LevelSnapp
Intuitive Camera Photography Error Reduction

Abstract

The Importance of Straightening The Horizon and Aligning Lines in Photographic Compositions
We as human beings generally prefer straight, leveled lines instead of odd angles. When one hangs a photo in their living room, they want to perfectly align it with the ceiling, their floor or furniture. TVs or computer monitors sit on flat surfaces or are fixed perfectly aligned on the wall. Having objects with the right-angle is important because the human eyes like things straight and if things looked crooked, the human sub-conscious perception is thrown out of balance. Humans seek visual balance in everything they do, including photography. That's why looking at a photo with a crooked horizon does not feel natural. A tilted image makes one want to tilt their heads to try to align those lines with the horizontal line joining their eyes. This can cause strain to the eyes and neck as they have to continuously readjust w.r.t. to their background. The following project aims to develop an app— *LevelSnapp* to look into solving this problem for photographers through visual guidance.

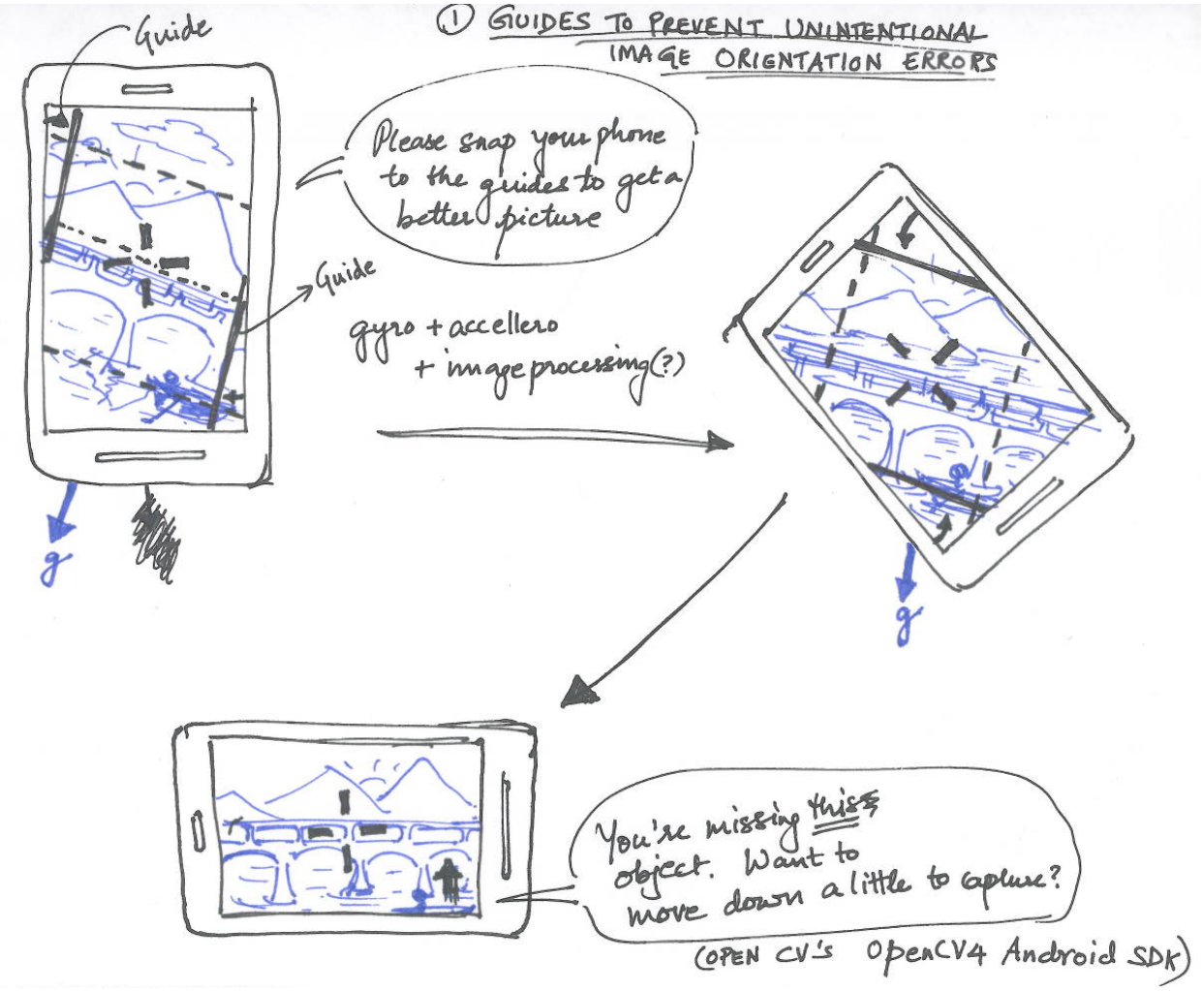
Why is what you are doing important? What is the promise of your project? Who is your project important for?

The idea of this project is to show an overlay of guides on the real time dynamic camera capture preview of the android default camera app. The guides at any point represent the lines about the centre of the screen which mark the frame (boundary) that an ideal image should take up in order to be perfectly aligned with respect to the direction of Earth's gravitational pull. Aligning images in this manner implies that the horizontal guide is tangential to the horizon and the vertical line is in the same direction as the Earth's gravitational pull. Ensuring such an alignment in photography is often important because it improves the aesthetic quality of the images and also prevents them from looking visually imbalanced or "heavy" to one side unless the photographer has intentionally desired such a tilt.

Below are a few specific scenarios where maintaining alignment with gravity might be difficult while capturing an image and a camera plugin that could assist, at least visually, in guiding against error, would be valuable.

- **Amateur photographers**: An amateur photographer might unknowingly introduce a slight tilt into his image as his/her hand may not stable enough yet, to take perfectly aligned photographs.
- **Rapid photography**: Often, there may not be enough time to make the perfect capture unaided. (Eg: Journalism)
- **Small Screen phones**: It is difficult to perceive how aligned the actual picture is going to look if the phone screen is small. Guides in such a case will behave like a spirit level. Aligning the edge of the screen with the guides envisioned in this app, however small and low resolution it may be, will help eliminate taking badly aligned pictures.

- **Photography on the Move**: Often, taking photos by a phone camera when on the move can be quite disappointing. One may see a lovely scenery they want to capture but they just don't get enough time to perfectly align the camera with the horizon or the line of Gravity. This may lead to capturing an unsatisfactory version of what could have been a great picture.



What are the obstacles? Why are existing solutions not good enough? If you are recreating an existing application/software - what were the obstacles when this existing application came out?

1. **No Existing Solutions**: There are no existing solutions to address this requirement at present. Also, the existing camera apps are not open source. Some cameras automatically adjust the horizon alignment after the photo has been snapped. However, this typically does not exploit gravitometric data that is easily obtained from the phone's accelerometer-gyroscope combination. Instead they exploit image processing algorithms to guess the horizon, usually successfully.
2. **Existing Popular Android Camera apps have not shared their Code base**: The existing Android Camera apps seem to be closed communities and do not exist on the developer framework

3. **Determining accurately, the guides via the direction of gravity**: The guides that will be displayed on the phone will essentially have to be projections on the screen, of what vectors the accelerometer defines as the vertical direction (direction of gravity)
   A. **Tilt Detector**: A tilt event is defined by the direction of the 2-seconds window average gravity changing by at least 35 degrees since the activation or the last event generated by the sensor. Here is the algorithm:
   - *reference_estimated_gravity* = average of accelerometer measurements over the first second after activation or the estimated gravity when the last tilt event was generated.
   - *current_estimated_gravity* = average of accelerometer measurements over the last 2 seconds.
   - trigger when angle(reference_estimated_gravity, current_estimated_gravity) > 35 degrees

   B. **Big changes in acceleration do not generate tilt event**: Large accelerations without a change in phone orientation do not generally trigger a tilt event. For example, a sharp turn or strong acceleration while driving a car should not trigger a tilt event, even though the angle of the average acceleration might vary by more than 35 degrees. Typically, this sensor is implemented with the help of only an accelerometer. Other sensors can be used as well if they do not increase the power consumption significantly. This is a low power sensor that should allow the SoC to go into suspend mode.
   CHALLENGE: The fact that big changes in acceleration do not generate a tilt event unless there is a change in phone orientation, might be a concern when trying to detect the tilt when flying on an aircraft. To handle such cases, the gyroscope might need to be tied to the accelerometer output to generate a custom tilt event.

Details of Idea implementation

- **Main Sensors**: I plan to use accelerometer-gyroscope to determine the line of gravity (orientation of the phone) and overlay the guides onto camera shooting screen accordingly
- **Sensors to accommodate Add Ons**: The camera itself, touch (to dismiss)
- **Main Output**: The alignment guides overlaid on top of the camera field view.
- **Storage**: The pictures clicked will have to be stored on the local file system of the phone.
- **Additional Technologies**: OpenCV (using OpenCV4 Android SDK), SnapDragon SDK for Android (for general Facial Processing and/or specifically Facial Recognition)

How does your solution tie with the concepts we have covered in lectures and labs?

- A major challenge of my solution will be to generate an appropriate struts-springs model and a Finite State Machine for the guides that will be overlain on the camera output.
- If I am to not use android's built in tilt event detector, I will have to greatly sample the accelerometer and gyro input to give me exactly what I need.

How can you make the solution more challenging?

There can be two options for upgrading my solution, of which I will pick one based on ease of implementation. They are listed in order of preference below

1. **Photographing the floor**: The solution can be made more challenging by addressing the scenario when the phone camera lens is measured to be horizontal. In such a case, the guides will disappear. However, there is still scope for improved photography in terms of alignment with lines in the image.
   - For Eg: If bold straight lines denoting room edges are detected, a different set of guides may appear and overlay the edge detected in the base image and help photographer align his/her photo accordingly.
   - Another example could be when photographing an insect on a carpet or wooden floor. The photographer may not want to skew his image with respect to the carpet/wood fibres.
   - **Additional SDK to be used**: OpenCV4 Android SDK
2. **Face recognition and Smile Valuation**: Good photographs of people often imply smiling people. Another way to make the solution more challenging it to use face detection to label the people in the photo and assess the *Smile Value* using the *Facial Processing Library of Snapdragon SDK*. Based on the degree to which the person is smiling or not smiling, the phone will call out his/her name and say something like, "Joseph, why don't you smile!", "Iuliana, why so serious?"
   - **Additional SDK to be used**: Snapdragon SDK for Android
   - Face Libraries in the Snapdragon SDK for Android: the following are provided out-of-the-box
     - **Facial Processing**— Blink Detection, Gaze Detection, Smile Value, Face Orientation)
     - **Facial Recognition**— ability to Add upto 10 people to a face album, ability to detect those people in an image or video stream
   - Additional Output:
     - *Screen*: boxes could be rendered on screen dynamically highlighting each face recognized in photo (Red until person smiles sufficiently. Green thereafter)
     - *Speaker*: playing pre-recorded audio instructions to smile/not smile.

How can you make the solution less challenging?

- **Only implement guides**: The solution can be made less challenging by purely concentrating on developing the guides and redrawing them dynamically with changing tilt event readings, instead of looking at the problem from the angle of a plug-in for overlaying guides when an existing camera app is launched.
- **No file storage of images captured**: Also, file storage of images will not be developed in the scoped down case. Images will be viewable only in a given session of the app.

References:

- Android Development with OpenCV— http://docs.opencv.org/2.4/doc/tutorials/introduction/android_binary_package/dev_with_OCV_on_Android.html?highlight=android

- Qualcomm Snapdragon SDK for Android— Facial Processing: https://developer.qualcomm.com/software/snapdragon-sdk-android/facial-processing
- Qualcomm Snapdragon SDK for Android— Facial Recognition: https://developer.qualcomm.com/software/snapdragon-sdk-android/facial-recognition
- Google Camera— https://play.google.com/store/apps/details?id=com.google.android.GoogleCamera&hl=en
- Motorola Camera— https://play.google.com/store/apps/details?id=com.motorola.camera&hl=en
- The Importance of Straightening the horizon and aligning lines— https://photographylife.com/straightening-the-horizon-and-aligning-lines
- Google Camera v3.0 Sneak Peek - Part 2: Camera2, Slow Motion, Dirty Lens Detection, Auto HDR+, And More [APK Teardown]— http://www.androidpolice.com/2015/09/15/google-camera-v3-0-sneak-peek-part-2-camera2-slow-motion-dirty-lens-detection-auto-hdr-and-more-apk-teardown/
- 15 Best Camera Apps for Android— http://www.androidauthority.com/best-camera-apps-for-android-188148/
- Android Camera Class— http://developer.android.com/guide/topics/media/camera.html#capture-picture
- Sensor Manager Android— http://developer.android.com/reference/android/hardware/SensorManager.html#GRAVITY_EARTH
- Base Sensor Types Android— https://source.android.com/devices/sensors/sensor-types.html
- Tilt Detection with Accelerometer in Android— https://source.android.com/devices/sensors/sensor-types.html#tilt_detector

---