Chapter 8
Variational Multiview Reconstruction

Multiple View Geometry
Summer 2013

Prof. Daniel Cremers
Chair for Computer Vision and Pattern Recognition
Department of Computer Science
Technische Universität München

Overview

1. **Variational Multiview Reconstruction**

2. **Multiview Reconstruction for Internet Photocollections**

3. **Realtime Structure and Motion**

4. **Realtime Dense Geometry**

5. **Autonomous Quadrocopters**

Shape Optimization

Shape optimization is a field of mathematics that is focused on formulating the estimation of geometric structures by means of optimization methods.

Among the major challenges in this context is the question how to mathematically represent shape. The choice of representation entails a number of consequences, in particular regarding the question of how efficiently one can store geometric structures and how efficiently one can compute optimal geometry.

There exist numerous representations of shape which can loosely be grouped into two classes:

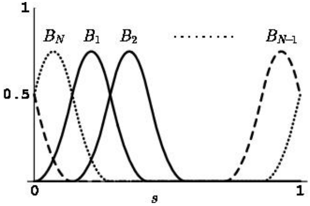
- Explicit representations: The points of a surface are represented explicitly (directly), either as a set of points, a polyhedron or a parameterized surface.
- Implicit representations: The surface is represented implicity by specifying the parts of ambient space that are inside and outside a given surface.

Explicit Shape Representations

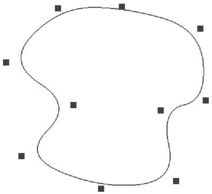
An explicit representations of a closed curve $C \subset \mathbb{R}^2$ is a mapping $C : \mathbb{S}^1 \to \mathbb{R}^2$ from the circle $\mathbb{S}^1$ to the plane $\mathbb{R}^2$. Examples are polygons or – more generally – spline curves:

$$C(s) = \sum_{i=1}^{N} p_i \, B_i(s),$$

where $p_1, \ldots, p_N \in \mathbb{R}^2$ denote control points and $B_1, \ldots, B_N : \mathbb{S}^1 \to \mathbb{R}^2$ denote a set of spline basis functions:



basis functions



spline & control points

Implicit Shape Representations

One example of an implicit representation is the indicator function of the surface S, which is a function $u : V \to \{0, 1\}$ defined on the surrounding volume $V \subset \mathbb{R}^3$ that takes on the values 1 inside the surface and 0 outside the surface:

$$u(x) = \begin{cases} 1, & \text{if } x \in \text{int}(S) \\ 0, & \text{if } x \in \text{ext}(S) \end{cases}$$

Another example is the signed distance function $\phi : V \to \mathbb{R}$ which assigns all points in the surrounding volume the (signed) distance from the surface S:

$$\phi(x) = \begin{cases} +d(x, S), & \text{if } x \in \text{int}(S) \\ -d(x, S), & \text{if } x \in \text{ext}(S) \end{cases}$$

Depending on the application it may be useful to know for every voxel how far it is from the surface. Signed distance functions can be computed in polynomial time. MatLab: bwdist.

Explicit Versus Implicit Representations

In general, compared to explicit rerpresentations the implicit representations have the following strengths and weaknesses:

- \- Implicit representations typically require more memory in order to represent a geometric structure at a specific resolution. Rather than storing a few points along the curve or surface, one needs to store an occupancy value for each volume element.
- \- Moving or updating an implicit representation is typically slower: rather than move a few control points, one needs to update the occupancy of all volume elements.
- \+ Methods based on implicit representations do not depend on a choice of parameterization.
- \+ Implicit representations allow to represent objects of arbitrary topology (i.e. the number of holes is arbitrary).
- \+ With respect to an implicit representation many shape optimization challenges can be formulated as convex optimization problems and can then be optimized globally.

Multiview Reconstruction as Shape Optimization

How can we cast multiple view reconstruction as a shape optimization problem? To this end, we will assume that the camera orientations are given.
Rather than estimate the correspondence between all pairs of pixels in either image we will simply ask:

How likely is a given voxel x on the object surface S?

If the voxel $x \in V$ of the given volume $V \subset \mathbb{R}^3$ was on the surface then (up to visibility issues) the projection of that voxel into each image should give rise to the same color (or local texture). Thus we can assign to each voxel $x \in V$ a so-called photoconsistency function

$$\rho : V \to [0, 1],$$

which takes on low values (near 0) if the projected voxels give rise to the same color (or local texture) and high values (near 1) otherwise.

A Weighted Minimal Surface Approach

The reconstruction from multiple views can now be formulated as a reconstruction of the maximally photoconsistent surface, i.e. a surface S_{opt} with an overall minimal photoconsistency score:

$$S_{opt} = \arg\min_{S} \int_{S} \rho(s) ds. \qquad (1)$$

This seminal formulation was proposed among others by Faugeras, Keriven (1998). Many good reconstructions from multiple views were computed using such formulations, typically by locally minimizing this energy starting from a meaningful initial guess of the surface S.

In principle, one could also revert to the implicit representation with the indicator function u introduced above and solve the problem optimally using convex relaxation and thresholding.

However, the above energy has a central drawback:

The global minimizer of (1) is the empty set.

It has zero cost while all surfaces have a non-negative energy.

Imposing Silhouette Consistency

Assume that we additionally have the silhouette S_i of the observed $3D$ object outlined in every image $i = 1, \ldots, n$. Then we can formulate the reconstruction problem as a constrained optimization problem (Cremers, Kolev, PAMI 2012):

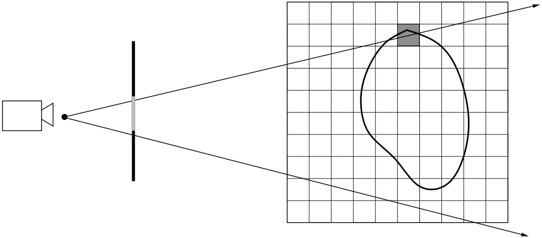
$$\min_S \int_S \rho(s) ds, \quad \text{such that } \pi_i(S) = S_i \ \forall i = 1, \ldots, n.$$

Written in the indicator function $u : V \to \{0, 1\}$ of the surface S this reads:

$$\begin{aligned} &\min_{u:V\to\{0,1\}} \int_V \rho(x)|\nabla u(x)| \, dx \\ &\text{s. t.} \quad \int_{R_{ij}} u(x) \, dR_{ij} \geq 1, \text{ if } j \in S_i \\ &\qquad\quad \int_{R_{ij}} u(x) \, dR_{ij} = 0, \text{ if } j \notin S_i, \end{aligned}$$

where R_{ij} denotes the visual ray through pixel j of image i.

Imposing Silhoutte Consistency



Top view of the geometry and respective visual rays.

Any ray passing through the silhoutte must intersect the object in at least one voxel.

Any ray passing outside the silhouette may not intersect the object in any pixel.

Convex Relaxation and Thresholding

By relaxing the binarity constraint on u and allowing intermediate values between 0 and 1 for the function u, the overall optimization problem becomes convex.

Proposition

The set

$$\mathcal{D} := \left\{ u : V \to [0,1] \;\middle|\; \begin{array}{ll} \int_{R_{ij}} u(x)\, dR_{ij} \geq 1 & \text{if } j \in S_i \;\forall i,j \\ \int_{R_{ij}} u(x)\, dR_{ij} = 0 & \text{if } j \notin S_i \;\forall i,j \end{array} \right\}$$

of silhouette consistent functions is convex.

Proof.

For a proof we refer to Kolev, Cremers, ECCV 2008. □

Thus we can compute solutions to the silhouette constrained reconstruction problem by solving the relaxed convex problem and subsequently thresholding the computed solution.

Reconstructing Complex Geometry

three out of 33 **input images of resolution** 1024×768

Reconstructing Complex Geometry

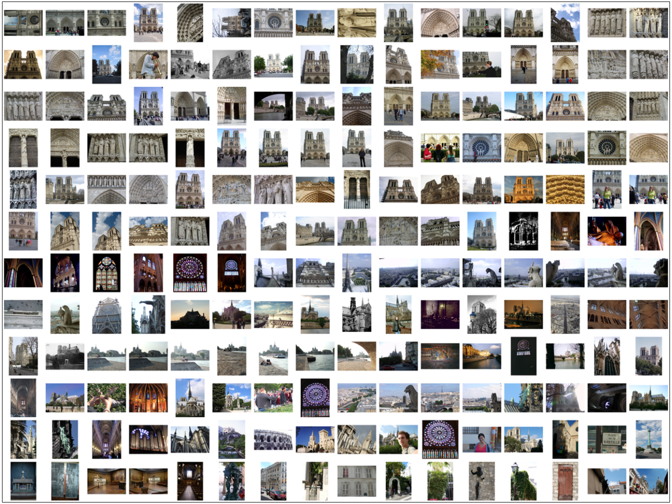
Estimated multiview reconstruction

Reconstruction from a Handheld Camera

2/28 images

Estimated multiview reconstruction

From Internet Photo Collections...



Flickr images for search term “Notre Dame”

Snavely, Seitz, Szeliski, “Modeling the world from Internet photo collections,” IJCV 2008.

...to Sparse Reconstructions

Snavely, Seitz, Szeliski, “Modeling the world from Internet photo collections,” IJCV 2008.

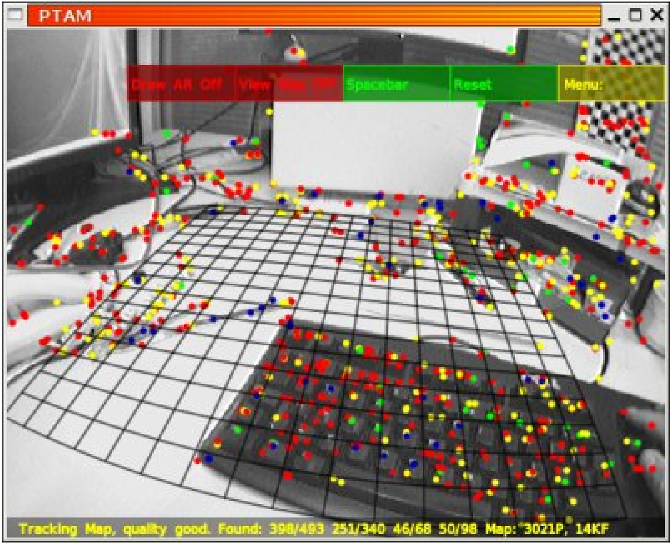
Rome in 3D

Author: N. Snavely

Dubrovnik in 3D

Author: N. Snavely

Realtime Structure and Motion: PTAM



Klein & Murray, "Parallel Tracking and Mapping for Small AR Workspaces," ISMAR 2007.

Realtime Structure and Motion: PTAM

Author: G. Klein

Structure and Motion for Virtual Reality

Author: G. Klein

Dense Geometry from a Handheld Camera

Let $g_i \in SE(3)$ be the rigid body motion from the first camera to the i-th camera, and let $I_i : \Omega \to \mathbb{R}$ be the i-th image. A dense depth map $h : \Omega \to \mathbb{R}$ can be computed by solving the optimization problem:

$$\min_h \sum_{i=2}^{n} \int_\Omega \left| I_1(x) - I_i\big(\pi g_i(hx)\big) \right| \, \mathrm{d}x \; + \; \lambda \int_\Omega |\nabla h| \, \mathrm{d}x,$$

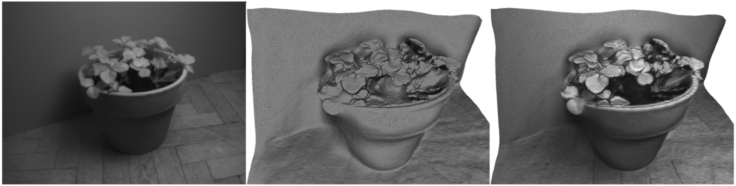
where x is represented in homogeneous coordinates and hx is the 3D point.

Like in optical flow estimation, the unknown depth map should be such that for all pixels $x \in \Omega$, the transformation into the other images I_i should give rise to the same color as in the reference image I_1.

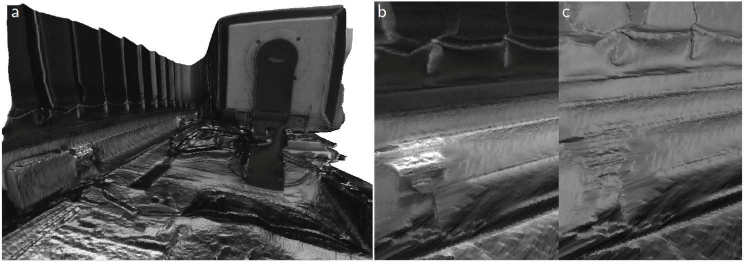
This cost function can be minimized in realtime by coarse-to-fine linearization solved in parallel on a GPU.

Stuehmer, Gumhold, Cremers, “Realtime Dense Geometry from a Handheld Camera,” DAGM 2010.

Dense Geometry from a Handheld Camera



(a) Image of current view (b) Reconstruction (c) Textured geometry



Stuehmer, Gumhold, Cremers, “Realtime Dense Geometry from a Handheld Camera,” DAGM 2010.

Dense Geometry from a Handheld Camera

Stuehmer, Gumhold, Cremers, “Realtime Dense Geometry from a Handheld Camera,” DAGM 2010.

Dense Geometry from a Handheld Camera

Stuehmer, Gumhold, Cremers, “Realtime Dense Geometry from a Handheld Camera,” DAGM 2010.

Visual Navigation of Autonomous Quadrocopters

Engel, Sturm, Cremers, “Camera-based Navigation of a Low-Cost Quadrocopter,” IROS 2012.