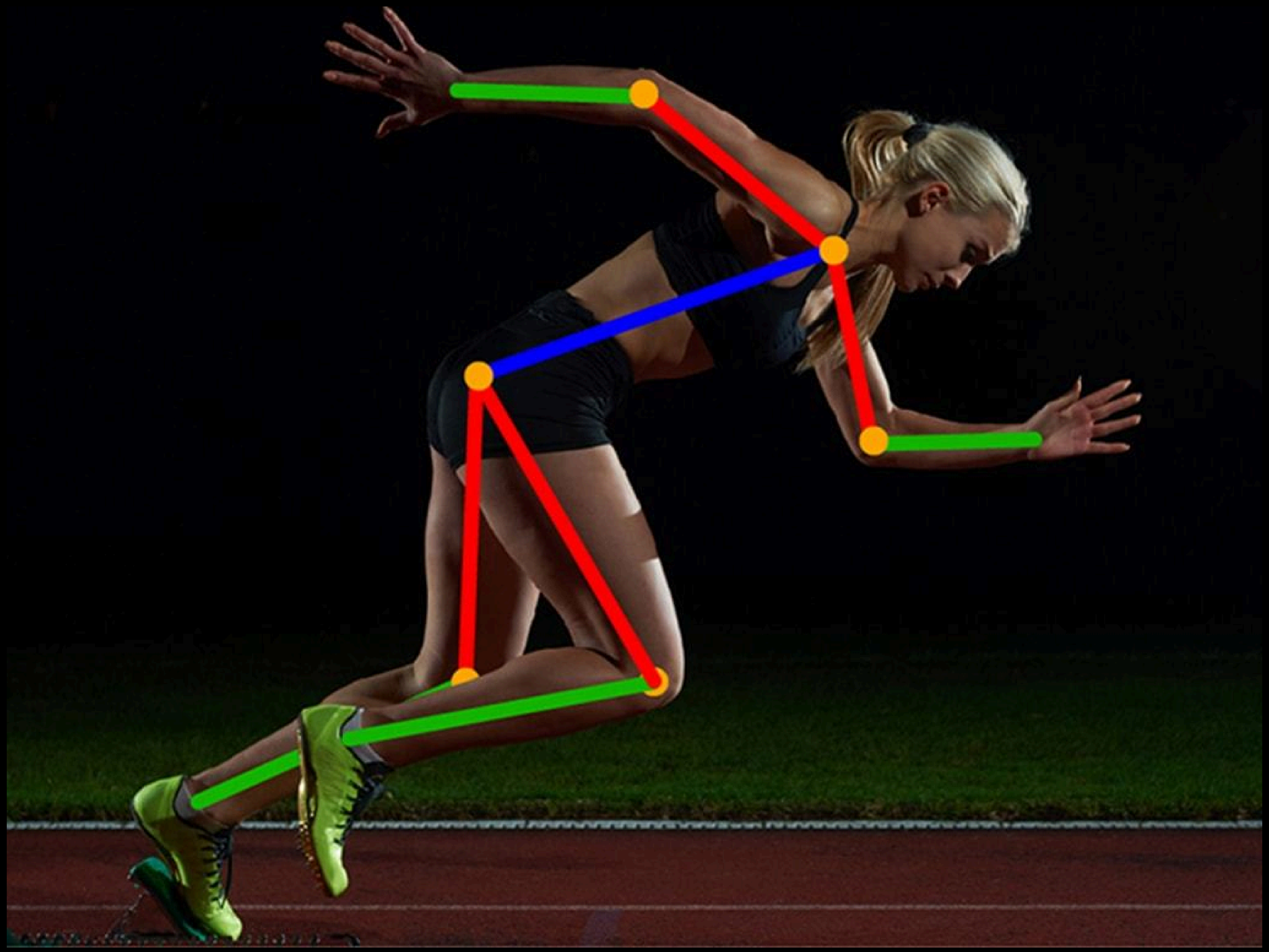
Human Unlimited, *Dimitri Daniloff, 2019*

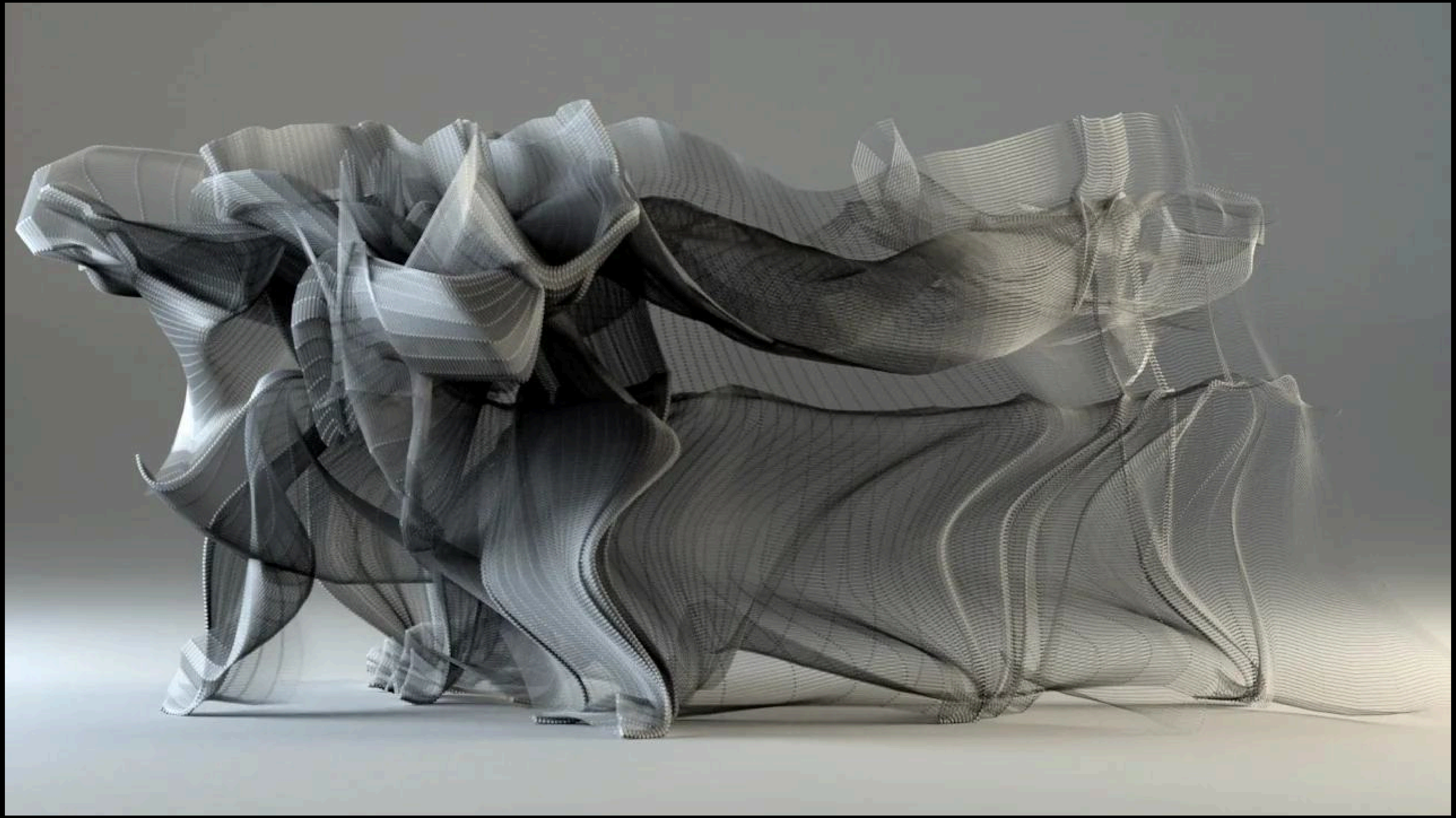
motion tracking

Motion tracking is the process of recording the movement of objects or people. It is a cinematic technique that allows the insertion of computer graphics into live-action footage with correct position, scale, orientation, and motion relative to the objects in the shot.

How it works
Softwares analyze the positional changes of pixels across frames to identify and track the movement of people, objects, backgrounds, and even cameras. It identifies key points or features that move, and then tracks their movement over time.

It is used for
Visual Effects
Animation
Human-computer interaction
Security and Surveillance
Video communication
Augmented Reality





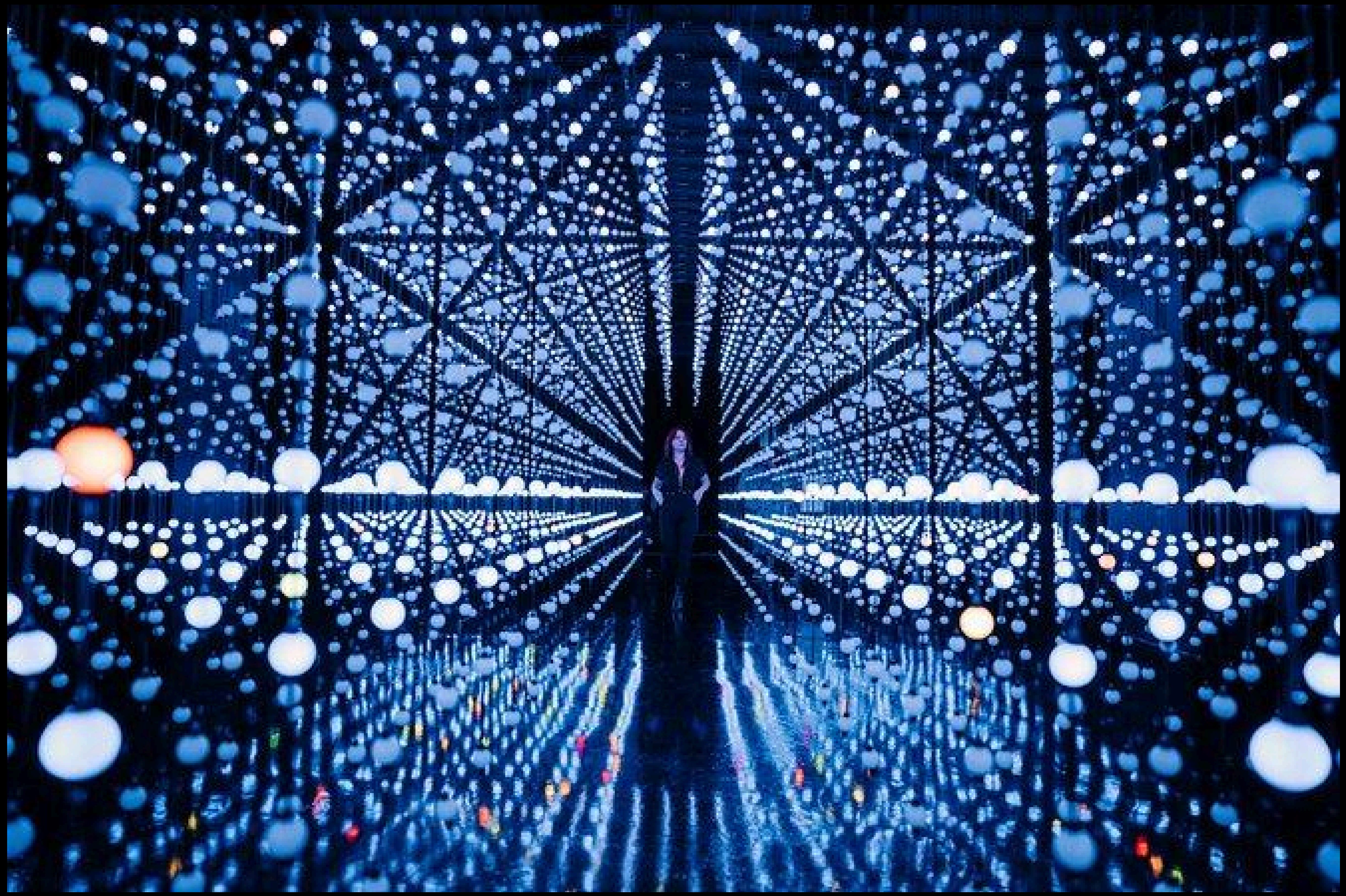
Kung Fu Motion Visualization, *Tobias Gremmler, 2003*

https://vimeo.com/163153865

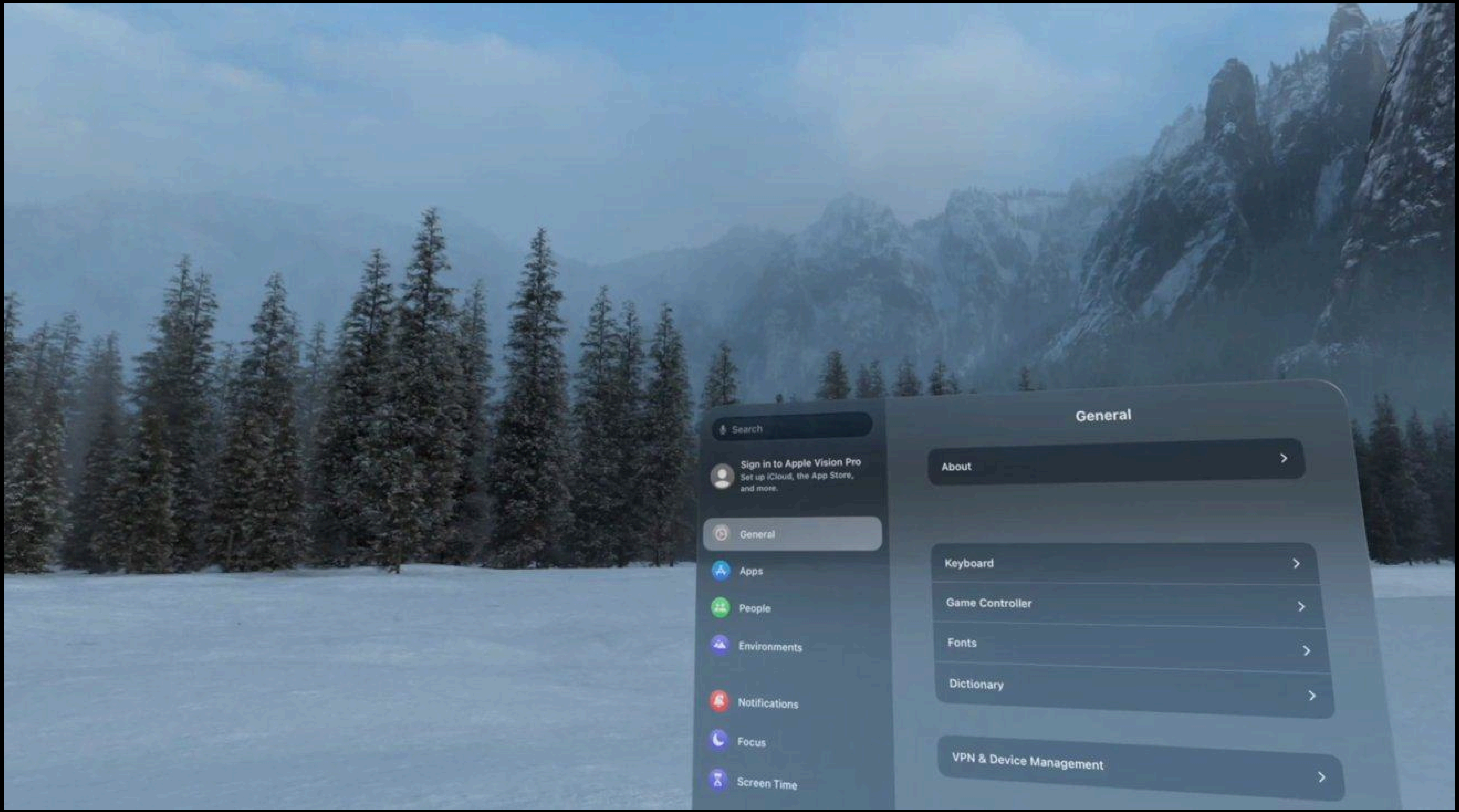
Place Illusion

Place Illusion(PI) is the sensation of being in a real place, a key component of presence in virtual environments. It focuses on the user's subjective experience of being in a virtual space, rather than simply perceiving it as a simulation. In arts, it is used for creating virtual environments. By manipulating visual elements, sounds, and interactions, coders can create experiences that make users feel present in the virtual world.

How it works
The headset tracks a user's head position and orientation, and hand/body positions are tracked in 3D space using optical, inertial, or inside-out tracking systems. Game engine, spatial audio engine, and physics engine update the camera view matrix, sound sources and other interactions in real-time.



It is used for
Immersive Environments
Interactive Art
Generative Art



Apple Vision Pro : Environment, *2024*

https://www.youtube.com/watch?v=rvRFKWQEitQ

Gamification

Gamification is the application of typical elements of game playing — point scoring, competition with others, badges, leaderboards — to other areas of activity.

It is used for
many occasions.
e.g. encouraging engagement with a product or service in markets, enhancing audiences' experiments to be more enjoyable in exhibitions.

three.js

Three.js is a cross-browser JavaScript library and application programming interface (API) used to create and display animated 3D computer graphics in a web browser using WebGL. It is the creation of GPU-accelerated 3D animations using the JavaScript language as part of a website.

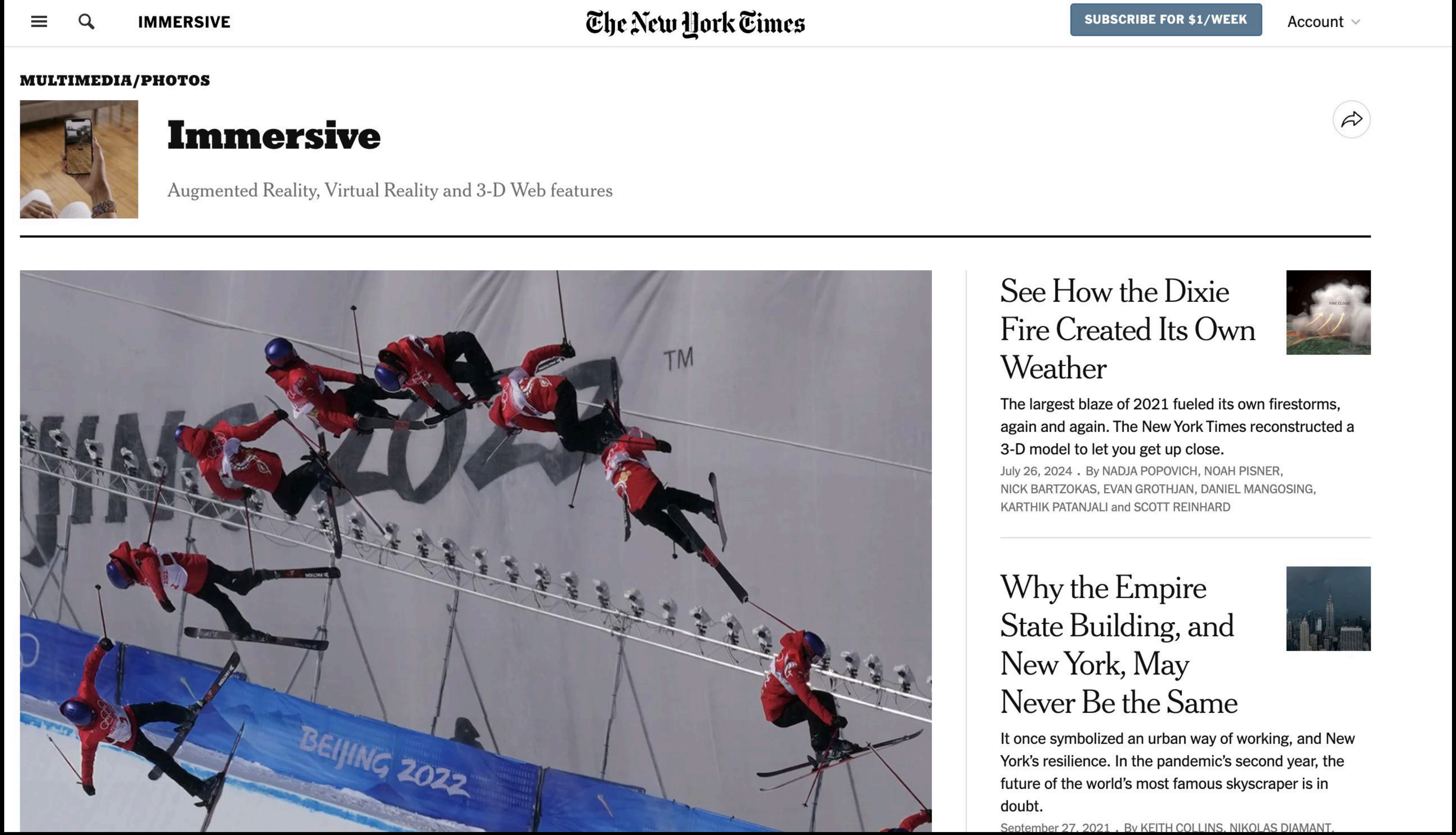
How it works

Set up a 3D scene—like a virtual stage. Add a camera that determines what part of the scene is visible, and lights to make the objects appear realistically. You create 3D objects and material, and position everything in 3D space using x, y, and z coordinates. Once the scene is set, a renderer draws everything from the camera's perspective onto the screen. Animate objects by updating their properties over time, and add user interactions like mouse movements or clicks.

Bruno Quintela's website, *Bruno Quintela*
(currently closed)

More examples on
https://threejs.org/examples/#webgl_animation_keyframes

case studies



New york times- Immersive

Experimental Capture

Computational & Expanded [redacted]ography

Golan Levin & Nica Ross, Carnegie Mellon University
Curricular Materials for CMU Course 60-461/761, 54-461/661

> This is an interdisciplinary studio course in expanded media practices that arise from using devices and algorithms to "capture" the world. We will explore experimental workflows, ranging from no-tech and low-tech to emerging and state-of-the-art techniques, in order to capture, model, and share new representations of people, objects, places and events. Through self-directed research projects, students will develop systems to capture a wide variety of phenomena, and creatively share the media they collect. We will cover a wide range of techniques and artistic practices that incorporate immersive, panoramic, high-speed, multiscopic, and multispectral imaging; depth sensors and 3D scanners; motion capture systems for gestures of the face, body, hand, and eye; computer vision and machine learning techniques for detection, tracking, recognition and classification; and other unusual, forgotten, and nascent technologies for transducing the unseen, ephemeral, and otherwise undetectable.

Experimental Capture