Dense, Auto-Calibrating Visual Odometry from a Downward-Looking Camera

Jacek Zienkiewicz, Robert Lukierski, Andrew Davison

Department of Computing
Imperial College London

September 12, 2013
Why yet another Visual Odometry system?

- Most generic VO systems use feature tracking and 3D reconstruction from mono or stereo rigs; e.g. just one example from our lab (Strasdat et al., ICCV 2011):

- Requires suitable texture and lighting for feature correspondences, and complex multi-stage algorithms.

- In low-cost robotics, can we estimate motion in a way which is simpler but takes advantage of prior knowledge to achieve accurate and robust results in a difficult setting?
Visual Odometry using Dense Planar Alignment


- Rear parking camera looks down at planar road texture during normal driving.
- Feature-less whole image alignment of video frames.
- Accurate real-time incremental visual motion estimates which can be fused with consumer GPS for a low-cost positioning solution.
Here we consider visual navigation for a small, low-cost robot from a single forward-looking camera.

The camera will most often observe the poorly-textured but nearly planar surface the robot is driving on.

As in the road vehicle case, we show that the strong planar prior enables robust and precise dense VO on a variety of surfaces (carpet, hard floor, wood, etc.).

We show that the camera extrinsics can be accurately auto-calibrated against the robot frame of reference.
Plane-induced Homography

Two images of the same plane are related by a homography.

\[ H = KT^{lr}(I | - n_{dc})^\top K^{-1} \]

Here \( T^{lr} \) is the relative motion of the camera between the two images.
Plane-induced Homography

Two images of the same plane are related by a homography.

\[ H = KT^{lr}(I | - n_{dc})^\top K^{-1} \]

Here \( T^{lr} \) is the relative motion of the camera between the two images.
Plane-induced Homography

Two images of the same plane are related by a homography.

\[ H = K T^{lr} (I - n_{dc}) \top K^{-1} \]

Here \( T^{lr} \) is the relative motion of the camera between the two images.
Two images of the same plane are related by a homography.

\[ H = KT^{lr}(I| - \mathbf{n}_{dc})^\top K^{-1} \]

Relation between the camera motion and the robot motion

\[ T^{lr} = T^{cv} T^{vlv_r} T^{vc} \]
Two images of the same plane are related by a homography.

\[ H = K T^{CV} T^{vlvr} T^{vc} (I \mid - n_{dc})^T K^{-1} \]

Homography depends on robot motion \( T^{vlvr} \).

- only 3 degrees of freedom
Parametric warp depending on the robot motion.

\[ H(\mathbf{x}) = K T^{c v} T^{v_l v_r} (\mathbf{x}) T^{v_c} (I | - \mathbf{n}_{d_c})^T K^{-1} \]

\[ \mathbf{x} = (x, y, \theta) \]

Original image  \[ \mathcal{I} \]

Warped image  \[ \mathcal{I}(H(0, 0, 0)) \]
Parametric warp depending on the robot motion.

\[ H(x) = K T^c v T^v_l v_r (x) T^v c (I | - n_{dc})^T K^{-1} \]

\[ x = (x, y, \theta) \]

Original image

Warped image

\[ I(H(10mm, 0, 0)) \]
Parametric warp depending on the robot motion.

\[ H(x) = K_T^{cv} T_{vlvr}(x) T_{vc} (I \mid - n_d)^	op K^{-1} \]

\[ x = (x, y, \theta) \]

Original image \( I \)

Warped image \( I(H(20\text{mm}, 0, 0)) \)
Parametric warp depending on the robot motion.

\[ H(x) = KT^{cv}Tv^r(x)Tv^c(I | - n_{dc})^TK^{-1} \]

\[ x = (x, y, \theta) \]

Original image \( I \)

Warped image \( I(H(0, 0, 0)) \)
Parametric warp depending on the robot motion.

\[ H(x) = K_T^{cv} T_{lvr}(x) T_{vc}(I \mid - n_{dc})^T K^{-1} \]

\[ x = (x, y, \theta) \]

Original image \( I \)

Warped image \( I(H(0, 0, 1^\circ)) \)
Parametric warp

Parametric warp depending on the robot motion.

\[ H(\mathbf{x}) = K T^{c v \, T^{v_l \, v_r}}(\mathbf{x}) T^{v_c}(I | - n_{dc})^T K^{-1} \]

\[ \mathbf{x} = (x, y, \theta) \]
Cost function

\[ |\mathcal{I}' - \mathcal{I}'(H(0, 0, 0))| = |\mathcal{I}_{\text{err}}| \]
Cost function

Ref. image $|\mathcal{I}^r| - |\mathcal{I}^l(\mathbf{H}(10, 0, 0))| = |\mathcal{I}^{err}|$

$0 10 20 30 40 50 60$

$x$ [mm]
Cost function

\[ |I^r| - |I^l(H(20, 0, 0))| = |I^{err}| \]
Cost function

\[ |I^r| - |I^l(H(28, 0, 0))| = |I^{err}| \]
Cost function

\[ |I^r| - |I^l(H(30, 0, 0))| = |I^{err}| \]
Cost function

\[
\|I^r\| - \|I^l(H(32, 0, 0))\| = \|I^{err}\|
\]
Cost function

\[ |I^r - I^l(\mathbf{H}(40, 0, 0))| = |I^{err}| \]
Cost function

\[
|I^r - I^l(H(50, 0, 0))| = |I^{err}|
\]
Cost function

$$|I^r - I^l(H(60, 0, 0))| = |I^{err}|$$

Ref. image

Live image

Error image

\[ \text{x [mm]} \]

\[ \begin{array}{c}
0 \\
10 \\
20 \\
30 \\
40 \\
50 \\
60 \\
\end{array} \]
Cost function

\[ F(x) = \| \mathcal{I}^r(p) - \mathcal{I}^l(H^l(x)p) \|_2^2 \]

for all pixels \( p \) in the domain.
Minimisation

Cost function (Lovegrove et al. 2011)

\[ F(x) = \| I^r(p) - I^l(H^l(x)p) \|_2^2 \]

- Variant of a Lucas-Kanade (Lucas & Kanade, 1981)
- Efficient Second-order Minimisation, ESM (Malis 2004, Mei et al. 2008)
- Easy to parallelise
- Real-time performance on a laptop GPU
  - gray-scale, VGA (640x480)
  - up to 60 fps on a MacBook Pro
- Initialisation
Dense visual odometry
Extrinsic calibration: what is the pose of the camera relative to the robot?

In the transformation $T^{vc}$:
- Two sets of parameters
- Two different calibration strategies

2 parameters
- $\alpha$ and $\beta$ angles (roll and pitch) define a plane

4 parameters
- $x_c$, $y_c$, $z_c$ and $\gamma$ angle (yaw) define a robot trajectory
Roll angle $\alpha$

Rotation with roll angle $\alpha$ does change plane normal.
Roll angle $\alpha$

Rotation with roll angle $\alpha$ does change plane normal.
Roll angle $\alpha$

Rotation with roll angle $\alpha$ does change plane normal
Roll angle $\alpha$

Rotation with roll angle $\alpha$ does change plane normal
Roll angle $\alpha$

Rotation with roll angle $\alpha$ does change plane normal
Roll angle $\alpha$

Rotation with roll angle $\alpha$ does change plane normal

Depth

Image
Pitch angle $\beta$

Rotation with pitch angle $\beta$ does change plane normal.
Pitch angle $\beta$

Rotation with pitch angle $\beta$ does change plane normal

Depth

Image
Pitch angle $\beta$

Rotation with pitch angle $\beta$ does change plane normal
Pitch angle $\beta$

Rotation with pitch angle $\beta$ does change plane normal.
Pitch angle $\beta$

Rotation with pitch angle $\beta$ does change plane normal

Depth

Image
Pitch angle $\beta$

Rotation with pitch angle $\beta$ does change plane normal

Depth

Image
Yaw angle $\gamma$

Rotation with yaw angle $\gamma$ does **not** change plane normal

Depth

Image
Yaw angle $\gamma$

Rotation with yaw angle $\gamma$ does **not** change plane normal depth.
Yaw angle $\gamma$

Rotation with yaw angle $\gamma$ does not change plane normal depth.
Rotation with yaw angle $\gamma$ does not change plane normal depth.
Rotation with yaw angle $\gamma$ does **not** change plane normal. Depth Image
Yaw angle $\gamma$

Rotation with yaw angle $\gamma$ does **not** change plane normal.
Yaw angle $\gamma$

Rotation with yaw angle $\gamma$ does **not** change plane normal.

![Diagram showing plane and camera with yaw angle $\gamma$]

Depth

Image
Plane calibration

Strategy for plane normal calibration

- We can estimate the roll and pitch angles from the visual input only.
- Extend homography \( H^{lr} \) to depend on plane parameters.
- Track camera planar motion by fixing \( x_c = 0, y_c = 0, z_c = 1, \gamma = 0 \)
Calibration of the remaining degrees of freedom

Still need to find:
- camera position $x_c, y_c, z_c$ (height defines scale)
- yaw angle $\gamma$

Source of external reference required:
- e.g. wheel odometry in good condition
Calibration of the remaining degrees of freedom

- Incremental measurements (Brookshire and Teller, 2011)
  \[
  T_{c_i c_{i+1}} = T_{cv} T_{v_i v_{i+1}} (T_{cv})^{-1}
  \]
- Formed into a factor graph and solved using g2o.
Experiments

- Tested on different surfaces
Experiments

Tested with different camera heights

- 70 mm
- 130 mm
- 200 mm
System performance

- In general graceful degradation
System performance

- Example of a tracking failure

Reference
Live
Live (wrapped)
Residuals
High precision motion estimation

- Camera height: 4 cm
- Frame to frame motion: 8 pixels
High precision motion estimation

- Camera height: 4 cm
- Frame to frame motion: 8 pixels
High precision motion estimation

- Camera height: 4 cm
- Frame to frame motion: 8 pixels
High precision motion estimation

- Camera height: 4 cm
- Frame to frame motion: 8 pixels
High precision motion estimation

- Camera height: 4 cm
- Frame to frame motion: 8 pixels
High precision motion estimation

- Camera height: 4 cm
- Frame to frame motion: 8 pixels
High precision motion estimation

- Camera height: 4 cm
- Frame to frame motion: 8 pixels
High precision motion estimation

- Camera height: 4 cm
- Frame to frame motion: 8 pixels
High precision motion estimation

- Camera height: 4 cm
- Frame to frame motion: 8 pixels
High precision motion estimation

- Camera height: 4 cm
- Frame to frame motion: 8 pixels
Conclusions

Contributions
- Dense alignment approach as a solution for a robust and accurate visual odometry
- Highly practical thanks to rapid auto-calibration

Issues and limitations
- Non-planar motion, non-planar structure (robust cost function)
- Shadows, motion blur

Future works
- Segmentation of the non-planer regions
- Moving towards dense reconstruction
- Auto-calibration using nonholonomic constraints
Thank you for your attention.