Implementation of vertical-rectification and CNN models for an analogic matching algorithm from a stream of images.

Salah Derrouich, Kenji Murao, Kiichiro Izumida, Kazuhisa Shiiya.

Abstract-- Recovering the three-dimension scene geometry and detecting moving targets simultaneously from a stream of images are an important tasks and have wide applicability in the creation of autonomous mobile robot, such as persistent choice of a safe route free of obstacles, targeting objects to avoid collisions, autonomous navigation and robot manipulations.

In the present work, we focus on exploiting the robustness of the analogic-array-processing-aspect introduced by the Cellular Nonlinear Network paradigm to develop a real time tracking method for a stream of general signals coming from space-distributed sources for monocular autonomous mobile robot. The motivation for developing the new tracking method is from one hand the matching operation has to be performed in real time, while from the other hand a 32 bit floating point accuracy is not that much required, which, together with a vertical rectification, as an intermediate process to minimize the token relative displacements between two frames, can lead to robust real-time object tracking system.

The technique has been successfully applied to indoor sequences of images. The results of the simulations are presented and discussed.

Index Hybrid inertial-vision, Vertical rectification, Cellular Nonlinear Network, Stream of images.

I. INTRODUCTION

The exploration of unknown natural environment by autonomous mobile robot (AMR) offers several challenges. By definition the AMR is supposed to have the ability to operate autonomously and intelligently on all kind of terrain, in other words, an operation that involve all the 6 degree of freedom (6DOF), with minimal human interaction. Thus an “onboard intelligence” capable of real-time navigation on uneven terrain and moving-target detection with a moving-sensing-system is needed for the implementation of AMR.

The wideness of artificial visual system field attracted the intention of a big number of researchers who came up with a variety of studies presenting different techniques with different sensing technologies, each with unique strengths and weakness [1]- [4].

The invention called Cellular Nonlinear Network (CNN) has emerged as a powerful and practically realizable paradigm of multidimensional, locally connected, nonlinear processor arrays [11]- [14].

We know that the ability to both track-pose and manage residual errors is unique to vision, however, from one hand, vision as passive-target-sensor suffers from a notorious lack of robustness, including subject to signal degradation such as poor lightning, and high computational expense. on the other hand vision-pose-tracking methods often compute two-dimension (2D)-image motions, since these motions are often due to rotation, inertial gyro sensor can aid the vision system in tracking these motions, and increase the robustness and computing efficiency of a vision system by providing a frame-to-frame prediction of camera orientation and position. Recently a number of research have been involved in a hybrid technique based on visual and inertial sensors [5]- [9].

To overcome the pose changes that could happen between two positions in the trajectory of an AMR and afford the same kind of information as that provided by passive navigation algorithms using artificial vision, but with a different dynamic range and precision, we previously proposed [10] the use of gyroscopes and accelerometers as active-target-sensor to improve the efficiency of vision tracking and overcome its occlusion and computation expense. The presence of Inertial-sensor makes monocular vision just as efficient as stereo vision for AMR.

In this paper a matching analogic algorithm for monocular autonomous mobile robot designed for natural terrain is presented. This algorithm projects the gyroscopes’ sensed data into a stream of images coming from a given visual static scene, using vertical rectification method. The rectified stream is then processed by a CNN algorithm, for generating vertical and horizontal paths of the stream. For testing the accuracy and usefulness of our algorithm, we estimate the image-space distance for several points from the scene.

We begin with a review of vertical rectification in section 2. Section 3 presents a brief preliminary introduction to the CNN, and its implementation in our matching method. We give the simulation results in section 4, and draw conclusion of this paper in section 5.

II. VERTICAL RECTIFICATION

People can rotate their heads very quickly, so the case of a head-mounted camera the 2D image motions are often mainly due to head rotations. Vision-pose-racking methods often compute 2D-image motions, since these motions are often due to rotations, inertial gyro sensor can aid the vision system in tracking these motions.

In both cases, mobile robot and manipulator, the displacement is bounded and several realistic physical hypotheses can be made [7].

- Inertial forces can be calculated using classical law of kinematics and dynamics.
- The gravity field is a constant, homogeneous and isotropic field of acceleration, not varying during robot displacements.
- Astronomic movements such as earth rotation have no influence as generators of force, on the robot.

Considering these hypotheses, we can separate gravity from linear acceleration. From this mechanism we can obtain an estimation of the vertical, combine it with a visual estimation of the vertical and reanalyze visual information considering this important 3D cue.

Let’s consider a 2D point \( P(i,j) \) from an image-w affected with 3 rotations \((\theta, \psi, \phi) = (\text{Pitch}, \text{Yaw}, \text{Roll})\). To exploit the vertical rectification constraint we apply the rotation motion relationship (1) between the coordinates of \( P(i,j) \) and its correction \( P(i',j') \):

\[
\begin{align*}
(i, j) &= \begin{pmatrix}
\cos \phi & -\sin \phi \\
\sin \phi & \cos \phi
\end{pmatrix} \begin{pmatrix}
f \tan \left( \frac{1-(N_x-1)/2}{f} + \theta \right) \\
f \tan \left( \frac{1-(N_y-1)/2}{f} - \psi \right)
\end{pmatrix} + \begin{pmatrix}
\frac{N_x}{2} \\
\frac{N_y}{2}
\end{pmatrix}
\end{align*}
\]

(1)

Where \( f, (N_x, N_y) \) are focal length and image resolution respectively.

A vertically rectified frame-w parallel to the absolute vertical is derived. Fig. 1(b) shows the results of applying (1) to Fig. 1(a) with \((\theta, \psi, \phi) = (-6.07, 3.54, 13.42)\). This process will rearrange the whole stream parallel to the ground-truth absolute vertical, reduces the disparity between two relative tokens in consecutive-frames, and physically simplifies the tracking phase.

### III. CNN MATCHING PROCESS

Since we are dealing with a mobile robot, to increase the geometric constraint and simplify the matching process, a stream-of-images are taken in consideration. In this paper we use a Cellular-Nonlinear-Network, as a powerful paradigm for time signals coming from space distributed sources, for matching process.

#### A. Introduction to CNN

CNN is an acronym for Cellular Neural Network, a multi-disciplinary research area, with broad applications in image and video signal processing, robotic and biological visions, and higher brain functions. More recently it has also been used as a paradigm for generating static and dynamic patterns, autowaves, spiral waves, scroll waves and spatio-temporal chaos with diverse applications in image and video signal processing. Since these latter applications are broader and not necessarily related to neural networks, it may be more appropriate to decode CNN as Cellular Nonlinear Networks in such applications.

A CNN is defined by two mathematical constructs:

1. A spatially discrete collection of continuous nonlinear dynamical systems called cells, where information can be encrypted into each cell via three independent variables called input, threshold, and initial state.
2. A coupling law relating one or more relevant variables of each cell \( C_{ij} \) to all neighbor cells \( C_{kl} \) located within a prescribed sphere of influence \( S_{ij}(r) \) of radius \( r \), centered at \( C_{ij} \).

CNNs can be either single-layer or multilayer. Consider a single layer consisting of 2D regular grid of cells \( C_{ij} \), where \( i \) and \( j \) are the row and column coordinates. The topography of such a structure is shown in Fig. 2.

Assume each cell host a processor with its real-valued input, state(s), and output signals, \( u_{ij}(t), x_{ij}(t), \) and \( y_{ij}(t) \), respectively. The simplest first-order cell state dynamics called standard CNN equation is given by

\[
\dot{x}_{ij} = -x_{ij} + z_{ij} + \sum_{C_{kl} \in S_{ij}(r)} A(ij;kl) y_{kl} + \sum_{C_{kl} \in S_{ij}(r)} B(ij;kl) u_{kl}
\]

(2)

where \( z_{ij} \in \mathbb{R} \) is called the threshold of the cell \( C_{ij} \).

\( A(ij;kl) \) and \( B(ij;kl) \) are called the feedback and control synaptic operators or templates; in case of a 3x3 neighborhood of radius 1, they are 3x3 matrices.

The state and the output signals of each cell are typically related through the following equation depicted in Fig. 3.

\[
y_{ij} = f(x_{ij}) = \frac{1}{2} \left( |x_{ij} + 1| - |x_{ij} - 1| \right)
\]

(3)

Once the cell dynamics is fixed, the interaction patterns \( A \) and \( B \) and the offset value \( z \) define the functionality of the
CNN layer. Given an input signal array $u_{ij}$ for $1 \leq i \leq M$ and $1 \leq j \leq N$, defined as a picture with pixel values $u_{ij}$, the set of values $(z, B, A)$ determines the outcome of the CNN dynamic process. This set is called a cloning template or a gene (see TEM in Fig. 4). In the space-invariant case, the templates are $3 \times 3$ (or $5 \times 5$ or $7 \times 7$) matrices. This means that a CNN array can be defined by the cell dynamics and the $19$ (or $51$ or $99$) numbers of the $A, B$ templates and the offset $z$. The input image could be either static or dynamic; hence, a CNN layer plays the role of an image processor.

The peculiar property of controlling the functionality of a whole array of interconnected cells by means of just a few interconnection weights (e.g., 19 numbers) is very familiar to neurobiologists. Indeed, the cloning template is no more than a receptive field organization in the retinotopic part of the visual pathway [15].

There exists a very wide catalog of templates covering a myriad of applications. Also, because these templates are programmable by definition, learning can be incorporated to a whole array of interconnected cells by means of just a few interconnection weights (e.g., 19 numbers) is very familiar to neurobiologists. Indeed, the cloning template is no more than a receptive field organization in the retinotopic part of the visual pathway [15].

There exists a very wide catalog of templates covering a myriad of applications. Also, because these templates are programmable by definition, learning can be incorporated to a whole array of interconnected cells by means of just a few interconnection weights (e.g., 19 numbers) is very familiar to neurobiologists. Indeed, the cloning template is no more than a receptive field organization in the retinotopic part of the visual pathway [15].

**B. CNN matching algorithm for a stream of images**

If a CNN is used as a real-time analog processor for arrays of data according to a well defined template set plus local and global logic, which represent the instructions of the CNN Universal Machine processor [12], Spatio-temporal algorithms can be defined for this processor, where a given template set is allowed to operate in a given time window, giving way to another template set able to continue to process data produced by the previous templates. Because of properties of real time computing, very complicated array data processes can be performed in very limited time frames. The CNN analog logic consists of sequential and parallel algorithmic steps. The flow diagram for solving the matching problem is shown in Fig. 6, and the genes or templates $(TEM)$ used are presented in Fig. 4 with their boundary conditions $x_{ij}$ and $u_{ij}$, where the stepwise-discontinuous function $\text{sgn}$ is as follow:

$$
\begin{align*}
\text{sgn}(x) &= 1 \text{ if } x > 0 \\
\text{sgn}(x) &= 0 \text{ if } x = 0 \\
\text{sgn}(x) &= -1 \text{ if } x < 0
\end{align*}
$$

The matching algorithm takes as input a stream of $K$ images (with $(N_x, N_y)$ as resolution), the outputs of vertical rectification algorithm, and builds as output two gray-scale images that represent the horizontal and vertical paths and have as resolution $(K, N_x)$ and $(K, N_y)$ respectively (Fig. 5).

The algorithm is as following:

- **Binary each image** $w$ with offset $z_{ij}$ equal to a threshold $d$ (in our experiments $z=0.31$) $(TEM)$
- **Normalize the Output-I with column’s height** $(N_y)$ $(TEM)$
- **Extract the vertical histogram** $(TEM)$
- **Store the sum of column’s cells in the last cell and store zero elsewhere** $(TEM)$
- **Shift one-pixel to the top** $(TEM)$
- **Project the $N_x-1$ row of Output-5 in a pre-vertical path of size $(N_x, N_y)$ $(TEM)$
- **Shift one-pixel to the top the pre-vertical path** $(TEM)$

After applying the previous seven-steps to each image $w$ $1 \leq w \leq K$ we restrict the **pre-vertical path** $(K, N_y)$ to the final output $(K, N_y)$ vertical path using $(TEM)$ $(suppose K<N_y)$.

The same process done for vertical-path extraction is done for horizontal-path with slight genes’ adjustment.

**B. CNN matching algorithm for a stream of images**

If a CNN is used as a real-time analog processor for arrays of data according to a well defined template set plus local and global logic, which represent the instructions of the CNN Universal Machine processor [12], Spatio-temporal algorithms can be defined for this processor, where a given template set is allowed to operate in a given time window, giving way to another template set able to continue to process data produced by the previous templates. Because of properties of real time computing, very complicated array data processes can be performed in very limited time frames. The CNN analog logic consists of sequential and parallel algorithmic steps. The flow diagram for solving the matching problem is shown in Fig. 6, and the genes or templates $(TEM)$ used are presented in Fig. 4 with their boundary conditions $x_{ij}$ and $u_{ij}$, where the stepwise-discontinuous function $\text{sgn}$ is as follow:

$$
\begin{align*}
\text{sgn}(x) &= 1 \text{ if } x > 0 \\
\text{sgn}(x) &= 0 \text{ if } x = 0 \\
\text{sgn}(x) &= -1 \text{ if } x < 0
\end{align*}
$$

The matching algorithm takes as input a stream of $K$ images (with $(N_x, N_y)$ as resolution), the outputs of vertical rectification algorithm, and builds as output two gray-scale images that represent the horizontal and vertical paths and have as resolution $(K, N_x)$ and $(K, N_y)$ respectively (Fig. 5).

The algorithm is as following:

- **Binary each image** $w$ with offset $z_{ij}$ equal to a threshold $d$ (in our experiments $z=0.31$) $(TEM)$
- **Normalize the Output-I with column’s height** $(N_y)$ $(TEM)$
- **Extract the vertical histogram** $(TEM)$
- **Store the sum of column’s cells in the last cell and store zero elsewhere** $(TEM)$
- **Shift one-pixel to the top** $(TEM)$
- **Project the $N_x-1$ row of Output-5 in a pre-vertical path of size $(N_x, N_y)$ $(TEM)$
- **Shift one-pixel to the top the pre-vertical path** $(TEM)$

After applying the previous seven-steps to each image $w$ $1 \leq w \leq K$ we restrict the **pre-vertical path** $(K, N_y)$ to the final output $(K, N_y)$ vertical path using $(TEM)$ $(suppose K<N_y)$.

The same process done for vertical-path extraction is done for horizontal-path with slight genes’ adjustment.

**B. CNN matching algorithm for a stream of images**

If a CNN is used as a real-time analog processor for arrays of data according to a well defined template set plus local and global logic, which represent the instructions of the CNN Universal Machine processor [12], Spatio-temporal algorithms can be defined for this processor, where a given template set is allowed to operate in a given time window, giving way to another template set able to continue to process data produced by the previous templates. Because of properties of real time computing, very complicated array data processes can be performed in very limited time frames. The CNN analog logic consists of sequential and parallel algorithmic steps. The flow diagram for solving the matching problem is shown in Fig. 6, and the genes or templates $(TEM)$ used are presented in Fig. 4 with their boundary conditions $x_{ij}$ and $u_{ij}$, where the stepwise-discontinuous function $\text{sgn}$ is as follow:

$$
\begin{align*}
\text{sgn}(x) &= 1 \text{ if } x > 0 \\
\text{sgn}(x) &= 0 \text{ if } x = 0 \\
\text{sgn}(x) &= -1 \text{ if } x < 0
\end{align*}
$$

The matching algorithm takes as input a stream of $K$ images (with $(N_x, N_y)$ as resolution), the outputs of vertical rectification algorithm, and builds as output two gray-scale images that represent the horizontal and vertical paths and have as resolution $(K, N_x)$ and $(K, N_y)$ respectively (Fig. 5).

The algorithm is as following:

- **Binary each image** $w$ with offset $z_{ij}$ equal to a threshold $d$ (in our experiments $z=0.31$) $(TEM)$
- **Normalize the Output-I with column’s height** $(N_y)$ $(TEM)$
- **Extract the vertical histogram** $(TEM)$
- **Store the sum of column’s cells in the last cell and store zero elsewhere** $(TEM)$
- **Shift one-pixel to the top** $(TEM)$
- **Project the $N_x-1$ row of Output-5 in a pre-vertical path of size $(N_x, N_y)$ $(TEM)$
- **Shift one-pixel to the top the pre-vertical path** $(TEM)$

After applying the previous seven-steps to each image $w$ $1 \leq w \leq K$ we restrict the **pre-vertical path** $(K, N_y)$ to the final output $(K, N_y)$ vertical path using $(TEM)$ $(suppose K<N_y)$.

The same process done for vertical-path extraction is done for horizontal-path with slight genes’ adjustment.

Given a 3D point $A$, its correspondent image in the binary vertically-rectified image $s$ is referred by Cell $C_{ij}$. The correspondent $C_{ij}$ of $C_{ij}$ into $image-s$ is tracked by following the edge lines of horizontal path for $i'$, and vertical path for $j'$.

Since all the stream is vertically rectified, the shape at the vertical and horizontal histogram level are preserved from a picture to the next with slight difference caused by the translation and angle of projection.
Fig. 6 A CNN analogic algorithm for vertical path extraction (a), Simulation of each step of the algorithm with Input image $U$, Initial state $X$, and Output image $Y$ (b).
IV. SIMULATIONS

In our Experiments the 3DOF inertial gyro sensor and 3DOF accelerometer (TAGAWA TA 7511N3XX0) are attached to a CCD digital camera (SONY 3CCD DCR-VX1000) to continually report the ground-truth absolute pose of the sensor/camera. By back-projecting the 3D orientation changes reported by the gyroscopes we vertically-rectify the observed feature in the image planes.

After extracting the vertical and horizontal paths we match 11 tokens then estimate their range [10]. We conducted experiment test for the proposed matching approach, and the effectiveness of second part of this paper: Vertical-rectification.

Fig. 7 represents the stream of 13 frames with their vertical and horizontal path. From 1st frame to 13th frame the camera performed two translations, \( l = (-0.02, 12.01, 0.00) \) is the covered distance between \( t_1 \) and \( t_{13} \). Because of the movement, the gyroscopes registered some rotations around their axis that we took in consideration in the analyses. The image-1 and image-13’s orientations respectively are: \((\theta_1, \psi_1, \phi_1) = (-0.05, 0.05, 0.44)\) and \((\theta_{13}, \psi_{13}, \phi_{13}) = (-0.07, 0.10, 0.68)\).

Table. I Range estimation for 11 points from a static scene

<table>
<thead>
<tr>
<th>Pt(i,j)</th>
<th>( L_e ) (cm)</th>
<th>( L_m ) (cm)</th>
<th>Error (cm)</th>
<th>Error (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>139.24</td>
<td>140.60</td>
<td>-1.36</td>
<td>-0.96</td>
</tr>
<tr>
<td>B</td>
<td>145.92</td>
<td>142.80</td>
<td>3.12</td>
<td>2.18</td>
</tr>
<tr>
<td>C</td>
<td>137.95</td>
<td>139.80</td>
<td>-1.85</td>
<td>-1.33</td>
</tr>
<tr>
<td>D</td>
<td>142.64</td>
<td>142.40</td>
<td>0.24</td>
<td>0.17</td>
</tr>
<tr>
<td>E</td>
<td>135.11</td>
<td>134.80</td>
<td>0.31</td>
<td>0.23</td>
</tr>
<tr>
<td>F</td>
<td>138.78</td>
<td>136.80</td>
<td>1.98</td>
<td>1.45</td>
</tr>
<tr>
<td>G</td>
<td>153.72</td>
<td>149.00</td>
<td>4.72</td>
<td>3.17</td>
</tr>
<tr>
<td>H</td>
<td>137.58</td>
<td>136.60</td>
<td>0.98</td>
<td>0.72</td>
</tr>
<tr>
<td>I</td>
<td>139.32</td>
<td>138.40</td>
<td>0.92</td>
<td>0.66</td>
</tr>
<tr>
<td>J</td>
<td>158.13</td>
<td>152.00</td>
<td>6.13</td>
<td>4.03</td>
</tr>
<tr>
<td>K</td>
<td>163.36</td>
<td>155.40</td>
<td>7.96</td>
<td>5.12</td>
</tr>
</tbody>
</table>

Tracking 11 points, from 1st frame to 13th frame, and estimating their range we derived Table. I that shows the points from A to K in the 1st frame, the estimated distance \( L_e \) (using our algorithm), the measured distance \( L_m \) (manually measured), the difference between \( L_e \) and \( L_m \) in (cm), and the error in %. The estimated distance \( L_e \) is plotted with distance \( L_m \) in Fig. 8.

The quantitative errors in Table. I are appropriate since the changes in the image-space distances are proportional to the errors accumulated by the inertial system [5]. Beside the qualitatively and quantitatively good result, the recognition of appeared/disappeared objects from the vertical and horizontal paths are clear in Fig. 7.

To demonstrate the necessity of vertical-rectification as an intermediate process in our method in the case of the existence of rotation, we extracted the vertical and horizontal path directly from the stream of images, without rectifying them vertically. Vertical (and horizontal) path in Fig. 9 proves the almost-impossibility of using our matching approach without passing through the rectification constraint.
V. CONCLUSION

Using CNN tracking method, with a combination of inertial and visual tracking technologies, we have presented an analogic matching algorithm from a static-scene’s stream of images.

Referring to vertical and horizontal paths reviles the translation history of the sensors and simplifies the matching process. In the case of rotation presence in the movement of the sensors we have shown how the use of vertical rectification is crucial.

Furthermore, based on this approach, moving target detection seems to be a promising subject for future research.

REFERENCES