Rugby Vision - Requirements Specification

Executive Summary

Rugby Vision is a multi-camera 3D forward pass detection system designed to assist Television Match Officials (TMOs), referees, and analysts in making accurate decisions about forward passes in rugby matches. The system processes synchronized video from multiple camera angles, reconstructs 3D positions of players and the ball, and applies physics-based criteria to determine if a pass is forward according to rugby laws.

Primary Users

1. Television Match Official (TMO)

- **Context**: In stadium control room during live matches
- **Need**: Quick, accurate decision support for contentious pass calls
- **Requirements**:
- Clear FORWARD / NOT FORWARD indicator
- Confidence score to assess reliability
- Simple interface requiring minimal training
- Results within seconds (Phase 3)

2. Referees

- **Context**: Post-match review and training
- **Need**: Understanding of marginal decisions for learning
- **Requirements**:
- Detailed explanation of decision reasoning
- Visual representation of ball trajectory
- Ability to review multiple angles

3. Analysts / Coaches

- **Context**: Match analysis and performance review
- **Need**: Detailed statistical data on pass patterns
- **Requirements**:
- Debug data and raw metrics
- Export capabilities
- Historical comparison tools (future)

Key Workflows

Workflow 1: Analyse Single Pass (Primary)

Actors: TMO, Referee, Analyst

Preconditions:
- Multi-camera video footage available

- Cameras are calibrated (intrinsic/extrinsic parameters known)
- Time window of pass identified

Steps:

1. **User loads clip**
   - System accepts multiple video sources (files or streams)
   - System validates video format and quality
   - System displays video player with multi-camera view

2. **User selects pass event**
   - User marks start and end times of pass
   - OR System auto-detects pass events (Phase 2+)
   - User selects which cameras to use (minimum 2)

3. **System processes pass**
   - **Stage 1**: Ingest and synchronize frames from selected cameras
   - **Stage 2**: Detect players and ball in each frame
   - **Stage 3**: Track objects across frames (maintain IDs)
   - **Stage 4**: Reconstruct 3D positions using camera calibration
   - **Stage 5**: Compute ball trajectory and player positions
   - **Stage 6**: Apply forward pass criteria
   - **Stage 7**: Generate decision with confidence and explanation

4. **System displays results**
   - Primary indicator: FORWARD (red) or NOT FORWARD (green)
   - Confidence percentage (0-100%)
   - Text explanation of decision reasoning
   - Optional: 3D visualization overlay on video
   - Optional: Top-down field view with trajectory

5. **User reviews decision**
   - User can step through frames
   - User can toggle camera views
   - User can access debug data (analysts only)

Postconditions:
- Decision logged for audit trail
- Results available for export

Error Scenarios:
- Insufficient cameras (< 2): Error message prompts for more cameras
- Ball not detected: System reports low confidence, suggests manual review
- Calibration missing: System prompts for calibration data or uses defaults
- Processing timeout: System cancels and reports error

Workflow 2: Batch Analysis (Future)

Actors: Analyst

Description: Analyse multiple passes across an entire match for statistical reporting.

Deferred to: Phase 2+

Workflow 3: Live Match Integration (Future)

Actors: TMO

Description: Analyse passes in near-real-time during live match with TMO trigger.

Deferred to: Phase 3

Functional Requirements

FR1: Multi-Camera Video Ingestion

Priority: MUST HAVE

Description: System shall accept video from multiple camera sources and synchronize frames.

Acceptance Criteria:
- Support MP4, MOV, AVI video formats (Phase 1)
- Support RTSP/RTMP streams (Phase 2+)
- Minimum 2 cameras, maximum 8 cameras
- Handle frame rate differences (interpolation/decimation)
- Synchronize frames within ±33ms (1 frame at 30fps)
- Handle missing frames gracefully (skip or interpolate)

FR2: Player and Ball Detection

Priority: MUST HAVE

Description: System shall detect players and ball in each camera frame.

Acceptance Criteria:
- Detect all visible players (minimum precision: 80%)
- Detect ball when visible (minimum precision: 60% - known hard problem)
- Provide bounding boxes and confidence scores
- Process at minimum 15fps equivalent

FR3: Object Tracking

Priority: MUST HAVE

Description: System shall track players and ball across frames maintaining consistent IDs.

Acceptance Criteria:
- Maintain track IDs for at least 3 seconds continuous
- Handle occlusions (players obscured temporarily)
- Re-identify after brief occlusion (within 1 second)

FR4: 3D Reconstruction

Priority: MUST HAVE

Description: System shall compute 3D positions of ball and players in field coordinates.

Acceptance Criteria:
- Use camera calibration parameters (intrinsic + extrinsic)
- Triangulate 3D points from multi-view 2D detections

- Transform to field coordinate system (origin at try line, x=width, y=length, z=height)
- Accuracy: ±0.5m horizontal, ±0.2m vertical (Phase 1 target)

FR5: Forward Pass Decision Logic

Priority: MUST HAVE

Description: System shall determine if pass is forward based on 3D trajectory and physics.

Acceptance Criteria:
- Identify pass start time (ball leaves passer's hands)
- Identify pass end time (ball caught by receiver or hits ground)
- Compute ball displacement vector in field coordinates
- Account for player momentum (ball carries forward momentum of passer)
- Provide binary decision: FORWARD or NOT FORWARD
- Provide confidence score (0.0 to 1.0)
- Provide text explanation of decision

Decision Criteria (Phase 1 - Simplified):
1. Measure ball position when leaving passer's hands: `B_start`
2. Measure ball position when caught/grounded: `B_end`
3. Compute displacement along field: `Δy = B_end.y - B_start.y`
4. If `Δy > threshold` (e.g., 0.3m): FORWARD
5. If `Δy < -threshold`: NOT FORWARD
6. If `|Δy| ≤ threshold`: borderline, lower confidence

Future Enhancement (Phase 2+):
- Incorporate passer velocity vector
- Use metric tensor for relativistic-like treatment (player reference frame)

FR6: User Interface

Priority: MUST HAVE

Description: System shall provide web-based UI for TMOs and analysts.

Acceptance Criteria:
- Video player with multi-camera view (or switchable views)
- Timeline with pass event markers
- Decision indicator (prominent, color-coded)
- Confidence percentage display
- Text explanation display
- Responsive design (works on tablets)

Phase 1: Basic static UI with manual controls

Phase 2+: Interactive timeline, frame stepping, 3D visualization

FR7: Camera Calibration Management

Priority: SHOULD HAVE (Phase 1), MUST HAVE (Phase 2+)

Description: System shall load and manage camera calibration parameters.

Acceptance Criteria:
- Accept calibration files (JSON/YAML format)

- Store intrinsic matrix (focal length, principal point, distortion)
- Store extrinsic matrix (rotation, translation relative to field)
- Validate calibration completeness before processing

Phase 1: Manual upload of pre-calibrated parameters

Phase 2+: Semi-automatic calibration from field markings

FR8: Logging and Audit Trail

Priority: SHOULD HAVE

Description: System shall log all decisions for transparency and review.

Acceptance Criteria:
- Log each analysis request (clip ID, cameras, time window, user)
- Log decision result (forward/not, confidence, timestamp)
- Store logs for minimum 30 days
- Exportable as CSV/JSON

FR9: Debug Data Access (Analysts)

Priority: COULD HAVE (Phase 1), SHOULD HAVE (Phase 2+)

Description: System shall provide detailed debug data for analysts.

Acceptance Criteria:
- Per-frame detections (bounding boxes, classes, confidence)
- 3D point clouds over time
- Ball trajectory data (position, velocity)
- Intermediate computation results
- Downloadable as JSON

Non-Functional Requirements

NFR1: Performance - Latency

Phase 1 (Offline POC):
- Target: Analysis completes within **10 seconds** for 5-second clip
- Acceptable: Up to 30 seconds for complex clips

Phase 2 (Semi-Real-Time):
- Target: Analysis completes within **5 seconds** for 3-second clip
- Acceptable: Up to 10 seconds

Phase 3 (Live Integration):
- Target: Analysis completes within **3 seconds** for 2-second clip
- Critical: Maximum 5 seconds (TMO decision window)

Performance Budget (Phase 1):
- Video ingestion & sync: 2 seconds
- Detection & tracking: 4 seconds
- 3D reconstruction: 2 seconds
- Decision computation: 1 second
- UI rendering: 1 second

NFR2: Accuracy

Detection Accuracy:
- Player detection: 80% precision, 85% recall (Phase 1)
- Ball detection: 60% precision, 50% recall (Phase 1 - ball is small/fast)
- Improvement targets: 90%+ for both (Phase 2+)

3D Reconstruction Accuracy:
- Horizontal position error: ±0.5m (Phase 1)
- Vertical position error: ±0.2m (Phase 1)
- Improvement targets: ±0.2m horizontal, ±0.1m vertical (Phase 2+)

Decision Accuracy:
- Target: 85% agreement with expert human referees (Phase 1)
- Target: 90%+ agreement (Phase 2+)
- False positive rate (calling legal pass forward): <10%
- False negative rate (missing forward pass): <15%

NFR3: Reliability

- System uptime: 99% (Phase 2+)
- Graceful degradation: If ball not detected, provide low confidence result
- Error recovery: If processing fails, return clear error message within 2 seconds

NFR4: Scalability

Phase 1: Single-instance deployment, process one clip at a time

Phase 2+:
- Support concurrent analysis of 5+ clips
- Horizontal scaling via worker pool
- Queue-based architecture for batch processing

NFR5: Usability

- TMO interface: Usable with **<5 minutes** training
- Analyst interface: Usable with **<30 minutes** training
- Visual design: High contrast, large buttons, accessible
- Error messages: Clear, actionable, non-technical language for TMOs

NFR6: Compatibility

Video Formats (Phase 1):
- MP4 (H.264), MOV, AVI
- Resolution: 720p minimum, 4K maximum
- Frame rates: 24fps to 60fps

Browser Support:
- Chrome 100+
- Firefox 100+
- Safari 15+
- Edge 100+

Server Environment:
- Linux (Ubuntu 22.04+)

- Docker containers
- Cloud-deployable (AWS, GCP, Azure)

NFR7: Security

Phase 1 (POC): Minimal security (localhost only, no authentication)

Phase 2+:
- HTTPS for all communication
- JWT-based authentication
- Role-based access control (TMO, Referee, Analyst)
- Video content not stored (process and discard)
- Audit logs encrypted at rest

Constraints

Technical Constraints

1. **Camera Requirements**:
   - Minimum 2 cameras per analysis
   - Cameras must have overlapping field of view
   - Calibration parameters must be available

2. **Computational Resources**:
   - GPU recommended for detection (NVIDIA with CUDA)
   - Minimum 16GB RAM
   - Minimum 4 CPU cores

3. **Network Requirements** (Phase 2+):
   - Minimum 10 Mbps per camera stream
   - Low latency network (<50ms)

Business Constraints

1. **Cost**:
   - Phase 1: Use open-source models (YOLO, etc.)
   - Avoid expensive commercial licenses

2. **Timeline**:
   - Phase 1 POC: 4-6 weeks
   - Phase 2: 8-12 weeks
   - Phase 3: 16-20 weeks

Legal/Regulatory Constraints

1. **Data Privacy**:
   - No storage of personal identifiable information
   - Match footage rights respected (process only, don’t store)

2. **Officiating Integration**:
   - System is decision **support**, not decision **maker**
   - Final decision authority remains with human referee

Phase Boundaries

Phase 1: Offline POC (Current)

Goal: Prove core concept with recorded clips

Scope:
- Load pre-recorded video files
- Manual pass event selection
- Basic 2-camera 3D reconstruction
- Simple forward pass decision logic
- Mock/synthetic data for testing

Success Criteria:
- System produces correct decision for 10/10 test cases (obvious passes)
- Latency <10 seconds per analysis
- Clean, functional UI
- Complete documentation

Out of Scope:
- Live streaming
- Automatic pass detection
- Advanced physics models
- Real match footage (use synthetic)

Deliverables:
- Working monorepo with frontend + backend + ML
- Video ingestion and sync modules
- Basic detection and 3D reconstruction (can be mocked)
- Decision engine with simple logic
- UI with decision display
- Docker deployment setup
- Complete documentation

Phase 2: Semi-Real-Time

Goal: Process near-live streams with automatic pass detection

Scope:
- RTSP/RTMP stream ingestion
- Automatic pass event detection (ML model)
- Improved 3D reconstruction accuracy
- Multi-clip concurrent processing
- Enhanced UI with frame stepping

Success Criteria:
- Latency <5 seconds
- 85%+ accuracy on validation set
- Support 3+ concurrent analyses

Deliverables:
- Stream ingestion pipeline
- Automatic event detection model

- Improved tracking and 3D algorithms
- Performance optimizations (GPU, parallelization)

Phase 3: Fully Integrated Live System

Goal: Production-ready system for live match integration

Scope:
- Sub-3 second latency
- TMO workflow integration
- Advanced metric tensor physics model
- 3D visualization overlays
- Broadcast graphics integration

Success Criteria:
- 90%+ accuracy
- <3 second latency
- 99% uptime
- Approved by rugby unions for official use

Deliverables:
- Production-grade deployment
- TMO hardware integration
- Real-time visualization
- Comprehensive monitoring and alerting

Interfaces Between Components

1. Frontend ↔ Backend

Protocol: HTTP REST API

Endpoints:

`POST /api/clip/analyse-pass`
- Input: `{ clip_id, cameras, start_time, end_time }`
- Output: `{ is_forward, confidence, explanation, metadata }`

`GET /api/clip/{clip_id}/debug-data`
- Output: `{ detections, tracks, 3d_points, trajectory }`

`GET /health`
- Output: `{ status }`

2. Backend ↔ Video Ingestion

Interface: Python class `VideoIngestor`

```python
class VideoIngestor:
    def load_sources(self, sources: List[VideoSource]) -> None
    def get_synchronized_frames(self, timestamp: float) -> List[Frame]
    def release(self) -> None
```

3. Video Ingestion ↔ Detection/Tracking

Interface: Frame batches

```
@dataclass
class Frame:
    camera_id: str
    timestamp: float
    image: np.ndarray
    frame_number: int
```

4. Detection/Tracking ↔ 3D Reconstruction

Interface: Detection lists

```
@dataclass
class Detection:
    camera_id: str
    frame_id: int
    bbox: Tuple[int, int, int, int]
    class_name: str
    confidence: float
    track_id: Optional[int]
```

5. 3D Reconstruction ↔ Decision Engine

Interface: Frame state

```
@dataclass
class FrameState:
    timestamp: float
    ball_pos_3d: Optional[np.ndarray]  # [x, y, z]
    players_pos_3d: Dict[int, np.ndarray]  # track_id -> [x, y, z]
```

6. Decision Engine → Backend

Interface: Decision result

```
@dataclass
class DecisionResult:
    is_forward: bool
    confidence: float
    explanation: str
    metadata: Optional[Dict[str, Any]]
```

Assumptions

1. **Camera Setup**: Cameras are professionally installed with known calibration
2. **Field Markings**: Field markings are visible and standard dimensions
3. **Lighting**: Adequate lighting conditions (broadcast quality)
4. **Video Quality**: Minimum 720p resolution, 30fps
5. **Ball Visibility**: Ball is visible in at least 2 cameras for majority of pass
6. **Network**: Stable network for video streaming (Phase 2+)

Risks and Mitigations

Risk	Impact	Probability	Mitigation
Ball detection accuracy too low	High	Medium	Use multiple cameras, ensemble models, manual fallback
3D reconstruction errors	High	Medium	Validate with synthetic data, provide confidence scores
Latency exceeds targets	High	Low	Performance profiling, GPU acceleration, frame subsampling
Camera calibration unavailable	Medium	Low	Auto-calibration from field markings, use approximate defaults
User interface too complex	Medium	Low	User testing with TMOs, iterative design
Real footage differs from synthetic	High	Medium	Early validation with real match footage samples

Success Metrics

Phase 1 Success Metrics

- **Technical**:
- 10/10 correct decisions on synthetic test cases
- <10 second latency per analysis
- 0 critical bugs in core pipeline
- **Usability**:
- 3/3 test users can complete workflow without help
- Positive feedback on UI clarity
- **Delivery**:
- All documentation complete
- Docker deployment functional
- Code review approval on all modules

Future Metrics (Phase 2+)

- Decision accuracy: 85%+ agreement with expert referees
- Processing latency: <5 seconds (Phase 2), <3 seconds (Phase 3)
- System uptime: 99%+
- User satisfaction: 4/5 rating from TMOs

Appendix: Glossary

- **TMO**: Television Match Official - official who reviews video footage for referee
- **Forward Pass**: Pass where ball travels toward opponent's goal line (illegal in rugby)
- **Camera Calibration**: Mathematical description of camera's intrinsic and extrinsic parameters
- **Triangulation**: Computing 3D position from multiple 2D observations
- **Track ID**: Unique identifier for an object tracked across frames
- **Confidence Score**: Numerical measure (0-1) of system's certainty in decision
- **Field Coordinates**: 3D coordinate system aligned with rugby field (origin at try line)