

Feature Documentation: Guided Capture Session

1. Overview

The **Guided Capture Session** is an intelligent workflow designed to assist operators in acquiring a series of high-quality, stable images of a patient's upper and lower dental arches.

Instead of relying on manual clicks or video recording, this feature uses real-time motion analysis to **automatically capture images** only when the camera is held perfectly steady. This significantly improves the quality and consistency of the captured data by minimizing motion blur and ensuring optimal framing.

The system guides the user through a predefined sequence: scanning the lower arch, followed by the upper arch, providing clear on-screen instructions and audio cues at each step.

2. User Guide: Step-by-Step Workflow

This section describes the process from the operator's perspective.

2.1. Starting the Session

1. Click the **"Start Session"** button on the main screen.
2. The main camera view will be overlaid with the capture workflow interface.

2.2. Phase 1: Lower Arch Scan

1. **Initial Instruction:** The screen will display "Place the scanner at the leftmost tooth of your Lower arch and Start." and play a corresponding audio instruction. The camera feed will be flipped vertically for intuitive control during the lower scan.
2. **Begin Scanning:** The operator positions the scanner and clicks the **"Start Lower Scan"** button.
3. **Real-Time Guidance:** The system is now in an active scanning state. The bottom guidance bar will provide real-time feedback:
 - "Ready to capture" (Green): The system is stable and waiting for a capture.
 - "Hold steady to capture..." (Cyan): The camera is perfectly stable, and a capture is imminent (within 1 second).
 - "Slow down" / "Keep steady" (Amber/Red): The camera is moving too fast or being shaken. No captures will occur.
4. **Automatic Captures:** Whenever the operator holds the scanner steady for the required duration (1 second), the system will:
 - Automatically capture and save an image.
 - Play a capture sound.
 - Display a subtle **corner flash effect** on the screen as visual confirmation.

5. **Finish Scanning:** Once the operator has scanned the entire lower arch, they click the **"Finish Lower Scan"** button.

2.3. Phase 2: Upper Arch Scan

1. **Transition Instruction:** The screen will display "Excellent. Now, place the scanner at the leftmost tooth of your Upper arch and Start." and play the next audio instruction. The camera feed will revert to its normal orientation.
2. **Begin Scanning:** The operator positions the scanner and clicks the **"Start Upper Scan"** button.
3. **Real-Time Guidance & Auto-Capture:** The process is identical to the lower arch scan. The system will automatically capture stable images as the operator moves along the upper arch.
4. **Finish Scanning:** After scanning the upper arch, the operator clicks the **"Finish Upper Scan"** button.

2.4. Completing the Session

1. **Completion Message:** The screen will display "Scan Complete!".
 2. **Finish:** The operator clicks the **"Finish Session"** button. The overlays disappear, the guidance system stops, and the application returns to the main idle screen.
-

3. Technical Deep Dive

This section details the internal logic and architecture of the feature.

3.1. Core Components

- **UI Layer:** Manages the ScanningState enum, user interactions (button clicks), and updates the UI based on the current state.
- **Processing Layer:** A background service responsible for all real-time motion analysis. It is completely decoupled from the UI's state machine. It exposes two key events:
 - GuidanceUpdated: Fires continuously to provide UI feedback (prompts, colors).
 - OnCaptureTriggered: Fires **only** when a valid capture condition is met.
- **ScanningState Enum:** A state machine in the UI layer that tracks the workflow progress (ReadyToScanLower, ScanningLower, ReadyToScanUpper, ScanningUpper, Complete).

3.2. Pause and Resume Logic (Efficiency)

To save CPU resources, the GuidanceSystem is not constantly processing frames. Its processing is controlled by the UI's state machine.

- When the session starts, `_guidanceSystem.Start()` is called. This creates the worker threads, but they remain dormant.
- When the user clicks "Start Lower/Upper Scan", the UI calls `_guidanceSystem.SetProcessingActive(true)`. This "un-pauses" the system, and it begins analyzing frames.
- When the user clicks "Finish Lower/Upper Scan", the UI calls `_guidanceSystem.SetProcessingActive(false)`. This "pauses" the system, and the worker threads go back to sleep, consuming no CPU.
- This is more efficient than repeatedly stopping and starting threads.

3.3. The "Three Gates of Capture" Mechanism

The `OnCaptureTriggered` event is the core of the auto-capture logic. It only fires after passing three sequential checks inside the `StateAndCaptureWorker`:

1. **Gate 1: Absolute Stability Check**
 - **Condition:** The real-time smoothed motion values must be below strict, predefined thresholds.
 - `motionState.Mu < CAPTURE_SPEED_THRESH` (e.g., 4.0)
 - `motionState.Sigma < CAPTURE_STAB_THRESH` (e.g., 3.0)
 - **Purpose:** Ensures the image will be free of motion blur from both linear movement and shakiness.
2. **Gate 2: The "Hold Steady" Timer**
 - **Condition:** The "Absolute Stability" condition from Gate 1 must be maintained continuously for `CAPTURE_DELAY_S` (e.g., 0.5 second).
 - **Purpose:** Confirms the user's intent to capture and prevents accidental shots from brief moments of stillness.
3. **Gate 3: The Cooldown Timer**
 - **Condition:** At least `CAPTURE_COOLDOWN_S` (e.g., 1.5 seconds) must have passed since the last successful capture.
 - **Purpose:** Prevents a rapid-fire burst of near-identical images and paces the workflow, encouraging the user to reposition after each shot.

Only when all three gates are passed does `OnCaptureTriggered?.Invoke()` execute, telling the UI layer it is time to save an image.

4. Configuration & Tuning Parameters

These constants, located in `GuidanceSystem.cs`, allow for fine-tuning the system's performance and behavior.

Constant	Default	Purpose & Tuning Guidelines
MOTION_TARGET_WIDTH	480	Performance vs. Accuracy. The width the camera frame is scaled down to for analysis. Lowering it (e.g., to 320) significantly reduces CPU usage but makes motion detection less sensitive. Increasing it (e.g., to 640) improves sensitivity at a high CPU cost. 480 is a good balance.
LK EVERY_N	3	Performance vs. Responsiveness. Analyzes motion only once every N frames. Increasing it (e.g., to 5) reduces CPU load but makes feedback less smooth. Decreasing it (to 1 or 2) provides more real-time data but increases CPU load.
CAPTURE_SPEED_THRESH	4.0	Capture Strictness (Linear Motion). The maximum average pixel movement allowed for a capture. Lowering it (e.g., to 2.0) enforces a stricter hold, resulting in sharper images but making captures harder. Increasing it makes the system more lenient.
CAPTURE_STAB_THRESH	3.0	Capture Strictness (Shake/Rotation). The maximum motion deviation allowed. Lowering it is less tolerant of hand tremors. Increasing it is more forgiving.
CAPTURE_DELAY_S	0.5	User Experience (Intent). The time (in seconds) the user must hold steady before a capture. Lowering it (e.g., to 0.5s) makes captures faster but potentially more accidental. Increasing it makes the process feel more deliberate.
CAPTURE_COOLDOWN_S	1.5	User Experience (Pacing). The minimum time (in seconds) between captures. Lowering it allows for faster successive shots. Increasing it enforces a slower, more methodical pace.