A CNN-Inertial-based Odometry for Agile 6D Pose Estimation.

J. Arturo Cocoma-Ortega[1], L. Oyuki Rojas-Perez[1] and Jose Martinez-Carranza[1,2]

***Abstract*— In this work, we present a CNN-inertial-based Odometry system that fuses Visual 3D pose estimation with inertial-based orientation estimation. For the visual 3D pose, we propose an Inception-based architecture that takes a sequence of two consecutive images and predicts the relative 3D motion between them. As the FPV VIO Challenge require, our system can estimates 6D pose at real-time up to 50 Hz allowing the estimation of agile flights.**

I. System Overview

The proposed system is based on the integration of 3D pose (translation) estimated by a CNN and the orientation estimation by IMU data (see figure 1). The network developed is a two-branch architecture based on inception modules that take as input a sequence of two consecutive images, one for each branch, and predicts by regression the relative motions between them. The inertial data is used as input for the orientation estimation algorithm. The combination of the translation estimated by CNN and orientation is taken by a module that calculates work 6D pose.

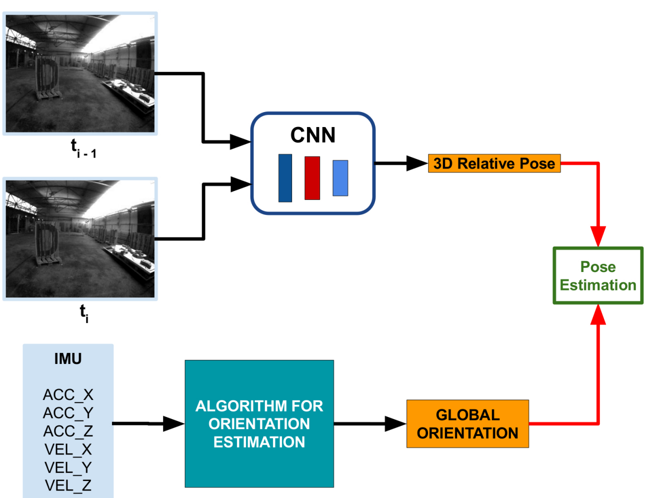

Fig. 1. General overview of the proposed system

II. Pose Estimation

Our proposed network is based on inception modules [1]; the network estimates the relative motion of the camera between two consecutive frames. We generates the labels from ground truth data as follows:

$$p_i^c = Rot^T(q_{i-1})(p_i^w - p_{i-1}^w) \quad (1)$$

where p_i^c is the relative motion between p_i^w, p_{i-1}^w; p_i^w is the position in the world; q_{i-1} is the orientation in the world and Rot^T is the transpose rotation matrix of q.

The pose p^c is relative to camera motion, to generate the motion in the world we use the world orientation q^w,thus:

$$p_i^w = Rot(q^w) * p_i^c + p^w \quad (2)$$

III. Orientation Estimation

The orientation is calculated by Madgwick algorithm [2], that fuse accelerometer and gyroscope data to estimate Inertial Measurement Unit (IMU) orientation using a gradient descent algorithm. The IMU's world orientation is expressed as quaternion $q_{imu_i}^w$. Figure 2, shows a diagram of the orientation estimation.

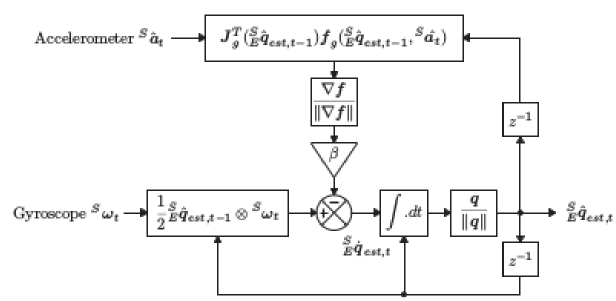

Fig. 2. Block diagram representation of the complete orientation estimation algorithm for an IMU implementation [2]

IV. Results

For the experiments, we train and test in a Laptop with a GTX860m (640 CUDA cores, 2GB VRAM), Intel $i7-$ 4710HQ (Octa-core 2.5GHz) and 16GB ram.

We work with the bag files to trajectory estimation, we process each pose as soon as it is published by image_raw topic (about 30Hz); but our system is not limited by this frequency, our system can process data up to 50Hz by the use of the GPU.

Figure 3 shows the estimated trajectories for outdoor face forward dataset.

[1] Instituto Nacional de Astrofísica, Óptica y Electrónica (INAOE), Puebla, México. Department of Computer Science at INAOE. Email addresses: {cocoma, oyukirojas, carranza}@inaoep.mx
[2]University of Bristol, Bristol, UK.

TABLE I

PARAMETERS USED TO TRAIN MODEL TO LEARN TRANSLATION.

Parameters	Value
Optimiser	Adam
Epoch	500
Batch size	32
Activation function	ReLu
Learning rate	0.001

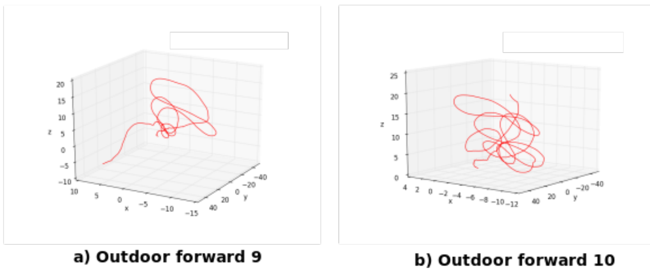

Fig. 3. Estimated trajectories for Outdoor face forward datasets.

REFERENCES

[1] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Computer Vision and Pattern Recognition (CVPR)*, 2015.

[2] S. O. H. Madgwick, A. J. L. Harrison, and R. Vaidyanathan. Estimation of imu and marg orientation using a gradient descent algorithm. In *2011 IEEE International Conference on Rehabilitation Robotics*, pages 1–7, 2011.