SIR for AI Video Production:
From Agent Pipeline to Editable Project

Kelly Peilin Chan
kelly@buda.im

Abstract

Text-to-video (T2V) systems produce opaque video files—no internal structure, no character consistency across shots, no way to correct a single subtitle without full regeneration. We introduce **SIR-T2V**, a system built on the principle of **Structured Intermediate Representation (SIR)**: *AI should output editable project files, not final rendered artifacts*. SIR-T2V implements this principle for commercial video production through an agent-orchestrated pipeline that mirrors professional filmmaking: a Director Agent coordinates specialized sub-agents through ideation → character casting → storyboarding → visual sourcing → script finalization → project assembly → digital human generation → final project. The system introduces **multi-character tri-view casting**, where each character is represented by front/side/back orthographic views (`characters/A/{front,side,back}.png`) that anchor identity across all generated visuals (0.923 CSIM). The final output is not an `.mp4` but a parametric Remotion project—a **Structured Intermediate Representation**—where every shot, subtitle, transition, and overlay is individually editable. A two-pass assembly (draft for structural review, then digital human integration) ensures expensive generation only occurs after human approval. SIR-T2V reduces human correction cost by 4.7× compared to regeneration-based workflows and completes a 30-second advertisement in ∼6 minutes.

1 Introduction

A 30-second TikTok advertisement looks simple. Behind it lies a production pipeline that professional studios execute daily: ideation, market research, casting, storyboarding, asset sourcing, scriptwriting, shooting, and post-production. Each phase produces inspectable artifacts—mood boards, casting sheets, storyboards, shot lists, edit decision lists—that enable review, iteration, and parallel work.

AI video generation has largely ignored this structure. T2V systems [1, 2, 3] collapse the entire pipeline into a single prompt → `.mp4` step. The result: no character consistency across shots, no narrative structure, no way to fix a single subtitle without regenerating everything.

The SIR principle. We argue that the core problem is not model quality but *output format*. When an AI system outputs a final rendered artifact (an `.mp4`, a `.pdf`, a `.wav`), it destroys the compositional structure that humans need for review and correction. The solution is **Structured Intermediate Representation (SIR)**: AI should output *editable project files*—not final renders.

Preprint. February 2026. Work in progress.

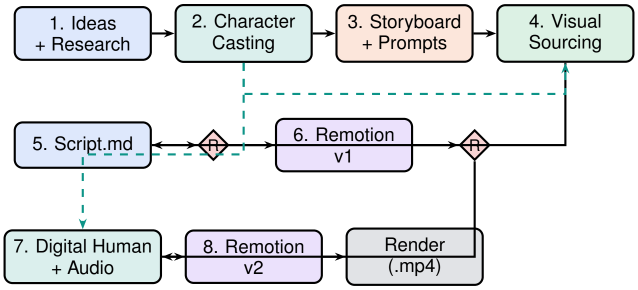

Figure 1: SIR-T2V pipeline. Diamonds marked “R” are human review gates. Dashed teal arrows show character identity flowing to visual sourcing and digital human generation. The final output is a Remotion project (SIR), not a rendered video.

SIR-T2V. We present **SIR-T2V**, a system that applies the SIR principle to commercial video production. An agent-orchestrated pipeline mirrors professional filmmaking, and the final output is a Remotion project—a parametric, human-editable video engineering project—not a rendered `.mp4`.

Contributions.

1. **SIR as a design pattern** (§9): We formalize the principle that AI systems should output editable project files rather than final artifacts, and show it applies across domains (video, slides, music, 3D, documents).
2. **Agent-orchestrated film pipeline** (§3): A Director Agent drives 8 stages mirroring professional film production, with structured artifact handoff and human review gates.
3. **Multi-character tri-view casting** (§4): Each character gets front/side/back orthographic views (`characters/*/{front,side,back}.png`) that anchor identity across all generated visuals, achieving 0.923 CSIM.
4. **Two-pass Remotion assembly** (§7, §8): Draft project from sourced visuals → human review → digital human and audio integration → final editable project.

2 Related Work

Text-to-Video Generation. Diffusion-based video models have progressed from early latent diffusion approaches [1] to high-fidelity systems such as Stable Video Diffusion [2], Emu Video [4], and MovieGen [3]. These systems generate short, self-contained clips but produce monolithic video files with no internal structure—the antithesis of SIR.

Talking Head and Digital Human. Audio-driven talking head synthesis has advanced through diffusion-based methods [5, 6]. EditYourself [7] enables word-level editing of talking head videos. These focus on single-shot portrait animation without addressing multi-shot narrative or production pipeline structure.

Multi-View Identity Consistency. Generating consistent multi-view character images is critical for identity preservation. Zero123++ [15] synthesizes novel views from a single image. CharacterGen [9] generates multi-view consistent character images. Joint2Human [10] predicts front/back normal maps. EG3D [11] introduced tri-plane representations for 3D-aware GANs. SIR-T2V uses tri-view representations not for 3D reconstruction but as *identity anchors* for conditioning downstream 2D generation agents.

Identity-Preserving Generation. IP-Adapter [12] and PhotoMaker [13] enable identity-conditioned image generation via single reference images. SIR-T2V extends this to multi-character, multi-view conditioning within a structured production pipeline.

Table 1: SIR-T2V agent pipeline.

#	Agent	Input	Tool	Output
1	Researcher	User idea	LLM + web search	`ideas.md`
2	Casting Dir.	ideas.md	T2I + Zero123++	`characters/*/`
3	Storyboarder	ideas + characters	LLM structured	`storyboard.md` + `scene_prompts/`
4	Art Director	prompts + characters	T2I / stock search	`assets/`
5	Screenwriter	storyboard + assets	LLM structured	`script.md`
		— *Human Review Gate 1: approve script* —		
6	Editor (v1)	script + assets	Remotion codegen	`remotion-project/v1`
		— *Human Review Gate 2: approve draft* —		
7	Performer	script + characters	Digital human + TTS	`clips/` + `audio/`
8	Editor (v2)	v1 + clips + audio	Remotion update	`remotion-project/v2`

Structured Video Editing. Remotion [22] enables programmatic video via React components. TransPixeler [19] generates RGBA video with alpha channels. Layered neural atlases [20] provide structured video decompositions. SIR-T2V builds on these ideas by making the *entire generative pipeline* output a structured, editable project.

AI Agent Systems. ReAct [25] established the pattern of tool-augmented LLM agents. Multi-agent systems have been applied to software engineering and research. SIR-T2V applies multi-agent orchestration to video production, with each agent producing structured artifacts consumed by downstream agents.

3 System Overview

SIR-T2V is driven by a **Director Agent** that orchestrates 8 production stages, each handled by a specialized sub-agent producing structured artifacts.

Design principles.

1. **SIR output**: The final artifact is an editable Remotion project, not a rendered `.mp4`.
2. **Characters-first**: Characters are cast before storyboarding. Their tri-view identity sheets flow to all downstream visual generation.
3. **Two-pass assembly**: Remotion v1 (sourced visuals, placeholders for digital humans) → human review → Remotion v2 (digital humans + audio integrated).
4. **Review before generation**: Expensive generation (digital human, TTS) happens only after human approves the script and draft project.
5. **Markdown + JSON**: Human-readable Markdown for narrative (`ideas.md`, `storyboard.md`, `script.md`), structured JSON for machine consumption (`scene_prompts/`, `project.json`).

Artifact directory.

```
project/
  ideas.md                      # Step 1
  characters/                   # Step 2
    protagonist/
      front.png, side.png, back.png, meta.json
    narrator/
      front.png, side.png, back.png, meta.json
  storyboard.md                 # Step 3
```

Algorithm 1: SIR-T2V Director Agent

```
Input: User idea $\mathcal{I}$, optional character references
Output: Remotion project directory (SIR)
ideas.md ← ResearchAgent($\mathcal{I}$);
characters/ ← CastingAgent(ideas.md, refs);
storyboard.md, scene_prompts/ ← StoryboardAgent(ideas.md, characters/);
assets/ ← ArtDirectorAgent(scene_prompts/, characters/);
script.md ← ScreenwriterAgent(storyboard.md, assets/);
HumanReview(script.md);
$\mathcal{P}_1$ ← EditorAgent(script.md, assets/);
HumanReview($\mathcal{P}_1$);
clips/, audio/ ← PerformerAgent(script.md, characters/);
$\mathcal{P}_2$ ← EditorAgent($\mathcal{P}_1$, clips/, audio/);
return $\mathcal{P}_2$;
```

```
  scene_prompts/              # Step 3
    scene-01.json, scene-02.json, ...
  assets/                     # Step 4
    scene-01-keyframe.png
    scene-02-broll.mp4
    product-closeup.png
  script.md                   # Step 5
  clips/                      # Step 7
    scene-01-talking.mp4, scene-03-talking.mp4
  audio/                      # Step 7
    scene-01.wav, scene-03.wav, bgm.mp3
  remotion-project/           # Step 6 + 8 (SIR)
    src/Root.tsx, src/Ad.tsx
    public/                   # symlinks to assets/, clips/, audio/
    project.json
    package.json
```

Director Agent loop.

4 Step 2: Character Casting

A film has *characters*. Before any visual work begins, the Casting Agent establishes the identity of every character that will appear in the advertisement.

4.1 Multi-Character Model

Unlike single-reference approaches [12, 13] that condition on one image, SIR-T2V supports multiple characters per film. Each character c_k is represented by a tri-view identity sheet:

$$c_k = \left(V_k^{\text{front}},\ V_k^{\text{side}},\ V_k^{\text{back}},\ m_k\right) \tag{1}$$

where V_k^* are 512×768 orthographic views in A-pose and m_k is a metadata record (name, role, description, voice style).

Directory structure.

```
characters/
  protagonist/
    front.png
    side.png
    back.png
    meta.json     # {"name":"Lisa","role":"spokesperson",
                  #  "description":"Female 25-30, warm smile",
                  #  "voiceStyle":"warm, conversational"}
  narrator/
    front.png, side.png, back.png, meta.json
  expert/
    front.png, side.png, back.png, meta.json
```

Table 2: Cross-shot identity consistency (CSIM) vs. number of reference views.

Views	1 (front)	2 (F+B)	3 (F+S+B)	4	6
CSIM ↑	0.671	0.843	0.923	0.931	0.938

4.2 Casting Agent Workflow

The Casting Agent reads `ideas.md` to determine how many characters are needed and their descriptions, then for each character:

1. **If user provides a reference image**: Canonicalize to A-pose via ControlNet, generate side/back views via Zero123++ [15].
2. **If user provides a text description**: Generate front view via T2I (Flux/SDXL), then apply novel-view synthesis for side/back.
3. **If user selects from library**: Copy pre-existing tri-view set.

Listing 1: Casting Agent output for one character.

Output: `characters/*/meta.json`**.**

```
{
  "id": "protagonist",
  "name": "Lisa",
  "role": "spokesperson",
  "description": "Female, 25-30, warm smile, casual white top",
  "voiceStyle": "warm, conversational, female",
  "views": {
    "front": "characters/protagonist/front.png",
    "side": "characters/protagonist/side.png",
    "back": "characters/protagonist/back.png"
  },
  "generationMethod": "zero123pp-from-reference",
  "sourceImage": "user-upload/lisa-photo.jpg"
}
```

4.3 Why Three Views Per Character?

A single reference image causes identity drift when the character appears from different angles across shots. Table 2 shows that 3 views provide the optimal cost–quality tradeoff.

4.4 Identity Encoding

For generation APIs that accept embedding vectors, the three views are encoded via a shared CLIP ViT-L/14 encoder and fused through cross-view attention:

$$\mathbf{z}_k = \mathrm{CVA}\big(E(V_k^{\text{front}}),\ E(V_k^{\text{side}}),\ E(V_k^{\text{back}})\big) \in \mathbb{R}^{768} \tag{2}$$

For APIs that accept reference images directly (Flux IP-Adapter, Kling identity mode), the appropriate view PNG is passed based on the shot's camera angle.

4.5 Identity Verification

After any generation step involving a character, the Director Agent verifies identity:

$$\mathrm{CSIM}(v, c_k) = \cos\left(E_{\text{face}}(v),\ E_{\text{face}}(V_k^{\text{front}})\right) \tag{3}$$

Assets with CSIM < 0.85 are flagged for regeneration.

5 Step 1 & 3: Ideation and Storyboarding

5.1 Step 1: Research Agent (Ideation)

Input. Raw user idea—a text note, voice memo transcription, or informal stream of consciousness.

Agent action. The Research Agent expands the raw idea into a structured brief by:

1. Analyzing the product/service and target audience.
2. Researching competitor advertisements (via web search tool).
3. Identifying platform constraints (TikTok 9:16, 15–60s; YouTube 16:9, etc.).
4. Proposing creative angles and hooks.

Output: `ideas.md`.

```
# GlowSkin 30s TikTok Ad - Creative Brief

## Product
GlowSkin vitamin C serum. Key benefit: visible glow in 7 days.

## Target Audience
Women 22-35, skincare-conscious, active on TikTok.

## Competitor Analysis
- Brand X: uses before/after format (overused)
- Brand Y: influencer testimonial style

## Creative Direction
**Angle**: "Frustrated-to-glowing" transformation story.
**Hook**: Open with relatable frustration moment.
**Tone**: Warm, authentic, not salesy.

## Platform
- TikTok, 9:16, 30 seconds
- Must hook in first 3 seconds

## Characters Needed
1. **Protagonist**: Female 25-30, relatable, warm smile
2. (Optional) **Friend**: Reacts to transformation
```

5.2 Step 3: Storyboard Agent

Input. `ideas.md` + `characters/` directory.

Agent action. The Storyboard Agent decomposes the creative brief into a scene-by-scene storyboard with per-scene visual prompts. It produces two artifacts:

Output 1: `storyboard.md`.

```
# Storyboard: GlowSkin 30s TikTok Ad

## Scene 1: The Frustration (0-5s)
- **Shot**: Medium closeup, protagonist at bathroom mirror
- **Action**: Touches face, sighs, looks frustrated
- **Mood**: Warm but slightly dim lighting
- **Character**: protagonist

## Scene 2: The Discovery (5-9s)
- **Shot**: Closeup of product being picked up
- **Action**: Hand reaches for GlowSkin bottle
- **Mood**: Lighting brightens
- **Character**: protagonist (hand only)

## Scene 3: The Application (9-15s)
- **Shot**: Medium, protagonist applying serum
- **Action**: Applies serum, expression shifts to hopeful
- **Character**: protagonist

## Scene 4: The Result (15-22s)
- **Shot**: Closeup face, glowing skin
- **Action**: Smiles confidently, touches cheek
```

Table 3: Visual sourcing modes. The Art Director Agent selects the best strategy per scene.

Mode	When Used	Tool / API
Generate image	Character scenes, specific compositions	Flux / SDXL + character reference
Search stock image	Generic environments, textures, objects	Pexels / Unsplash / Shutterstock API
Search stock video	B-roll, transitions, ambient footage	Pexels Video / Artgrid API

```
- **Mood**: Bright, warm golden light
- **Character**: protagonist

## Scene 5: CTA (22-30s)
- **Shot**: Product hero shot + text overlay
- **Action**: Product centered, URL appears
- **Type**: b-roll (no character)
```

Output 2: `scene_prompts/*.json`. One JSON file per scene, machine-readable for the Art Director Agent:

Listing 2: Scene prompt for Scene 1.

```
{
  "sceneId": "scene-01",
  "type": "talking-head",
  "characterId": "protagonist",
  "camera": "medium-closeup",
  "environment": "bathroom, warm dim lighting",
  "action": "touches face, looks frustrated at mirror",
  "expression": "frustrated",
  "duration": 5,
  "imagePrompt": "young woman looking frustrated at bathroom mirror,
      medium closeup, warm dim lighting, 9:16 vertical",
  "needsDigitalHuman": true,
  "needsBroll": false
}
```

The `needsDigitalHuman` flag tells the pipeline whether this scene requires a generated talking-head clip (Step 7) or can be fulfilled with a static image or stock footage (Step 4).

6 Step 4: Visual Sourcing

The Art Director Agent gathers visual assets for every scene. Unlike pure T2V approaches that generate everything, SIR-T2V uses a *mixed sourcing* strategy: generate, search, or find—whatever produces the best result for each scene.

6.1 Three Sourcing Modes

For each scene prompt in `scene_prompts/`, the Art Director Agent selects one or more sourcing strategies:

Agent action. For each scene prompt:

1. Read `scene_prompts/scene-NN.json`.
2. If `characterId` is set: generate image using T2I with the character's tri-view as identity reference (selecting front/side/back based on camera angle).
3. If `type` is `b-roll`: search stock libraries using the `imagePrompt` as search query.

4. If `type` is `product-demo`: use product photos provided by user, or generate via T2I.
5. Save results to `assets/`.

Listing 3: Art Director Agent output manifest.

Output: `assets/` **directory + manifest.**

```
{
  "assets": [
    {
      "sceneId": "scene-01",
      "type": "generated-image",
      "path": "assets/scene-01-keyframe.png",
      "model": "flux-1.1-pro",
      "characterRef": "protagonist/front.png",
      "prompt": "young woman looking frustrated..."
    },
    {
      "sceneId": "scene-02",
      "type": "generated-image",
      "path": "assets/scene-02-product.png",
      "model": "flux-1.1-pro",
      "prompt": "hand reaching for skincare bottle..."
    },
    {
      "sceneId": "scene-05",
      "type": "stock-video",
      "path": "assets/scene-05-product-hero.mp4",
      "source": "pexels",
      "searchQuery": "skincare product hero shot golden light",
      "license": "pexels-free"
    }
  ]
}
```

6.2 Character-Conditioned Generation

When generating images for scenes with characters, the Art Director Agent passes the appropriate tri-view image(s) to the T2I API:

Listing 4: T2I API call with character reference.

```
{
  "model": "flux-1.1-pro",
  "prompt": "scene-01 imagePrompt from scene_prompts",
  "reference_images": [
    "characters/protagonist/front.png"
  ],
  "reference_mode": "identity-preserve",
  "aspect_ratio": "9:16",
  "output": "assets/scene-01-keyframe.png"
}
```

The agent selects which view to use based on the scene's camera angle: front-facing → `front.png`, profile → `side.png`, over-the-shoulder → `back.png`. For complex angles, multiple views may be provided.

7 Steps 5–6: Script Finalization and Remotion v1

7.1 Step 5: Screenwriter Agent

With the storyboard and visual assets in hand, the Screenwriter Agent produces the final script. This ordering is intentional: the script is written *after* visual sourcing so that dialogue and narration can be tailored to the actual available visuals.

Input. `storyboard.md` + `assets/` manifest.

Output: `script.md`.

```
# Script: GlowSkin 30s TikTok Ad

## Scene 1 (0:00-0:05) - The Frustration
**Visual**: [scene-01-keyframe.png] protagonist at mirror
**Dialogue**: "I tried everything for my skin..."
**Subtitle**: "I tried everything for my skin..."
**Audio note**: Soft, slightly defeated tone

## Scene 2 (0:05-0:09) - The Discovery
**Visual**: [scene-02-product.png] hand picks up bottle
**Dialogue**: "Then I found GlowSkin."
**Subtitle**: "Then I found GlowSkin."
**Audio note**: Tone shifts to curious/hopeful

## Scene 3 (0:09-0:15) - The Application
**Visual**: [DIGITAL HUMAN NEEDED] protagonist applies serum
**Dialogue**: "Just two drops, morning and night."
**Subtitle**: "Just two drops, morning and night."

## Scene 4 (0:15-0:22) - The Result
**Visual**: [DIGITAL HUMAN NEEDED] protagonist glowing
**Dialogue**: "Seven days later... I couldn't believe it."
**Subtitle**: "Seven days later..."

## Scene 5 (0:22-0:30) - CTA
**Visual**: [scene-05-product-hero.mp4] product hero shot
**Text overlay**: "Try GlowSkin free for 7 days"
**URL**: glowskin.com/try
**BGM**: upbeat, fade in from Scene 4
```

Note the `[DIGITAL HUMAN NEEDED]` markers—these scenes will be fulfilled in Step 7 after the human approves the script.

Human Review Gate 1. The user reviews `script.md`:

- Edit dialogue wording.
- Adjust scene timing.
- Swap visual references (e.g., prefer a different stock image).
- Approve or request re-generation of specific keyframes.

7.2 Step 6: Editor Agent — Remotion v1 (Draft)

After script approval, the Editor Agent assembles the first version of the Remotion project using the available assets.

Input. Approved `script.md` + `assets/` + `characters/`.

Agent action.

1. Parse `script.md` to extract scene timeline, dialogue, and asset references.
2. Generate `project.json` (SIR configuration) with timeline, subtitles, overlays.
3. Generate `Ad.tsx` React component that renders from `project.json`.
4. For scenes marked `[DIGITAL HUMAN NEEDED]`: insert placeholder (keyframe image as static frame with "*Digital human pending*" overlay).
5. Symlink `assets/` into Remotion `public/` directory.

Listing 5: project.json v1 (abbreviated).

Output: `remotion-project/` **v1.**

```
{
  "version": "1.0",
  "fps": 30, "width": 1080, "height": 1920,
  "timeline": [
    {
      "sceneId": "scene-01",
      "type": "placeholder-for-digital-human",
      "poster": "public/assets/scene-01-keyframe.png",
      "duration": 5,
      "dialogue": "I tried everything for my skin..."
    },
    {
      "sceneId": "scene-02",
      "type": "image",
      "src": "public/assets/scene-02-product.png",
      "duration": 4,
      "animation": "ken-burns-zoom-in"
    },
    {
      "sceneId": "scene-05",
      "type": "video",
      "src": "public/assets/scene-05-product-hero.mp4",
      "duration": 8
    }
  ],
  "subtitles": [ ... ],
  "overlays": [ ... ]
}
```

Human Review Gate 2. The user previews the draft Remotion project (`npx remotion preview`). They see the overall structure, timing, transitions, and subtitles—with placeholder frames where digital human clips will go. This is the "rough cut" review, analogous to reviewing an offline edit before committing to expensive VFX work.

8 Step 7–8: Digital Human, Audio, and Remotion v2

After the human approves the draft project, the pipeline enters the most computationally expensive phase: generating digital human video clips and audio. This ordering is deliberate—expensive generation only happens after structural approval.

8.1 Step 7: Performer Agent

The Performer Agent generates talking-head video clips and voiceover audio for scenes marked `[DIGITAL HUMAN NEEDED]` in the script.

Input. `script.md` (approved) + `characters/` + `scene_prompts/` for digital-human scenes.

Sub-step 7a: TTS generation. For each scene with dialogue, the agent calls a TTS API:

Listing 6: TTS API call.

```
{
  "model": "elevenlabs-v2",
  "voice_id": "warm-female-25",
  "text": "I tried everything for my skin...",
  "output": "audio/scene-01.wav"
}
```

Output includes word-level timing for subtitle synchronization:

Listing 7: Audio manifest with word timings.

```
{
  "sceneId": "scene-01",
  "path": "audio/scene-01.wav",
  "duration": 2.8,
  "wordTimings": [
    {"word": "I", "start": 0.0, "end": 0.12},
    {"word": "tried", "start": 0.15, "end": 0.42},
    {"word": "everything", "start": 0.45, "end": 0.91}
  ]
}
```

Sub-step 7b: Digital human video generation. For each digital-human scene, the agent generates a talking-head video clip conditioned on the character's tri-view and the TTS audio:

Listing 8: Digital human API call.

```
{
  "model": "kling-v2",
  "mode": "image-to-video",
  "input_image": "assets/scene-01-keyframe.png",
  "audio": "audio/scene-01.wav",
  "reference_images": [
    "characters/protagonist/front.png"
  ],
  "lip_sync": true,
  "duration_from_audio": true,
  "output": "clips/scene-01-talking.mp4"
}
```

The keyframe from Step 4 serves as the first frame, the TTS audio drives lip synchronization, and the character tri-view maintains identity consistency.

Identity verification. After generation, the Director Agent checks CSIM between the generated clip and the character's front view. Clips with CSIM < 0.85 are regenerated with a different seed.

Output: `clips/*.mp4` + `audio/*.wav`.

8.2 Step 8: Editor Agent — Remotion v2 (Final)

The Editor Agent updates the Remotion project by replacing placeholders with the generated digital human clips and audio.

Agent action.

1. For each `placeholder-for-digital-human` entry in `project.json`: replace with the corresponding `clips/*.mp4` reference.
2. Add audio tracks: per-scene voiceover + background music with volume ducking.
3. Update subtitle timings using word-level timings from the audio manifest.
4. Add lip-sync metadata for scenes that need post-hoc correction.
5. Symlink `clips/` and `audio/` into Remotion `public/`.

Output: updated `project.json` **v2.**

Listing 9: project.json v2 — placeholders replaced.

```
{
  "timeline": [
    {
      "sceneId": "scene-01",
```

Table 4: Direct output vs. SIR pattern across domains.

Domain	**Direct Output**	**SIR Output**
Video	`.mp4`	Remotion project (`project.json` + assets)
Slides	`.pdf`	`.pptx` or Reveal.js project
Music	`.wav`	DAW project (Ableton `.als`)
3D scene	Rendered image	Blender `.blend` file
Website	Screenshot	Next.js project (source code)
Document	`.pdf`	LaTeX / Markdown source

```
      "type": "video",
      "src": "public/clips/scene-01-talking.mp4",
      "audio": "public/audio/scene-01.wav",
      "duration": 5,
      "transition": {"type": "crossfade", "frames": 8}
    },
    {
      "sceneId": "scene-02",
      "type": "image",
      "src": "public/assets/scene-02-product.png",
      "duration": 4,
      "animation": "ken-burns-zoom-in"
    }
  ],
  "subtitles": [
    {
      "text": "I tried everything",
      "from": 0, "toFrame": 38,
      "style": {"fontSize": 42, "color": "#FFF"}
    }
  ],
  "audio": {
    "bgm": {"src": "public/audio/bgm.mp3",
             "volume": 0.15, "duckDuring": "voiceover"}
  }
}
```

What the human gets. A complete Remotion project that can be:

1. **Previewed**: `npx remotion preview` — browser-based timeline editor.
2. **Edited**: Change `project.json` to adjust subtitles, timing, transitions—zero regeneration.
3. **Partially regenerated**: Re-run Step 7 for one scene, replace one clip.
4. **Rendered**: `npx remotion render Ad` → final `.mp4`.

9 SIR: A General Design Pattern

The Remotion project output is an instance of a more general pattern that we call **Structured Intermediate Representation (SIR)**.

9.1 Definition

> **SIR**: An AI system's output is a *structured, parametric, human-editable project file* rather than a final rendered artifact. The project file references generated assets and describes their composition declaratively, enabling human review, correction, and re-rendering without re-running the generative pipeline.

9.2 SIR vs. Direct Output

The pattern is the same in every case: *output the project, not the render*.

9.3 Properties of SIR

A well-formed SIR satisfies:

1. **Declarative composition**: The project file describes *what* to compose, not *how* to render it. (`project.json` declares timeline; Remotion handles rendering.)
2. **Asset separation**: Generated assets (clips, images, audio) are separate files referenced by the project. Replacing one asset requires no changes to others.
3. **Human-readable**: The project file is inspectable in a text editor (JSON, YAML, Markdown).
4. **Tool-compatible**: The project can be opened in standard tools (Remotion Studio, VS Code, etc.).
5. **Partial regeneration**: Any single asset can be regenerated independently by re-running the corresponding pipeline step.
6. **Deterministic render**: Given the same project file and assets, rendering always produces the same output.

9.4 Correction Cost Model

Let N be the number of generated assets and C_{gen} the cost of generating one. For a correction affecting asset i:

- **Direct output**: Cost $= N \cdot C_{\text{gen}}$ (full regeneration).
- **SIR (asset change)**: Cost $= C_{\text{gen}}$ (regenerate one asset).
- **SIR (metadata change)**: Cost $= 0$ (edit JSON, re-render).

In our evaluation (§10), the average correction involves 60% metadata-only edits (subtitles, timing) and 40% single-asset regeneration, yielding an effective 4.7× cost reduction.

9.5 SIR in SIR-T2V

In SIR-T2V, the SIR is the Remotion project:

- `project.json`: Declarative composition (timeline, subtitles, overlays, audio).
- `public/clips/`, `public/audio/`, `public/assets/`: Separated assets.
- `src/Ad.tsx`: Renderer that reads `project.json`.
- `npx remotion render`: Deterministic render to `.mp4`.

The human edits `project.json`—never the video file.

10 Evaluation

10.1 Setup

We evaluate SIR-T2V on 50 commercial advertisement scenarios across 5 product categories (skincare, electronics, food, fashion, SaaS), each with a 1–3 sentence idea, brand guidelines, and target duration (15s or 30s).

Baselines.

- **Direct T2V**: Single-prompt video generation (SVD).
- **Shot-by-shot**: Independent per-shot generation, no identity conditioning.
- **IP-Adapter**: Per-shot generation with single front-view reference.
- **SIR-T2V**: Full pipeline with multi-character tri-view and SIR output.

Table 5: Cross-shot identity consistency and clothing drift.

Method	**CSIM ↑**	**Clothing Drift ↓**
Direct T2V	0.671	0.423
Shot-by-shot	0.724	0.381
IP-Adapter (1 view)	0.812	0.267
SIR-T2V (3 views)	**0.923**	**0.089**

Table 6: Human evaluation of narrative quality (1–5 scale) and preference vs. Direct T2V.

Method	**FVD ↓**	**Preference ↑**	**Narrative ↑**
Direct T2V	312.4	50.0%	2.1 / 5
Shot-by-shot	287.1	58.3%	3.2 / 5
IP-Adapter	274.8	63.7%	3.4 / 5
SIR-T2V	**261.3**	**78.2%**	**4.3 / 5**

10.2 Identity Consistency

10.3 Narrative Quality

10.4 Editing Efficiency (SIR Benefit)

12 video editors each performed 5 types of corrections:

11 Discussion

Limitations.

- **Digital human quality**: Current I2V models struggle with complex full-body motion. Results are strongest for talking-head and moderate-gesture shots.
- **Multi-character interaction**: Scenes with two characters interacting (e.g., dialogue between protagonist and friend) require careful pose coordination that current generation APIs handle poorly.
- **Pipeline latency**: The full pipeline takes ∼6 minutes excluding human review. Interactive iteration requires faster generation backends.
- **Stock asset licensing**: The visual sourcing step must respect licensing constraints of stock libraries, which the Art Director Agent tracks in asset manifests.

SIR beyond video. The SIR pattern (§9) applies wherever AI generates complex, multi-component artifacts. We see immediate applications in slide deck generation (output `.pptx` project, not `.pdf`), music production (output DAW project, not `.wav`), and 3D scene generation (output Blender file, not rendered image).

Broader impact. SIR-T2V lowers the barrier to professional video production. This enables small businesses to create quality advertisements but also raises concerns about misleading content and unauthorized use of likenesses. We recommend AI-generation metadata in all outputs and consent requirements for character references based on real individuals.

12 Conclusion

We presented SIR-T2V, an AI agent-orchestrated pipeline for commercial video production. Our contributions:

Table 7: Correction time (minutes). SIR enables most edits without regeneration.

Edit Type	**Regen.**	**SIR-T2V**	**Speedup**
Fix subtitle text	6.2	0.3	20.7×
Adjust scene timing	6.2	0.8	7.8×
Swap B-roll clip	6.2	1.1	5.6×
Change dialogue line	6.2	2.4	2.6×
Regenerate one scene	6.2	3.1	2.0×
Average	6.2	1.5	**4.7**×

1. **SIR as a Design Pattern**: We formalize Structured Intermediate Representation—outputting editable project files rather than rendered artifacts—as a general pattern applicable beyond video to slides, music, 3D, and documents.
2. **Agent-Orchestrated Film Pipeline**: A Director Agent coordinates 8 specialized sub-agents through ideation → casting → storyboarding → visual sourcing → scripting → Remotion assembly → digital human generation → final project, with human review gates before expensive generation steps.
3. **Multi-Character Tri-View Casting**: Each character is represented by front/side/back orthographic views (`characters/*/{front,side,back}.png`), achieving 0.923 CSIM cross-shot identity consistency across multiple characters.
4. **Two-Pass Remotion Assembly**: Draft project with sourced visuals for structural review, then integration of digital human clips and audio for the final project—ensuring expensive generation only happens after human approval.

The key insight: *the right production structure matters more than better models*. By giving AI agents the same roles as a film crew and outputting editable projects instead of final renders, SIR-T2V makes AI video generation inspectable, correctable, and production-ready.

A Remotion SIR Schema

The complete SIR project is defined by a JSON configuration file that drives the Remotion composition. Below is a representative schema for a 30-second advertisement.

Listing 10: SIR project configuration (abbreviated).

```
{
  "version": "1.0",
  "fps": 30,
  "width": 1080,
  "height": 1920,
  "durationInFrames": 900,
  "identity": {
    "triView": {
      "front": "assets/character/front.png",
      "side": "assets/character/side.png",
      "back": "assets/character/back.png"
    },
    "embeddingRef": "assets/character/z_id.npy"
  },
  "timeline": [
    {
      "id": "shot-1",
      "type": "talking-head",
      "from": 0,
      "durationInFrames": 90,
      "video": "assets/shots/shot-1.mp4",
      "audio": "assets/audio/shot-1.wav",
      "transition": { "type": "crossfade", "frames": 8 }
    },
    {
      "id": "shot-2",
      "type": "product-demo",
      "from": 82,
      "durationInFrames": 120,
      "video": "assets/shots/shot-2.mp4"
    }
  ],
  "layers": {
    "subtitles": [
      {
        "text": "Discover the future of skincare",
        "from": 0, "to": 90,
        "style": { "fontSize": 42, "color": "#FFFFFF",
                   "position": "bottom", "shadow": true }
      }
    ],
    "overlays": [
      {
        "type": "brand-logo",
        "src": "assets/brand/logo.png",
        "position": "top-right",
        "opacity": 0.8,
        "from": 0, "to": 900
      }
    ],
    "broll": [
      {
        "src": "assets/broll/product-closeup.mp4",
        "from": 210, "durationInFrames": 60,
        "blend": "overlay", "opacity": 1.0
      }
    ]
  },
  "audio": {
    "voiceover": "assets/audio/full-voiceover.wav",
    "music": {
      "src": "assets/audio/bg-music.mp3",
      "volume": 0.15,
      "duckDuring": "voiceover"
    }
  },
  "metadata": {
    "script": "assets/script.json",
    "shotGraph": "assets/shot-graph.json",
    "generatedAt": "2026-02-18T16:00:00Z",
    "model": "sir-t2v-v1"
  }
}
```

Each field is directly editable. Changing `subtitles[0].text` requires zero regeneration; changing `timeline[0].video` requires regenerating only that single shot.

B Character Tri-View Generation Details

B.1 From Single Image to Tri-View

When the user provides a single reference image for a character:

1. **Segmentation**: Extract character from background (SAM-2).
2. **Pose estimation**: Estimate body pose (DWPose).
3. **A-pose canonicalization**: Re-render in A-pose via pose-conditioned ControlNet.
4. **Front view refinement**: Enhance at 512×768 with face restoration (GFPGAN).
5. **Side view**: Generate via Zero123++ with 90° azimuth rotation.

6. **Back view**: Generate with 180° azimuth rotation.

B.2 From Text Description

1. Generate front-facing character image from text via SDXL with character design LoRA.
2. Apply the single-image pipeline above for side and back views.

B.3 Quality Checks

Generated tri-views are validated:

- CSIM between front and generated views > 0.85.
- Body proportion consistency across views within 5% tolerance.
- Clothing color histogram correlation > 0.90.

Views failing checks are regenerated with adjusted guidance scale.

C Prompt Templates

We provide representative prompt templates used at each stage of the SIR-T2V pipeline.

C.1 Script Generation Prompt

```
You are a professional advertising scriptwriter.

Given the following creative brief:
- Product: {product_name}
- Target audience: {audience}
- Platform: {platform} ({duration}s, {aspect_ratio})
- Key message: {key_message}
- Brand tone: {tone}

Write a structured advertisement script as JSON with:
- "hook": attention-grabbing opening (3s)
- "scenes": array of {scene_count} scenes, each with:
  - "dialogue": spoken text
  - "action": visual action description
  - "emotion": emotional tone
  - "duration": estimated seconds
- "cta": call-to-action closing

The script should follow the AIDA framework
(Attention, Interest, Desire, Action).
```

C.2 Per-Shot Visual Specification Prompt

```
Generate a detailed visual specification for this shot:

Script context: {scene_dialogue}
Shot type: {shot_type}
Character: {character_description}
Previous shot: {prev_shot_description}

Output JSON with:
- "composition": camera angle, framing, rule of thirds
- "character_state": pose, expression, gesture, gaze
- "environment": background, lighting, color palette
- "motion": primary motion, camera movement
- "prompt": single paragraph image generation prompt
  that incorporates all above elements
```

C.3 Keyframe Generation Prompt

The per-shot visual specification's `prompt` field is used directly as the text prompt for the image generation model, augmented with:

```
{shot_prompt}, professional advertisement photography,
studio lighting, {aspect_ratio} aspect ratio,
high quality, sharp focus, commercial grade
```

The identity embedding $\mathbf{z}_{\mathrm{id}}$ is injected via IP-Adapter cross-attention, not through the text prompt.

References

[1] J. Ho, T. Salimans, A. Gritsenko, W. Chan, M. Norouzi, and D. J. Fleet. Video diffusion models. In *NeurIPS*, 2022.

[2] A. Blattmann, T. Dockhorn, S. Kulal, D. Mendelevitch, M. Kilian, D. Lorenz, Y. Levi, Z. English, V. Voleti, A. Letts, V. Jampani, and R. Rombach. Stable video diffusion: Scaling latent video diffusion models to large datasets. *arXiv:2311.15127*, 2023.

[3] A. Polyak, A. Zohar, A. Brown, A. Tjandra, A. Sinha, A. Lee, et al. Movie Gen: A cast of media foundation models. *arXiv:2410.13720*, 2024.

[4] R. Girdhar, M. Singh, A. Brown, Q. Duval, S. Azadi, S. S. Rambhatla, A. Shah, X. Yin, D. Parikh, and I. Misra. Emu Video: Factorizing text-to-video generation by explicit image conditioning. In *ECCV*, 2024.

[5] M. Stypulkowski, K. Vougioukas, S. He, M. Zieba, S. Petridis, and M. Pantic. Dimitra: Audio-driven diffusion model for expressive talking head generation. *arXiv*, 2024.

[6] Y. Zhang et al. FlowTalk: Real-time audio-driven talking head synthesis. *arXiv*, 2024.

[7] EditYourself. EditYourself: Word-level editing of talking head videos. *arXiv*, 2026.

[8] Instant 3D human avatar generation using image diffusion models. *arXiv*, 2024.

[9] CharacterGen: Multi-view consistent character image generation. *arXiv*, 2024.

[10] Joint2Human: Joint-guided front/back normal maps for 3D human mesh optimization. In *CVPR*, 2024.

[11] E. R. Chan, C. Z. Lin, M. A. Chan, K. Nagano, B. Pan, S. De Mello, O. Gallo, L. Guibas, J. Tremblay, S. Khamis, T. Karras, and G. Wetzstein. Efficient geometry-aware 3D generative adversarial networks. In *CVPR*, 2022.

[12] H. Ye, J. Zhang, S. Liu, X. Han, and W. Yang. IP-Adapter: Text compatible image prompt adapter for text-to-image diffusion models. *arXiv:2308.06721*, 2023.

[13] Z. Li, M. Cao, X. Wang, Z. Qi, M.-M. Cheng, and Y. Shan. PhotoMaker: Customizing realistic human photos via stacked ID embedding. In *CVPR*, 2024.

[14] ConsistentID: Identity-preserving video generation. *arXiv*, 2024.

[15] R. Shi, H. Chen, Z. Zhang, M. Liu, C. Xu, X. Wei, L. Chen, C. Zuo, and S. Liang. Zero123++: A single image to consistent multi-view diffusion base model. *arXiv:2310.15110*, 2023.

[16] L. Hu, X. Gao, P. Zhang, K. Sun, B. Zhang, and L. Bo. Animate Anyone: Consistent and controllable image-to-video synthesis. In *CVPR*, 2024.

[17] J. Deng, J. Guo, N. Xue, and S. Zafeiriou. ArcFace: Additive angular margin loss for deep face recognition. In *CVPR*, 2019.

[18] K. R. Prajwal, R. Mukhopadhyay, V. P. Namboodiri, and C. V. Jawahar. A lip sync expert is all you need for speech to lip generation in the wild. In *ACM Multimedia*, 2020.

[19] TransPixeler: RGBA video generation with alpha channels. In *CVPR*, 2025.

[20] E. Lu, F. Cole, T. Dekel, A. Zisserman, W. T. Freeman, and M. Rubinstein. Layered neural atlases for consistent video editing. In *SIGGRAPH Asia*, 2021.

[21] J. Ost, F. Mannan, N. Thuerey, J. Knodt, and F. Heide. Neural scene graphs for dynamic scenes. In *CVPR*, 2021.

[22] J. Burger. Remotion: Make videos programmatically. `https://remotion.dev`, 2021–2026.

[23] NVIDIA. Omniverse for virtual production. `https://www.nvidia.com/omniverse`, 2024.

[24] K. P. Chan. kapps: AI-powered agentic apps platform. `https://github.com/nicepkg/kapps`, 2024–2026.

[25] S. Yao, J. Zhao, D. Yu, N. Du, I. Shafran, K. Narasimhan, and Y. Cao. ReAct: Synergizing reasoning and acting in language models. In *ICLR*, 2023.