IDEA 1

Intuitive Camera Photography Error Reduction

(PLEASE REFER IMAGES FOR CLEARER PICTURE)

Main Idea: Natural Alignment for Error Reduction

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 lovely picture.

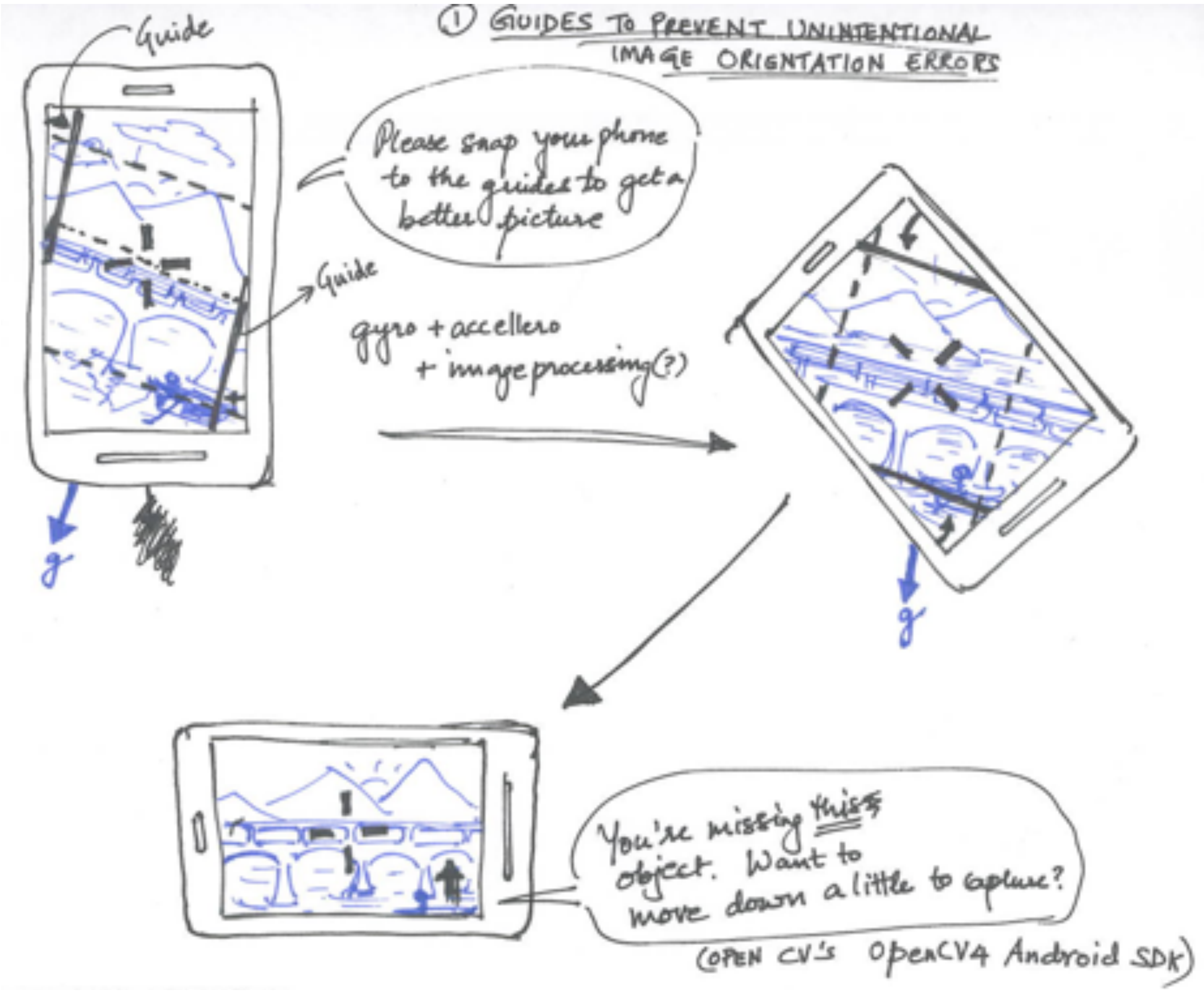
When taking photo, show guides to rotate the camera and align perfectly with the background.

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.

Additional Technologies: OpenCV (using OpenCV4 Android SDK), SnapDragon SDK for Android (for general Facial Processing and/or specifically Facial Recognition)

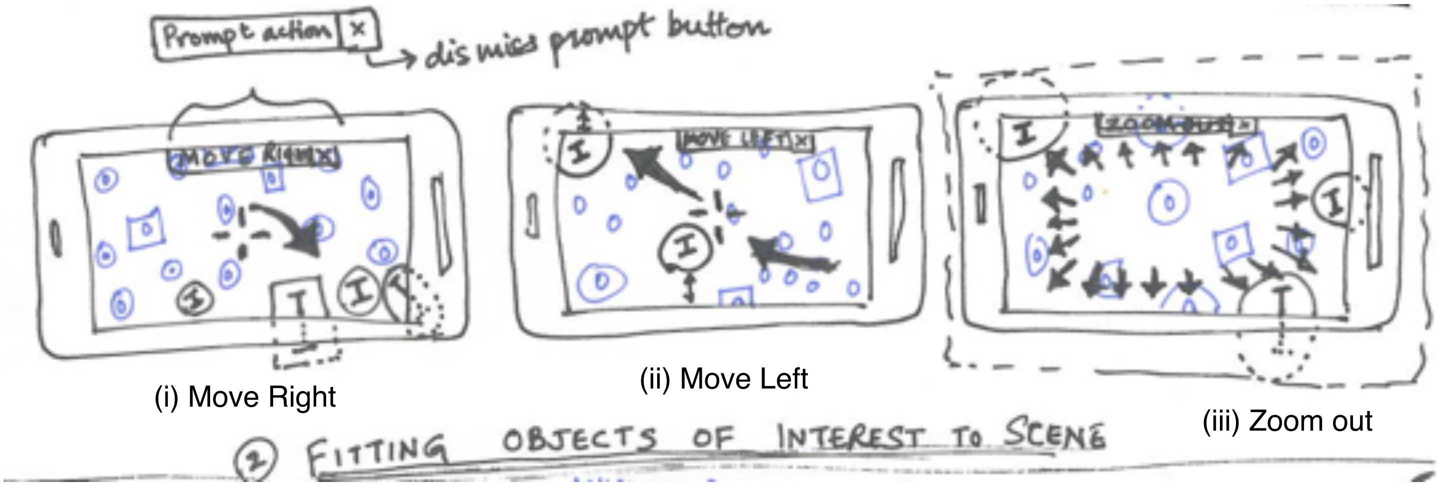

① GUIDES TO PREVENT UNINTENTIONAL IMAGE ORIENTATION ERRORS
Guide
Please snap your phone to the guides to get a better picture
Guide
gyro + accellero + image processing(?)
g
g
You're missing this~~s~~ object. Want to move down a little to capture?
(OPEN CV's OpenCV4 Android SDK)

# Add ons if the above problem ends up being too mundane:

Add-on idea 1: Don’t miss out on what could be important to capture

(PLEASE REFER IMAGES FOR CLEARER PICTURE)

very often, we photograph things that catch our eye. Certain shapes, forms, color distributions etc. could be considered as the visual parameters associated with what we see. The focus of this add on will be not on faces, but on objects in general. Imagine if your camera could help you avoid missing things that catch your eye while you are photographing something.



(i) Move Right

(ii) Move Left

(iii) Zoom out

Additional Output: rendering directional arrows on screen as an overlay over camera app

- The camera will continuously process the image and ask you to move left or right/ zoom out till the object of interest fits on screen.
- Alternately, if you know better what you are imaging, the prompt system can be dismissed by touch of a button/ swiping away of prompt window.
  - It won’t restart till the next time you launch camera.
  - You could also force turn it on.
  - Option will be given to always keep it off.
- How will the camera know what I’m interested in?
  - Initially, it might abide by a set of assumed rules regarding catchy objects (Based on **Gestalt’s Laws** talking about **emphasis**)
  - Over time, the system could potentially learn what catches YOUR eye in particular:
    - **Training Set**: Images you have taken on your device where no faces or human bodies were detected (In future, maybe the training set could be what you took on ANY device such that it went onto some cloud)
    - **Learning Parameters**: Maybe evaluation of different pixel parameters of the image interpreted in the form of the primary **Gestalt’s Laws**

Add-on Idea 2: Accessibility for the blind

- Speak out guiding instructions to the blind when accessibility is turned on. (Eg: “Hey Bob, good going! Tilt your phone about 20 degrees clockwise/ a **little**clockwise” - ”Little more…” - “You’re almost there. Just a tiny bit more…”

- **Additional Output**: Speaker to speak out the audio prompts
- Different blind people will have a different perception of "little". Some will under-estimate, some will over-estimate. Two comments on this problem :
  - They can correct the camera position and later, quality of their move by learning through their error. Eg: If Bob rotated too far CW instead of "little more", the reference line will ask him: "That was too much. Move a little Clockwise to orient perfectly"
  - Calibration of Feedback with Machine Learning: Machine learning can be used for automatically adjusting the audio feedback spoken out to guide the blind person with rotation and translation of his phone. Since blind people cannot see, their perception of very little - little - 90 degrees - 180 degrees etc may vary to very large degree.
    - **Training Set**: Errors (the measured angles) made by the blind person for every audio cue/prompt
    - **Learning Parameter**: just the angular measurement by which the blind person was off the phone generated command

---

IDEA 2

Morse Code Messaging system

Imagine if you could take advantage of the fact that you and your friend/confederate out there, both know Morse Code. Imagine if thanks to this asset, you didn't have to turn on the screen of your phone to read your friend's message. You could let the phone lie snugly in your pocket, and it would buzz based on the morse code message it wants to give you. Another day, you wanted to send your friend a very confidential message when you were traveling in a crowded bus. You simply decided to tap the message to him/her on Morse Code instead of writing explicit characters.

Yet another day, your fingers were wet and there was no way you could tap out a message. So you decided to jerk your phone to generate the Morse Code instead.

The sender: Taps/ touches on his screen in morse sequence to generate inputs.[Press = dot; Long Press = dash]. Alternatively the app could also support jerking the phone, where every jerk would correspond to one morse coded signal. [Small jerks = dots, exaggerated jerks = dashes]

The receiver: Receives the morse code in the form of long(dashes) and short(dots) phone vibrations (So, too bad if vibration mode is switched off). If he/she has enabled beeps, he/she will hear the corresponding beeps.

(Optional): When he/she looks at his/her phone screen, he/she sees the decoded message in text form.

Options for Transmission of message:

1. **A dedicated app that does just this—**
   1. Sender's phone: The tap sequences could be sent as such over a network, maybe in dot-dash or 1-0 format. There is no likely gain by processing and decoding the morse message to text on the sender's phone.
   2. Receiver's phone: The app will simply receive the transmitted sequence and accordingly trigger vibrate and beep events. It will also, process and decode the message from Morse Coded form into text and display in the chat thread between the two people on the app.
   3. Downside 1: From usability perspective, for something as specific as this, user will be forced to install a separate app and create a separate user login and maintain a separate chat/ text messaging system. (As if Hangouts, WhatsApp, Viber, WeChat, Messenger and SMS weren't enough separate threads to manage already).

    4. <u>Downside 2</u>: Also, your intended receiver will not only need to have this app installed and signed up for too, but also would have to know Morse Code.
    5. <u>Downside 3</u>: You can’t transmit your message without internet connection.

2. **Simply a Morse Keyboard**—An option for SMS or any other text messaging system. Installs just like SwiftKey/ PinYin keyboards.
    1. <u>Sender’s phone</u>: The tap/jerk sequence will have to be decoded at sender’s end first, converted into text, and then sent over the mobile network or internet as plain text within the framework of the pre-existing messaging app.
    2. Receiver’s phone:
    - The text will be displayed straight away on the receiver’s messaging app.
    - The text will have to be processed again and converted into morse code form. This morse code will then generate appropriate beeps and vibrations
    - The regular notification associated with the pre-existing messaging app will be suppressed when this particular keyboard is active.
    - <u>Downsides</u>: Double processing of the Morse Code. But maybe that’s okay and worth the convenience and scalability of the concept.

<u>Input</u>: Press, Long Press, jerk

<u>Output</u>: text, controlled system notification beep(s), controlled vibration

---

<u>IDEA 3</u>

Stroke Rehabilitation Game on Mobile

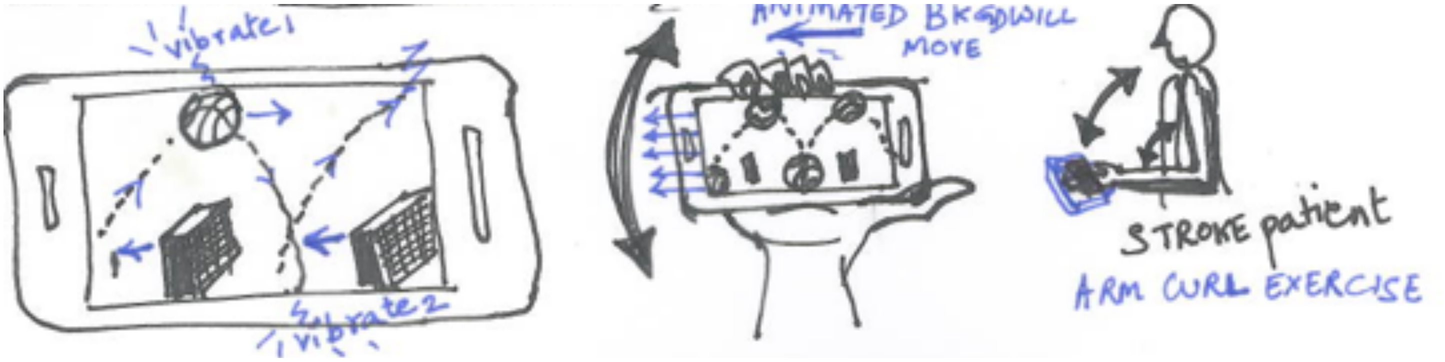
(PLEASE REFER IMAGES FOR CLEARER PICTURE)

<u>Game Description:</u>

Stroke patients often get paralyzed on one side and are unable to move one or more limbs. There are many standard clinician prescribed exercises, one of which is the Arm-Curl exercise.

To provide an immersive rehab experience with the arm-curl exercise, a simple Mobile Game can be used.

The game will mainly comprise of an animated background screen and game hurdles (like in a hurdle race) that continuously are moving to the left, giving the player the impression that he is moving to the right.



As the patient does the arm curl exercise and curls his arm up and down, the ball will bounce down and up respectively, on the screen.

Whenever the ball touches the edge of screen (bounces off the lower or upper edge of the screen), the phone vibrates to give user feedback of having completed one repetition of the exercise.

If patient is not doing any exercise, the ball will remain floating and stationary and patient won't gain points.

Calibration of arm movement:

Therapy is a gradual process where patients are able to curl their arms by different amounts at different phases of their rehabilitation exercises. Hence, it is important to ensure that the acceleration of the bouncing ball (g) is calibrated according to the maximum range that can currently be reached by patient. Lower the movement range of the patient, higher should be the gravity in the game, so that ball moves a larger distance on screen for a given angle traversed by the arm of the patient in a certain time. As the patients gets better and his range increases, the gravity decreases so that the patient now required to curl his arm by a larger angle to get the same displacement of the ball.

The game can also incorporate reduction of range and the number of repetitions if it detects that the patient is fatigued/ idle. (When no sensor input change is detected/ the patient underperforms several times in continuation)

Input: accelerometer-gyroscope combination

Output: Ball bouncing within the game

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