**Lab Exercise: Video Processing with Python**

**Objective:**

This lab will guide you through working with video in Python. You will connect to a camera, work with video files, draw geometrical figures on live video, implement callback functions, and convert video from BGR to RGB format. This exercise focuses on applying fundamental video processing techniques.

**Tasks:**

1. **Connecting to a Camera:**
   * Establish a connection to your computer's camera using a Python library. Ensure the camera feed is displayed in a window.
   * Observe and manipulate the frame rate or resolution of the video feed.
2. **Using Video Files:**
   * Load a video file and display its contents frame by frame.
   * Explore the properties of the video, such as the total number of frames, frame height, and width.
   * Try pausing, fast-forwarding, or rewinding the video at different frame intervals.
3. **Drawing Geometrical Figures on Video:**
   * While the video is playing (live or from a file), draw different geometrical figures (such as rectangles, circles, and lines) on the video frames.
   * Use user input to determine the position and size of the geometrical figures.
   * Ensure that the drawn figures update dynamically with the video feed.
4. **Use of Callback Functions:**
   * Implement a callback function that gets triggered on specific user interactions, such as a mouse click or keyboard press while the video is playing.
   * The callback should allow users to dynamically draw or modify shapes on the video feed.
5. **Conversion of BGR to RGB:**
   * Capture video either from the camera or from a video file, and convert the video frames from BGR to RGB format.
   * Display the original BGR video feed and the RGB-converted video feed side by side.
   * Observe the visual differences between the two color spaces.

**Deliverables:**

* A brief report describing the challenges faced in each task.
* Screenshots of video with geometrical shapes drawn.
* A reflection on the role of callback functions in interactive video applications.