Sparse Tracking? and Dense Mapping

Liangzu Peng*, Xiaocheng Song*[1]
School of Information Science and Technology
ShanghaiTech University
penglz@shanghaitech.edu.cn

May 16, 2019

[1] equal contribution, names listed in alphabetic order.

Contents

- A short summary
  - of what we have done
- Dense mapping in DTAM [1]
  - with an emphasis on the regularization part
- Experimental results and **justifications**
  - figures
  - videos demos

What we have done

- An exhaustive study on the dense mapping algorithm [1].
  - found several typos in [1]
  - wrote a 5-page report (without figures)

What we have done

- An exhaustive study on the dense mapping algorithm [1].
  - found several typos in [1]
  - wrote a 5-page report (without figures)
- A CPU implementation
  - sparse tracking?
  - dense mapping

Figure 1: Our development activity on Github: 52 commits, 5 branches, and 2 contributors.

What we have done

- An exhaustive study on the dense mapping algorithm [1].
  - found several typos in [1]
  - wrote a 5-page report (without figures)
- A CPU implementation
  - ~~sparse traking~~
  - dense mapping with groundtruth pose

Figure 1: Our development activity on Github: 52 commits, 5 branches, and 2 contributors.

What we have done

- An exhaustive study on the dense mapping algorithm [1].
  - found several typos in [1]
  - wrote a 5-page report (without figures)
- A CPU implementation
  - ~~sparse traking~~
  - dense mapping with groundtruth pose
  - slow but works
    - 200 frames without regularization: 10min
    - 200 frames with regularization: 100min

Figure 1: Our development activity on Github: 52 commits, 5 branches, and 2 contributors.

The regularization algorithm for dense mapping

the energy function for dense mapping

$$E_{\boldsymbol{\xi}_r,\boldsymbol{\alpha}} = \int_{\Omega} \{\lambda \boldsymbol{C}(\boldsymbol{u}, \boldsymbol{\alpha}(\boldsymbol{u})) \tag{1}$$

$$\underline{+\tfrac{1}{2\theta}(\boldsymbol{\alpha}(\boldsymbol{u}) - \boldsymbol{\xi}_r(\boldsymbol{u}))^2 + g_r(\boldsymbol{u}) \left\|\nabla \boldsymbol{\xi}_r(\boldsymbol{u})\right\|_{\epsilon}}\}\mathrm{d}\boldsymbol{u}, \tag{2}$$

where $\boldsymbol{\alpha}, \boldsymbol{\xi}_r \in \mathbb{R}^{m\times n}$ are the inverse depth maps, and other symbols are given or known.

The regularization algorithm for dense mapping

the energy function for dense mapping

$$E_{\boldsymbol{\xi}_r,\boldsymbol{\alpha}} = \int_\Omega \{\lambda \boldsymbol{C}(\boldsymbol{u}, \boldsymbol{\alpha}(\boldsymbol{u})) \tag{1}$$

$$+\frac{1}{2\theta}(\boldsymbol{\alpha}(\boldsymbol{u}) - \boldsymbol{\xi}_r(\boldsymbol{u}))^2 + g_r(\boldsymbol{u}) \left\| \nabla \boldsymbol{\xi}_r(\boldsymbol{u}) \right\|_\epsilon \} \mathrm{d}\boldsymbol{u}, \tag{2}$$

where $\boldsymbol{\alpha}, \boldsymbol{\xi}_r \in \mathbb{R}^{m \times n}$ are the inverse depth maps, and other symbols are given or known.

- red: the non-convex data term in variable $\boldsymbol{\alpha}$
- blue: the convex regularization term in variable $\boldsymbol{\xi}_r$

The regularization algorithm for dense mapping

the energy function for dense mapping

$$E_{\boldsymbol{\xi}_r,\boldsymbol{\alpha}} = \int_{\Omega} \{\lambda \boldsymbol{C}(\boldsymbol{u}, \boldsymbol{\alpha}(\boldsymbol{u})) \tag{1}$$

$$\underline{+\tfrac{1}{2\theta}(\boldsymbol{\alpha}(\boldsymbol{u}) - \boldsymbol{\xi}_r(\boldsymbol{u}))^2 + g_r(\boldsymbol{u}) \left\|\nabla \boldsymbol{\xi}_r(\boldsymbol{u})\right\|_{\epsilon}}\}\mathrm{d}\boldsymbol{u}, \tag{2}$$

where $\boldsymbol{\alpha}, \boldsymbol{\xi}_r \in \mathbb{R}^{m\times n}$ are the inverse depth maps, and other symbols are given or known.

- red: the non-convex data term in variable $\boldsymbol{\alpha}$
- blue: the convex regularization term in variable $\boldsymbol{\xi}_r$
- alternatively update $\boldsymbol{\alpha}$ (with $\boldsymbol{\xi}_r$ fixed) and $\boldsymbol{\xi}$ (with $\boldsymbol{\alpha}_r$ fixed)
  - update $\boldsymbol{\alpha}$: brute-force search
  - update $\boldsymbol{\xi}_r$: the regularization algorithm

The regularization algorithm for dense mapping

the regularization energy:

$$E_{\boldsymbol{\xi}_r} = \int_\Omega \{ \frac{1}{2\theta} (\boldsymbol{\alpha}(\boldsymbol{u}) - \boldsymbol{\xi}_r(\boldsymbol{u}))^2 + g_r(\boldsymbol{u}) \left\| \nabla \boldsymbol{\xi}_r(\boldsymbol{u}) \right\|_\epsilon \} \mathrm{d}\boldsymbol{u}. \qquad (3)$$

rewrite (3) into its discrete form (i.e., in our familiar language):

$$E^1_{\boldsymbol{d}_r} = \frac{1}{2\theta} \left\| \boldsymbol{a} - \boldsymbol{d}_r \right\|_2^2 + \left\| \boldsymbol{A} \boldsymbol{G}_r \boldsymbol{d}_r \right\|_\epsilon , \qquad (4)$$

where

- $\boldsymbol{a} = \mathrm{vec}(\boldsymbol{\alpha}), \boldsymbol{d}_r = \mathrm{vec}(\boldsymbol{\xi}_r) \in \mathbb{R}^{mn}$,
- $\boldsymbol{G}_r = \mathrm{diag}(g_r(\boldsymbol{u})) \in \mathbb{R}^{mn \times mn}$, and
- $\boldsymbol{A}$ is such that $\boldsymbol{A}\boldsymbol{d}_r \in \mathbb{R}^{2mn}$ is the gradient vector[2].

[2] see our report for details.

The regularization algorithm for dense mapping

the discrete energy:

$$E_{\boldsymbol{d}_r} = \frac{1}{2\theta} \left\| \boldsymbol{a} - \boldsymbol{d}_r \right\|_2^2 + \left\| \boldsymbol{A}\boldsymbol{G}_r\boldsymbol{d}_r \right\|_\epsilon . \tag{5}$$

The regularization algorithm for dense mapping

the discrete energy:

$$E_{\boldsymbol{d}_r} = \frac{1}{2\theta} \left\| \boldsymbol{a} - \boldsymbol{d}_r \right\|_2^2 + \left\| \boldsymbol{A} \boldsymbol{G}_r \boldsymbol{d}_r \right\|_\epsilon . \tag{5}$$

image denoising

- bears a formulation similar to (5):

$$\min_I \frac{\lambda}{2} \left\| I - O \right\|_2^2 + \left\| \nabla I \right\|_1 , \tag{6}$$

  where O is the observed image.

The regularization algorithm for dense mapping

the discrete energy:

$$E_{\boldsymbol{d}_r} = \frac{1}{2\theta} \left\| \boldsymbol{a} - \boldsymbol{d}_r \right\|_2^2 + \left\| \boldsymbol{A} \boldsymbol{G}_r \boldsymbol{d}_r \right\|_\epsilon . \tag{5}$$

image denoising

- bears a formulation similar to (5):

$$\min_I \frac{\lambda}{2} \left\| I - O \right\|_2^2 + \left\| \nabla I \right\|_1 , \tag{6}$$

  where O is the observed image.

an abstraction of (5) and (6):

$$E_{\boldsymbol{x}} = G(\boldsymbol{x}) + F(\boldsymbol{K}\boldsymbol{x}), \tag{7}$$

- where G, F are simple in the sense that their proximal operator [2] can be efficiently computed [3].
- We will review the algorithm for solving (7).

The regularization algorithm for dense mapping

Minimizing abstract energy

$$E_{\boldsymbol{x}} = G(\boldsymbol{x}) + F(\boldsymbol{K}\boldsymbol{x}). \tag{8}$$

is equivalent to the following problem

$$\min_{\boldsymbol{x},\boldsymbol{z}} \{G(\boldsymbol{x}) + F(\boldsymbol{z})\} \tag{9}$$

$$\text{subject to } \boldsymbol{z} = \boldsymbol{K}\boldsymbol{x}.$$

The regularization algorithm for dense mapping

Minimizing abstract energy

$$E_{\boldsymbol{x}} = G(\boldsymbol{x}) + F(\boldsymbol{K}\boldsymbol{x}). \tag{8}$$

is equivalent to the following problem

$$\min_{\boldsymbol{x},\boldsymbol{z}} \{G(\boldsymbol{x}) + F(\boldsymbol{z})\} \tag{9}$$

$$\text{subject to } \boldsymbol{z} = \boldsymbol{K}\boldsymbol{x}.$$

The lagrangian of (9) is given by

$$\mathcal{L}(\boldsymbol{x}, \boldsymbol{z}; \boldsymbol{y}) = G(\boldsymbol{x}) + F(\boldsymbol{z}) + \boldsymbol{y}^{\top}(\boldsymbol{K}\boldsymbol{x} - \boldsymbol{z}). \tag{10}$$

The regularization algorithm for dense mapping

Minimizing abstract energy

$$E_{\boldsymbol{x}} = G(\boldsymbol{x}) + F(\boldsymbol{K}\boldsymbol{x}). \tag{8}$$

is equivalent to the following problem

$$\min_{\boldsymbol{x},\boldsymbol{z}} \{G(\boldsymbol{x}) + F(\boldsymbol{z})\} \tag{9}$$

$$\text{subject to } \boldsymbol{z} = \boldsymbol{K}\boldsymbol{x}.$$

The lagrangian of (9) is given by

$$\mathcal{L}(\boldsymbol{x}, \boldsymbol{z}; \boldsymbol{y}) = G(\boldsymbol{x}) + F(\boldsymbol{z}) + \boldsymbol{y}^\top(\boldsymbol{K}\boldsymbol{x} - \boldsymbol{z}). \tag{10}$$

Hence minimizing (8) or (9) is equivalent to

$$\max_{\boldsymbol{y}} \{\min_{\boldsymbol{x},\boldsymbol{z}} \mathcal{L}(\boldsymbol{x}, \boldsymbol{z}; \boldsymbol{y})\} \tag{11}$$

$$\iff \max_{\boldsymbol{y}} \{\min_{\boldsymbol{x},\boldsymbol{z}} \{G(\boldsymbol{x}) + F(\boldsymbol{z}) + \boldsymbol{y}^\top(\boldsymbol{K}\boldsymbol{x} - \boldsymbol{z})\}\} \tag{12}$$

The regularization algorithm for dense mapping

The equivalent problem:

$$
\begin{aligned}
& \max_{\boldsymbol{y}}\{\min_{\boldsymbol{x},\boldsymbol{z}} \mathcal{L}(\boldsymbol{x},\boldsymbol{z};\boldsymbol{y})\} \\
\iff & \max_{\boldsymbol{y}}\{\min_{\boldsymbol{x},\boldsymbol{z}}\{G(\boldsymbol{x}) + F(\boldsymbol{z}) + \boldsymbol{y}^\top(\boldsymbol{K}\boldsymbol{x} - \boldsymbol{z})\}\} \\
\iff & \max_{\boldsymbol{y}}\{\min_{\boldsymbol{x}}\{G(\boldsymbol{x}) + \boldsymbol{y}^\top \boldsymbol{K}\boldsymbol{x}\} + \min_{\boldsymbol{z}}\{-(\boldsymbol{y}^\top \boldsymbol{z} - F(\boldsymbol{z}))\}\} \\
\iff & \max_{\boldsymbol{y}}\{\min_{\boldsymbol{x}}\{G(\boldsymbol{x}) + \boldsymbol{y}^\top \boldsymbol{K}\boldsymbol{x}\} - F^*(\boldsymbol{y})\} \\
\iff & \max_{\boldsymbol{y}} \min_{\boldsymbol{x}}\{G(\boldsymbol{x}) + \boldsymbol{y}^\top \boldsymbol{K}\boldsymbol{x} - F^*(\boldsymbol{y})\},
\end{aligned} \tag{13}
$$

- where F^* is the Fenchel conjugate F^* (definition: $F^*(\boldsymbol{y}) = \max_{\boldsymbol{z}}\{\boldsymbol{y}^\top \boldsymbol{z} - F(\boldsymbol{z})\}$).

The regularization algorithm for dense mapping

The equivalent problem:

$$
\begin{aligned}
& \max_{\boldsymbol{y}}\{\min_{\boldsymbol{x},\boldsymbol{z}} \mathcal{L}(\boldsymbol{x},\boldsymbol{z};\boldsymbol{y})\} \\
\Longleftrightarrow\ & \max_{\boldsymbol{y}}\{\min_{\boldsymbol{x},\boldsymbol{z}}\{G(\boldsymbol{x}) + F(\boldsymbol{z}) + \boldsymbol{y}^\top(\boldsymbol{K}\boldsymbol{x} - \boldsymbol{z})\}\} \\
\Longleftrightarrow\ & \max_{\boldsymbol{y}}\{\min_{\boldsymbol{x}}\{G(\boldsymbol{x}) + \boldsymbol{y}^\top\boldsymbol{K}\boldsymbol{x}\} + \min_{\boldsymbol{z}}\{-(\boldsymbol{y}^\top\boldsymbol{z} - F(\boldsymbol{z}))\}\} \\
\Longleftrightarrow\ & \max_{\boldsymbol{y}}\{\min_{\boldsymbol{x}}\{G(\boldsymbol{x}) + \boldsymbol{y}^\top\boldsymbol{K}\boldsymbol{x}\} - F^*(\boldsymbol{y})\} \\
\Longleftrightarrow\ & \max_{\boldsymbol{y}}\min_{\boldsymbol{x}}\{G(\boldsymbol{x}) + \boldsymbol{y}^\top\boldsymbol{K}\boldsymbol{x} - F^*(\boldsymbol{y})\}, \qquad (13)
\end{aligned}
$$

- where F^* is the Fenchel conjugate F^* (definition: $F^*(\boldsymbol{y}) = \max_{\boldsymbol{z}}\{\boldsymbol{y}^\top\boldsymbol{z} - F(\boldsymbol{z})\}$).
- Problem (13) is known as the prime-dual or saddle-point formulation of our original problem.
  - prime variable $\boldsymbol{x}$, dual variable $\boldsymbol{y}$.

The regularization algorithm for dense mapping

The prime-dual problem:

$$\max_{\boldsymbol{y}} \min_{\boldsymbol{x}} \{G(\boldsymbol{x}) + \boldsymbol{y}^\top \boldsymbol{K}\boldsymbol{x} - F^*(\boldsymbol{y})\}. \tag{14}$$

The algorithm proposed in [3] for solving (14):

1. initialize $\boldsymbol{x}^0, \boldsymbol{y}^0, \tau > 0, \sigma > 0$.
2. $\boldsymbol{y}^{n+1} = \mathrm{prox}_{\mathrm{F}^*}(\boldsymbol{y}^n + \sigma \boldsymbol{K}\boldsymbol{x}^n)$.
3. $\boldsymbol{x}^{n+1} = \mathrm{prox}_{\mathrm{G}}(\boldsymbol{x}^n - \tau \boldsymbol{K}^\top \boldsymbol{y}^{n+1})$.
4. repeat steps 2 and 3 until convergence.

The regularization algorithm for dense mapping

The prime-dual problem:

$$\max_{\boldsymbol{y}} \min_{\boldsymbol{x}} \{G(\boldsymbol{x}) + \boldsymbol{y}^\top \boldsymbol{K}\boldsymbol{x} - F^*(\boldsymbol{y})\}. \tag{14}$$

The algorithm proposed in [3] for solving (14):

1. initialize $\boldsymbol{x}^0, \boldsymbol{y}^0, \tau > 0, \sigma > 0$.
2. $\boldsymbol{y}^{n+1} = \mathrm{prox}_{\mathrm{F}^*}(\boldsymbol{y}^n + \sigma \boldsymbol{K}\boldsymbol{x}^n)$.
3. $\boldsymbol{x}^{n+1} = \mathrm{prox}_{\mathrm{G}}(\boldsymbol{x}^n - \tau \boldsymbol{K}^\top \boldsymbol{y}^{n+1})$.
4. repeat steps 2 and 3 until convergence.

Remark.

- $\mathrm{prox}_h(x)$: the proximal operator of the function h at x.
- convergence guaranteed when $\tau\sigma \|\boldsymbol{K}\|_2^2 < 1$.
- no assumption on the differentiability of G and F^*.
- we will adopt this algorithm for regularization as in [1].

The regularization algorithm for dense mapping

go back to our regularization energy:

$$E_{\boldsymbol{d}_r} = \frac{1}{2\theta} \left\| \boldsymbol{a} - \boldsymbol{d}_r \right\|_2^2 + \left\| \boldsymbol{A} \boldsymbol{G}_r \boldsymbol{d}_r \right\|_\epsilon . \tag{15}$$

The prime-dual representation of (15) is given[3] by

$$\max_{\boldsymbol{q}} \min_{\boldsymbol{d}_r} \{ E_{\boldsymbol{q},\boldsymbol{d}_r} \}, \text{ where} \tag{16}$$

$$E_{\boldsymbol{q},\boldsymbol{d}_r} = \frac{1}{2\theta} \left\| \boldsymbol{a} - \boldsymbol{d}_r \right\|_2^2 + (\boldsymbol{B}_r \boldsymbol{d}_r)^\top \boldsymbol{q} - \delta_q(\boldsymbol{q}) - \frac{\epsilon}{2} \left\| \boldsymbol{q} \right\|_2^2 . \tag{17}$$

- δ is the indicator function on q such that

$$\delta_q(\boldsymbol{q}) = \begin{cases} \epsilon/2, & \text{if } \boldsymbol{q} \in q, \\ \infty, & \text{otherwise.} \end{cases} \tag{18}$$

- $E_{\boldsymbol{q},\boldsymbol{d}_r}$ differentiable $\Rightarrow$ gradient descent/ascend on $\boldsymbol{d}_r / \boldsymbol{q}$.

---

[3] see our report for details.

The regularization algorithm for dense mapping

The prime-dual representation of the regularization energy:

$$\max_{\boldsymbol{q}} \min_{\boldsymbol{d}_r} \{E_{\boldsymbol{q},\boldsymbol{d}_r}\}, \text{ where} \tag{19}$$

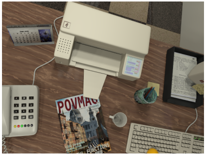
$$E_{\boldsymbol{q},\boldsymbol{d}_r} = \frac{1}{2\theta} \|\boldsymbol{a} - \boldsymbol{d}_r\|_2^2 + (\boldsymbol{B}_r \boldsymbol{d}_r)^\top \boldsymbol{q} - \delta_q(\boldsymbol{q}) - \frac{\epsilon}{2} \|\boldsymbol{q}\|_2^2 . \tag{20}$$

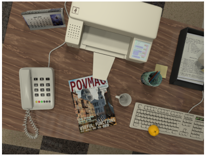
Algorithm in [1]:

1. Notice $\frac{\partial E_{\boldsymbol{q},\boldsymbol{d}_r}}{\partial \boldsymbol{q}} = \boldsymbol{B}_r \boldsymbol{d}_r - \epsilon \boldsymbol{q}$ and $\frac{\partial E_{\boldsymbol{q},\boldsymbol{d}_r}}{\partial \boldsymbol{d}_r} = \boldsymbol{B}_r^\top \boldsymbol{q} + \frac{1}{\theta}(\boldsymbol{d}_r - \boldsymbol{a})$.
2. initialize $\boldsymbol{d}_r^0 = \boldsymbol{a}, \boldsymbol{q}^0 = \boldsymbol{0}, \sigma_{\boldsymbol{d}_r} > 0, \sigma_{\boldsymbol{q}} > 0$.
3. compute $\boldsymbol{q}^{n+1}$ such that
$$\frac{\boldsymbol{q}^{n+1} - \boldsymbol{q}^n}{\sigma_{\boldsymbol{q}}} = \boldsymbol{B}_r \boldsymbol{d}_r^n - \epsilon \boldsymbol{q}^{n+1} \text{ (gradient ascend)}. \tag{21}$$
4. project $\boldsymbol{q}^{n+1}$ onto the set q via $\Pi_{\boldsymbol{q}}(x) = x / \max(1, \|x\|_2)$.
5. compute $\boldsymbol{d}^{n+1}$ such that
$$\frac{\boldsymbol{d}_r^{n+1} - \boldsymbol{d}_r^n}{\sigma_{\boldsymbol{d}_r}} = -\boldsymbol{B}_r^\top \boldsymbol{q}^{n+1} - \frac{1}{\theta}(\boldsymbol{d}_r^{n+1} - \boldsymbol{a}) \text{ (gradient descent)}. \tag{22}$$
6. repeat steps 3-5 until convergence.

Experiments

Figure 2: the reference frame





Dataset.

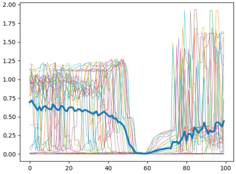
- taken from the author's website [4][a].
- synthetic dataset
- brightness constancy
- narrow-baseline frames
- but not designed for DTAM
  - hence challenging enough

[a] this link

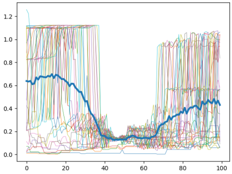
Experiments

Cost Volume.

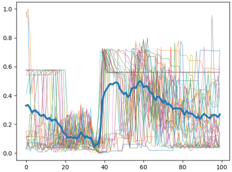
Figure 4: The reference image with 6 colored points on the phone (black), table (yellow), corner (purple), edge (green), printer (red), and wall (blue).



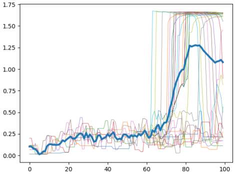
(a) Phone (black)



(b) Table (yellow)



(c) Corner (purple)



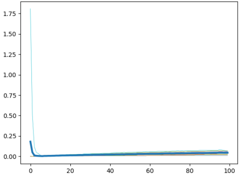
(d) Edge (green)

- in agreement with the results in the DTAM paper.

Experiments

Cost Volume.

Figure 6: The reference image with 6 colored points on the phone (black), table (yellow), corner (purple), edge (green), printer (red), and wall (blue)

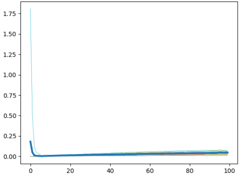


(a) Printer (red)

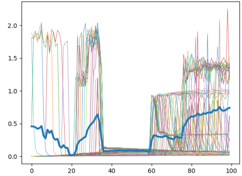
Experiments

Cost Volume.

Figure 6: The reference image with 6 colored points on the phone (black), table (yellow), corner (purple), edge (green), printer (red), and wall (blue)



(a) Printer (red), 10 frames

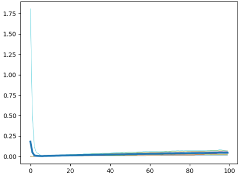


(b) Printer (red), 50 frames

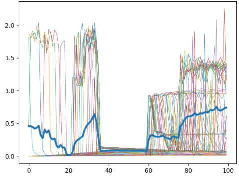
Experiments

Cost Volume.

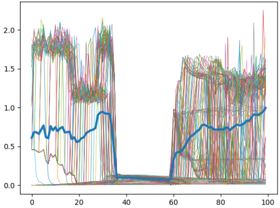
Figure 6: The reference image with 6 colored points on the phone (black), table (yellow), corner (purple), edge (green), printer (red), and wall (blue)



(a) Printer (red), 10 frames



(b) Printer (red), 50 frames



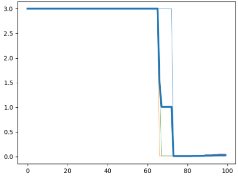
(c) Printer (red), 100 frames

- Improvements over time.

Experiments

Cost Volume.

Figure 8: The reference image with 6 colored points on the phone (black), table (yellow), corner (purple), edge (green), printer (red), and wall (blue)

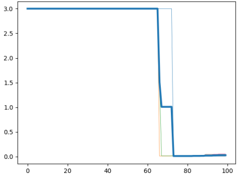


(a) Wall (blue), 10 frames

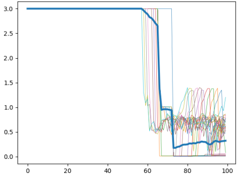
Experiments

Cost Volume.

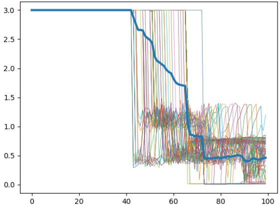
Figure 8: The reference image with 6 colored points on the phone (black), table (yellow), corner (purple), edge (green), printer (red), and wall (blue)



(a) Wall (blue), 10 frames



(b) Wall (blue), 50 frames



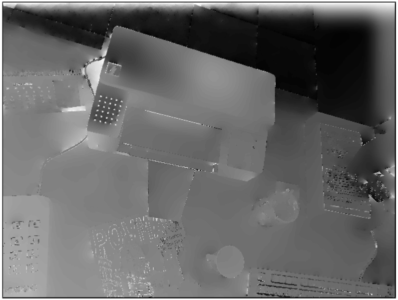
(c) Wall (blue), 100 frames

- Insufficient observations.

Experiments

Regularization.

(a) Before regularization



(b) After Regularization

- regularization adds smoothness
- smaller weight at edges $\Rightarrow$ weaker regularization

R. A. Newcombe, S. J. Lovegrove, and A. J. Davison.
Dtam: Dense tracking and mapping in real-time.
In 2011 International Conference on Computer Vision, pages 2320–2327, Nov 2011.

A. Beck.
First-Order Methods in Optimization.
Society for Industrial and Applied Mathematics, Philadelphia, PA, 2017.

Antonin Chambolle and Thomas Pock.
A first-order primal-dual algorithm for convex problems with applications to imaging.
Journal of Mathematical Imaging and Vision, 40(1):120–145, May 2011.

A. Handa, R. A. Newcombe, A. Angeli, and A. J. Davison.
Real-time camera tracking: When is high frame-rate best?
In Proc. of the European Conference on Computer Vision(ECCV), Oct. 2012.