

DEVELOPMENT OF INTERACTIVE AR OBJECTS: FROM CODE TO ARTISTIC IMAGE

The development of interactive AR objects represents a synthesis of low-level technical procedures and high-level artistic representation [1]. This process requires a strict sequential chain: from mathematical modeling of space and geometry (using libraries such as OpenCV for computer vision and frameworks like ARCore/ARKit for SLAM tracking) through software implementation (C# in Unity or C++ in Unreal Engine with OpenGL ES/Vulkan) to final artistic polishing. A critically important technical parameter at this stage is optimizing polygon count (reducing to 50-100 thousand for complex objects on mobile platforms) and texture size to achieve a frame rate of 60 on target devices [2].

The technological foundation of an AR object is based on precise spatial anchoring, ensured by tracking systems with different architectural approaches. Marker-based tracking provides high accuracy (error less than 0.5 cm) in controlled lighting but is limited by static conditions. Markerless tracking, which uses features of the natural environment, requires computationally powerful structure-from-motion algorithms for point cloud construction. The most modern systems, like Apple's LiDAR Scanner paired with ARKit 3.5+, provide deep environmental mapping and generate mesh in real-time, which is crucial for implementing physical interaction of the AR object with the environment (occlusion, collision detection).

Interaction of the AR object with the user is implemented through abstraction layers: from low-level sensory signals to high-level logic. The system input uses gesture recognition algorithms (for example, via MediaPipe for skeletal hand tracking with latency ~10-20 ms) and gaze control technologies. These signals are transformed into events within the game engine, which trigger scripted logic.

The visual quality of an AR object as an artistic image is determined by the rendering pipeline, technical material parameters, and adaptation to variable lighting. Modern engines use URP (Universal Render Pipeline) or HDRP (High Definition Render Pipeline) with support for shaders written in HLSL. Key to achieving photorealism is precise configuration of PBR materials, where parameters such as roughness (values 0-1), metallic (0-1), and normals (normal map, using tangent space) are calculated based on reference images. The Ambient Lighting Estimation system via the device camera's API allows for dynamic adjustment of shader parameters (IBL - Image-Based Lighting) for accurate integration of the virtual object

into the real environment, including matching light color temperature and intensity.

Animation of an AR object as an artistic-dynamic image requires applying skeletal animation (skinning), morph targets (blend shapes), and procedural animation synchronized with external data. Skeletal animation uses bone transformation matrices for each frame, requiring optimization of bone count. For implementing smooth transitions between animation clips, animation blending with motion curves is used [3].

Performance optimization of an AR object is a mandatory stage, requiring profiling (using tools like Unity Profiler or Xcode Instruments) and applying a range of techniques. These include: level of detail, geometry batching to reduce draw calls, texture atlasing, occlusion culling, as well as using texture compression systems. Critical metrics for stable operation are: peak RAM usage, stable frame rate, CPU/GPU temperature (to avoid thermal throttling).

Network interaction between multiple AR objects and users in a collaborative session is based on a client-server or peer-to-peer (P2P) architecture with state synchronization via real-time protocols. For this purpose, libraries such as Photon Engine, Normcore, or Fish-Networking are used, which operate over the transport layer. The technical foundation is the synchronization of transforms (position, rotation, scale) of each object between clients at a frequency of 10-20 times per second (network tick rate) [4].

Testing and validation of AR object quality are conducted at various levels: from unit testing of source code to user experience (UX) research under varying lighting and spatial geometry conditions. For automation, testing frameworks are used. Critical technical parameters for validation include: positional accuracy, tracking stability, and interaction response time (must be <100 ms).

References

1. Чеботарьова, І.Б., Гуріна, К.В., & Чеботарьов, Р.І. (2022). AR-технології в сучасній рекламі. Поліграфічні, мультимедійні та web-технології. Т. 2. (с. 34-36).
2. Geroimenko, V. (Ed.). (2020). *Augmented reality in education: A new technology for teaching and learning*. Cham, Switzerland: Springer.
3. Kalyanaraman, S. (2022). Augmented reality for enhanced user experience in theatre. *Journal of Arts & Humanities*, 11(2), 33-47.
4. Leshchenko, O.V. (2021). *Tsyfrovі tekhnolohii v suchasnomu teatral'nomu mystetstvi*. Digital technologies in modern theatrical art: Monograph. Kyiv, Ukraine: NaUKMA.
Academic supervisor: Cand. Sc. (Technic), docent Suprun O.O.