Baby Suffocation Monitor

Vika Baranovsky, Daniel Maryakhin, Matan Fintz

1. Abstract

Babies are exposed to many hazardous situations in their first months. One of them is suffocation during sleep while in their crib when they are with their nose towards the mattress without clear air way. There are many other gadgets on the market for monitoring the baby, such as Baby Sense, but they only alert AFTER the baby is not breathing and not before. We wanted to offer an alternative product that can foresee the danger before it occurs. The idea was having 2 simple everyday webcams that can be found in most houses, using them to detect the baby while in its crib and follow its face's angles relatively to the mattress and alert when it might be going to turn its face down to the mattress or cover its face.

2. Process

a. First of all, we performed an intrinsic calibration to the cameras by taking shots of a board with a chessboard pattern on it and detecting it. The calibration gave us the camera matrix of each of our two cameras. These values are constant for the camera's life and don't need to be calculated again.
b. We then placed the two cameras in their positions and performed an extrinsic calibration by taking a shot of a chessboard on both cameras simultaneously, detecting it and using the SolvePnPRansac algorithm. This helps us understand where the cameras are placed in the world (in relation to the board) and get their rotation matrices and translation vectors.
c. Combining the camera matrices with the rotation and translation yields us the projection matrices of the cameras.
d. After that, we take an initial stereo shot of the baby and ask the user to pin the colors in both frames (one per camera). The BGR

(the OpenCV take on RGB) values of the colors are extracted and boundaries around them are calculated, giving a threshold around B, G and R from both sides, to allow a wider recognition window in case the colors change a little between frames (because of movement, light etc.).

e. To make sure the recognition of colors in the initial frames works – we applied some heuristics in order to improve the results. See details in the following section.
f. We then took out the centroids of the recognized colors and applied triangulation to receive 3D coordinates of the inital color locations.
g. After that, we started an open loop, in which both cameras take shots of the baby simultaneously and the colors are being detected. Here we've applied some heuristics as well. More on that later.
h. Since not all colors will be detected by each of the two cameras necessarily, we took the intersecting subset of colors and performed triangulations on their centroids.
i. We used Kabsch's algorithm to find the optimal rotation and translation of the corresponding subset of colors in the initial frame to the subset we currently have.
j. Using this rotation/translation and knowing where the center (mean) if all colors in the initial frame is, we found the location where this center would have been if we had all colors.
k. We translated the coordinates of the colors in the current frame to be shifted by the above center (needed for the next step).
l. We used Kabsch's algorithm once again to find the optimal rotation between the initial colors and the current ones.
m. We decomposed the rotation matrix we've got from the previous step into degrees.
n. We had the full 6-degrees-of-freedom representation of the baby's location and knew how it changed in comparison to the initial frame.

3. Heuristics applied in initial frame

a. Search for each color within defined boundaries of BGR values and not the specific color the user has pinned:
   Even though the sticker the user has pinned has a specific color – the BGR values the camera gets can be a little bit different due to light changes, reflection etc. In order for the camera to identify the whole sticker (and not only parts of it that have exactly the pinned color) we are looking for colors with *close* BGR values to the pinned color. The amount of closeness to the original color is configurable in the code.

b. Pinned coordinate must be inside the recognized blob of the color:
   After the user has pinned the color - we search for the biggest blob with close BGR value to the pinned color. It is possible that the camera will find another (foreign) object which has the same color. To make sure we find the sticker – we check that the pinned coordinate is inside the contours of the blob that we found.

4. Heuristics applied at each frame

a. Search for each color within defined boundaries of BGR values and not the specific color the user has pinned: (explained at “Heuristics applied in initial frame”)

b. Ignore blobs that are significantly larger than the original blob:

   While the baby is moving in his crib – he can be closer or farther from the camera, so the expected size of the sticker in the frame can change. However, the difference in size shouldn’t be very significant, thus we can improve the accuracy of recognition by ignoring blobs that are significantly larger than expected (this is a configurable parameter).

c. Search for a blob having a size close to the size of the original blob found in the initial frame:

Since the image has a lot of pixels with BGR values within the searched range – we want to make sure that the blob we find will be the sticker and not some other object. In order to do this – we search for all blobs, and among them choose the one with the closest size (area) to the original blob in the initial frame. We can do this since we know the movements of the baby won't be very big, so the size of each sticker, as seen by the cameras, won't differ a lot, and this way we can improve the accuracy of choosing the correct blob on each frame.

d. Verify that the distances between the stickers in each frame don't differ much from the original distances in the initial frame:

Right after taking and confirming the initialization frame, we calculate the distances between all pairs of stickers and save them for future reference. Since the movements of the baby aren't very big and the stickers stay on the same places on his face, the distances between each pair of stickers shouldn't change much and will be very close to the distances calculated in the initial frame. The blob will be "reliable" only if it is within proper distance range from at least 2 other blobs. If it's in proper distance range from less than 2 blobs – it's considered a recognition mistake and this blob won't be taken into account since most probably it's not a sticker.

Information on some of the algorithms we've mentioned can be found in the OprnCV documentation page:
http://docs.opencv.org/2.4/modules/calib3d/doc/camera_calibration_and_3d_reconstruction.html

5. Why tracking colors?

During the project, we attempted to solve the tracking problem by several approaches:

a. Tracking using the Viola Jones Face detection algorithm
   Our first attempt was to detect the face using Viola Jones algorithm, and then use the detection of the face to calculate its position.
   We've searched for face-features libraries, but we found only classifiers that were trained on mature faces (which are different from baby faces). In addition – the classifiers were trained only on direct facial images or profile images, and couldn't detect features on a face with another orientation.
b. Putting drawings on the baby's face and tracking the features in these drawings
   We used the SIFT feature detection algorithm to detect features in these drawings, and we've got rather accurate tracking results in various angles, but it worked only when the drawings were rather big and close to the camera. Once we decreased the size of the drawings or tried to move the camera farther away – the features were too small to be detected.
c. Tracking geometric shapes
   Another approach that we tried was to track the face using geometric shapes.
   We put stickers with different geometric shapes (Circle, rectangle, triangle, etc.…) on the face and tried to detect them. The results we've got weren't accurate since the geometric shape that the camera catches can change according to the baby's angle. Also, when part of the shape was hidden – in most cases the cameras couldn't recognize it correctly.

Due to all those problems, we decided it's best to do the tracking with colors since it gave us the best results among all the other options.