3D Computer Vision

Dense 3D Reconstruction

Prof. Didier Stricker
Kaiserslautern University
http://ags.cs.uni-kl.de/

DFKI – Deutsches Forschungszentrum für Künstliche Intelligenz
http://av.dfki.de
Outline

• Previous lecture:
  ▪ Structure and motion loop
  ▪ Triangulation

• Today: Dense Reconstruction
  ▪ Stereo reconstruction
  ▪ The matching problem
  ▪ Multi-view reconstruction
Why Dense Reconstruction?

- Accurate 3D models → cultural heritage!
- Reconstruction of houses, buildings, famous touristic sites…

Piazza San Marco – Venice
13.703 images
27.707.825 points
Dense Reconstruction - Examples

Piazza San Marco, Venice

Reference Photo

Colosseum
Outline

• **Previous lecture:**
  - Structure and motion loop
  - Triangulation

• **Today: Dense Reconstruction**
  - Stereo reconstruction
  - The matching problem
  - Multi-view reconstruction
Stereo Reconstruction

- **Input:** a pair of images (usually left and right)
- **Output:** (traditionally) a disparity or depth map

Stereo Reconstruction

Disparity map

Depth map

\[ \text{disparity} \propto \frac{1}{\text{depth}} \]

Essentially both disparity and depth maps convey the same information!
Stereo Reconstruction - Recall

How can we recover the depth?

Lecture on epipolar geometry…

Matching + Triangulation
Stereo Reconstruction – Image Rectification

- Idea: simplify the geometry of the “converging cameras” case
  - Rectifying one image
  - Rectifying both images

Image mapping is a 2D homography (projective transformation)

\[ H = KRK^{-1} \]

Project images onto plane parallel to baseline
Stereo Reconstruction – Image Rectification

- Define rotations of the two virtual views that that x-axes of camera coordinates system are parallel to the translation of the original camera positions
  - These rotations are distributed equally to the right and the left camera to reduce the resulting distortions
  - Additionally, the camera matrix representing the intrinsic parameters should be equal for both virtual views

- Assuming calibrated stereo cameras
- The intrinsic and extrinsic parameters are given
Image Rectification
– Determination of Image Transformation

• New camera matrix $K_n$ should be equal for both virtual views
  ▪ Compute from mean of old views
    \[
    K_n = \frac{(K_{left} + K_{right})}{2}
    \]

• New rotation matrix $R_n$ which represents $x, y, z$ – axes of virtual camera system w.r.t world coordinates system
  ▪ Virtual $x$-axis set parallel to image baseline $\rightarrow r_1$
  ▪ Virtual $y$-axis set orthogonal to virtual $x$-axis and
    ▪ orthogonal to old $z$-axis of left camera (3. column of rotation matrix), or
    ▪ orthogonal to the mean of old $z$-axes of two cameras $\rightarrow r_2$
  ▪ Virtual $z$-axis orthogonal to virtual $x$-axis (baseline) and virtual $y$-axis $\rightarrow r_3$
  \[
  r_1 = \frac{t_{left} - t_{right}}{||t_{left} - t_{right}||} \quad r_2 = 0.5(r_{3_{ol}} + r_{3_{or}}) \times r_1 \quad r_3 = r_1 \times r_2
  \]

http://people.scs.carleton.ca/~c_shu/Courses/comp4900d/notes/rectification.pdf
Image Rectification – Computation of Homography

- **Link transformations to homography** $H$
  - The aimed virtual images are rotated from original one, but not translated
    - Home position (left image): $x = K[I|0]X = KX$
  - Rotation by $R_n$
    $$x' = K[R|0]X = KRX$$
  - Gives:
    $$x' = KRK^{-1}x \quad H = KRK^{-1}$$
    - This transformation matrix $H$ maps original images to rectified images
    - Usually implemented as indirect transformation based $H^{-1}$
    - Right image is calculated in a similar way
      $$H^{-1}x' = x$$
Image Rectification – Direct and Indirect Geometric Transformation

• **Direct transformation**
  - For each pixel of the input image, the corresponding coordinates in the output image are computed.
  - These coordinates will usually have non-integer values. Also some pixels will have multiple or non assignments.
  - This requires additional computational effort during assignments of grey values.
  - Not recommended.

• **Indirect transformation**
  - The inverse transformation is used to calculate for each pixel of the output image the corresponding (non-integer) position of the input image.
  - The grey value of the surrounding pixels are then used for simple interpolation.
Stereo Reconstruction - Image Rectification
Stereo Reconstruction
– Recall: Triangulation with Parallel Cameras

• Triangulation reduces to a simple geometric problem

\[
K = K' = \begin{bmatrix}
    f & 0 & 0 \\
    0 & f & 0 \\
    0 & 0 & 1
\end{bmatrix} \quad R = I \quad t = \begin{pmatrix}
    t_x \\
    0 \\
    0
\end{pmatrix}
\]

- Then, \( y = y' \) and depth \( Z \) can be computed from disparity:

\[
Z = \frac{ft_x}{d}, \quad d = x' - x
\]
Stereo Reconstruction – Matching

Harris corners
Lines/edges
SIFT keypoints, etc.

Feature-based matching
Suitable for Structure and Motion

But we are interested in dense reconstruction

We need dense matching!

Match as many pixels as possible
Outline

• Previous lecture:
  - Structure and motion loop
  - Triangulation

• Today: Dense Reconstruction
  - Stereo reconstruction
  - The matching problem
  - Multi-view reconstruction
Stereo Reconstruction – Dense Matching

- For every point on an image, there are many possible matches on the other image(s)
- Many points look similar → high ambiguity

What can we do?
Stereo Reconstruction – Dense Matching

- We can use camera geometry
  → finding the correspondence becomes a 1D search problem

\[
\begin{align*}
I' &= Fx \\
F &= K'^{-T}t_xRK^{-1} \\
x'^T F x &= 0
\end{align*}
\]
Stereo Reconstruction – Dense Matching

• Even using the epipolar constraint, there are many possible matches

• Use of pixel neighborhood information → correlation-based approaches (also called window-based approaches)
Stereo Reconstruction – Window Based Matching

- Normalized Cross-Correlation (NCC)

\[
\frac{\sum_{(i,j) \in W} I_1(i, j) \cdot I_2(x + i, y + j)}{\sqrt{\sum_{(i,j) \in W} I_1^2(i, j) \cdot \sum_{(i,j) \in W} I_2^2(x + i, y + j)}}
\]

- Sum of Squared Differences (SSD)

\[
\sum_{(i,j) \in W} (I_1(i, j) - I_2(x + i, y + j))^2
\]

- Sum of Absolute Differences (SAD)

\[
\sum_{(i,j) \in W} |I_1(i, j) - I_2(x + i, y + j)|
\]
Stereo Reconstruction – Window Based Matching

- **Census transform**
  - Based on comparison of radiance values within the window
  - The effect of outliers is low

Hamming Distance: 3
Window Based Matching – NCC

Write regions as vectors

\[ A \rightarrow a, \quad B \rightarrow b \]

\[ \text{NCC} = \frac{a \cdot b}{|a||b|} \]

\[-1 \leq \text{NCC} \leq 1\]
Window Based Matching

• Search along the epipolar line for the best match

• Best match:
  • Maximum NCC
  • Minimum SSD or SAD
Window Based Matching – Example: NCC
Window Based Matching – Example: NCC

Reference window

Right image band

Normalized cross-correlation

Window Based Matching – Example: NCC
Window Based Matching – Example: NCC

Why is it not as good as before?

reference window

right image band

normalized cross-correlation
Window Based Matching – Neighborhood Size

- Smaller neighborhood: more details
- Larger neighborhood: fewer isolated mistakes

Window size: $w = 3$  

Window size: $w = 20$
Enhanced Matching

• Estimating disparity solely based on matching leads to errors

• There are two approaches of optimizing correspondences jointly
  - Single scan-line at a time with Dynamic Programing
  - Full 2D grid with graph cuts
Enhanced Matching – Scanline Stereo

- Coherently match pixels on the entire scan line

Slide credits James Hays, Aaron Bobick
Enhanced Matching – Scanline Stereo

- Coherently match pixels on the entire scan line

Enhanced Matching – Scanline Stereo

- Occlusion
Enhanced Matching – Scanline Stereo

- Scanline stereo generates streaking artifact


Slide credits James Hays, Aaron Bobick
Enhanced Matching – Energy Minimization

• In order to find more reliable correspondences, an energy function can be minimized.
• It can be adapted to the situation and become arbitrarily complicated.

\[
\min \ E_{\text{data}} + \lambda E_{\text{reg}}
\]

• Usually consists of a data term (consistency of matching information)
  • NCC, SSD, SAD, …

• Regularization Term
  • Smoothness (total variation regularization)

• Also different patch sizes can be included.

• Can be minimized using graph cuts.

Enhanced Matching – Energy Minimization

Window-based matching (best window size)  Ground truth

Image credits James Hays, Aaron Bobick
Enhanced Matching – Energy Minimization

Graph cut method  
Ground truth


Image credits James Hays, Aaron Bobick
Dense Matching – Challenges

Scene elements do not always look the same in the images

- **Camera-related problems**
  - Image noise
  - Lens distortion
  - Color/chromatic aberration

- **Viewpoint-related problems**
  - Perspective distortions
  - Occlusions
  - Specular reflections

- **Scene-related problems**
  - Illumination changes
  - Moving objects

- **Matching ambiguity**
  - Low texture regions
  - Repetitive texture patterns
Dense Matching – Challenges

Scene elements do not always look the same in the images

• Camera-related problems
  ▪ Image noise
  ▪ **Lens distortion**
  ▪ Color/chromatic aberration

• Viewpoint-related problems
  ▪ Perspective distortions
  ▪ Occlusions
  ▪ Specular reflections

• Scene-related problems
  ▪ Illumination changes
  ▪ Moving objects

• Matching ambiguity
  ▪ Low texture regions
  ▪ Repetitive texture patterns
Dense Matching – Challenges

Scene elements do not always look the same in the images

• Camera-related problems
  ▪ Image noise
  ▪ Lens distortion
  ▪ **Color/chromatic aberration**

• Viewpoint-related problems
  ▪ Perspective distortions
  ▪ Occlusions
  ▪ Specular reflections

• Scene-related problems
  ▪ Illumination changes
  ▪ Moving objects

• **Matching ambiguity**
  ▪ Low texture regions
  ▪ Repetitive texture patterns
Dense Matching – Challenges

Scene elements do not always look the same in the images

- **Camera-related problems**
  - Image noise
  - Lens distortion
  - Color/chromatic aberration

- **Viewpoint-related problems**
  - Perspective distortions
  - Occlusions
  - Specular reflections

- **Scene-related problems**
  - Illumination changes
  - Moving objects

- **Matching ambiguity**
  - Low texture regions
  - Repetitive texture patterns
Dense Matching – Challenges

Scene elements do not always look the same in the images

- **Camera-related problems**
  - Image noise
  - Lens distortion
  - Color/chromatic aberration
- **Viewpoint-related problems**
  - Perspective distortions
  - **Occlusions**
  - Specular reflections
- **Scene-related problems**
  - Illumination changes
  - Moving objects
- **Matching ambiguity**
  - Low texture regions
  - Repetitive texture patterns
Dense Matching – Challenges

Scene elements do not always look the same in the images

- **Camera-related problems**
  - Image noise
  - Lens distortion
  - Color/chromatic aberration
- **Viewpoint-related problems**
  - Perspective distortions
  - Occlusions
  - Specular reflections
- **Scene-related problems**
  - Illumination changes
  - Moving objects
- **Matching ambiguity**
  - Low texture regions
  - Repetitive texture patterns
Dense Matching – Challenges

Scene elements do not always look the same in the images

- **Camera-related problems**
  - Image noise
  - Lens distortion
  - Color/chromatic aberration

- **Viewpoint-related problems**
  - Perspective distortions
  - Occlusions
  - Specular reflections

- **Scene-related problems**
  - [Illumination changes](#)
  - Moving objects

- **Matching ambiguity**
  - Low texture regions
  - Repetitive texture patterns
Dense Matching – Challenges

Scene elements do not always look the same in the images

- Camera-related problems
  - Image noise
  - Lens distortion
  - Color/chromatic aberration
- Viewpoint-related problems
  - Perspective distortions
  - Occlusions
  - Specular reflections
- Scene-related problems
  - Illumination changes
  - Moving objects
- Matching ambiguity
  - Low texture regions
  - Repetitive texture patterns
Dense Matching – Challenges

Scene elements do not always look the same in the images

- **Camera-related problems**
  - Image noise
  - Lens distortion
  - Color/chromatic aberration

- **Viewpoint-related problems**
  - Perspective distortions
  - Occlusions
  - Specular reflections

- **Scene-related problems**
  - Illumination changes
  - Moving objects

- **Matching ambiguity**
  - Low texture regions
  - Repetitive texture patterns
Dense Matching – Challenges

Scene elements do not always look the same in the images

- **Camera-related problems**
  - Image noise
  - Lens distortion
  - Color/chromatic aberration

- **Viewpoint-related problems**
  - Perspective distortions
  - Occlusions
  - Specular reflections

- **Scene-related problems**
  - Illumination changes
  - Moving objects

- **Matching ambiguity**
  - Low texture regions
  - Repetitive texture patterns
Dense Matching – Assumptions

• **Appearance**
  - The projections of a scene (3D) patch should have similar appearances in all images

• **Uniqueness**
  - A point in an image will have only one match in another image

• **Ordering**
  - Order of appearance (e.g. left to right) is not altered by camera displacement

• **Smoothness**
  - Depth varies smoothly (objects have smooth surfaces)
Dense Matching – Choice of Camera Setup

**Short baseline**
- Matching is relatively easy
- Higher depth uncertainty

**Wide baseline**
- Matching is hard
- Lower depth uncertainty
Outline

• Previous lecture:
  ▪ Structure and motion loop
  ▪ Triangulation

• Today: Dense Reconstruction
  ▪ Stereo reconstruction
  ▪ The matching problem
  ▪ Multi-view reconstruction
Multi-View Reconstruction

Many views of the same scene

More images

More constraints!

Epipolar lines from different views theoretically intersect in corresponding image point.

\[ l_{31} = F_{13}^T x_1 \]
\[ l_{32} = F_{23}^T x_2 \]
Multi-View Reconstruction - Comparison

- More constraints
- Less occlusions
- More robust
- Appearance constraint can be relaxed

However...

- More complex
- Strong perspective distortions
- Propagation of calibration uncertainties

Dense matching along many images is still hard...

Precise calibration is essential!
Multi-View Reconstruction - Triangulation

Assuming we have found correct matches along \( n \) images, how do we triangulate them?

Exactly as we did before! Remember the algebraic solution…

Stack equations for all camera views:

\[
\begin{bmatrix}
A_1 \\
A_2 \\
\vdots \\
A_n
\end{bmatrix} X = 0
\]
Multi-View Reconstruction - Algorithms

- Large number of algorithms → Out of scope of this lecture

- All have in common the same assumptions
  - Lambertian surfaces (no specular reflections)
  - Static scenes (dynamic scenes → 4D recon., out of scope)
  - Camera parameters are available (projection matrices are known)
  - Underlying photometric consistency (NCC, etc)

They can be roughly classified according to the adopted scene representation!
Scene Representation

- Point cloud
- Depth map
- Voxel grid
- Mesh
Scene Representation – Voxel Grid

• **Properties**
  - Divides the scene into cubes (voxels)
  - Needs to sample the 3D space
  - Voxels are marked as occupied/empty according to a photometric consistency measure (NCC, SSD…)
  - Accuracy depends on size of voxels
  - No assumptions on the scene required
  - Memory allocation issues
Scene Representation – Voxel Grid

- Properties
  - Divides the scene into cubes (voxels)
  - Needs to sample the 3D space
  - Voxels are marked as occupied/empty according to a photometric consistency measure (NCC, SSD…)
  - Accuracy depends on size of voxels
  - No assumptions on the scene required
  - Memory allocation issues
Scene Representation – Voxel Grid

• Properties

- Divides the scene into cubes (voxels)
- Needs to sample the 3D space
- Voxels are marked as occupied/empty according to a photometric consistency measure (NCC, SSD…)
- Accuracy depends on size of voxels
- No assumptions on the scene required
- Memory allocation issues
Scene Representation – Voxel Grid

- **Properties**
  - Divides the scene into cubes (voxels)
  - Needs to sample the 3D space
  - Voxels are marked as occupied/empty according to a photometric consistency measure (NCC, SSD…)
  - Accuracy depends on size of voxels
  - No assumptions on the scene required
  - Memory allocation issues
Scene Representation – Depth Map

**Properties**

- Usually one depth map per view (by matching and triangulating features in small sets of images)
- No sampling of 3D space
- Simple
- Suitable for multi-core
- Depth maps have to be merged
Scene Representation – Depth Map

• Properties
  - Usually one depth map per view (by matching and triangulating features in small sets of images)
  - No sampling of 3D space
  - Simple
  - Suitable for multi-core
  - Depth maps have to be merged
Scene Representation – Point Cloud

- Properties
  - Local planar approximation of the surface at each point
  - Usually a normal vector is associated to each point
  - Sparse to dense reconstruction (region growing)
  - Suitable for multi-core
  - May be noisy
Scene Representation – Point Cloud

• Properties
  - Local planar approximation of the surface at each point
  - Usually a normal vector is associated to each point
  - Sparse to dense reconstruction (region growing)
  - Suitable for multi-core
  - May be noisy
Scene Representation – Point Cloud

- **Properties**
  - Local planar approximation of the surface at each point
  - Usually a normal vector is associated to each point
  - Sparse to dense reconstruction (region growing)
  - Suitable for multi-core
  - May be noisy
Scene Representation – Point Cloud

• Properties
  - Local planar approximation of the surface at each point
  - Usually a normal vector is associated to each point
  - Sparse to dense reconstruction (region growing)
  - Suitable for multi-core
  - May be noisy
Scene Representation – Point Cloud
Scene Representation – Mesh

• Properties
  - Models the surface as a connected set of planar facets
  - Usually triangular meshes
  - Triangles are locally good approximations of the surface
  - Less noisy
  - Good for optimization (mesh refinement)
  - May become difficult to handle for complex surfaces
Scene Representation – Mesh

• Properties
  - Models the surface as a connected set of planar facets
  - Usually triangular meshes
  - Triangles are locally good approximations of the surface
  - Less noisy
  - Good for optimization (mesh refinement)
  - May become difficult to handle for complex surfaces
Scene Representation – Mesh

• Properties
  - Models the surface as a connected set of planar facets
  - Usually triangular meshes
  - Triangles are locally good approximations of the surface
  - Less noisy
  - Good for optimization (mesh refinement)
  - May become difficult to handle for complex surfaces
Scene Representation – Mesh
Scene Representation – Mesh
References


- Furukawa, Y.; Curless, B.; Seitz, S. M. & Szeliski, R. Towards Internet-Scale Multi-View Stereo, CVPR, 2010


Interesting Links

- Bundler (SFM tool): http://www.cs.cornell.edu/~snavely/bundler/
- Middleburry stereo data set: http://vision.middlebury.edu/stereo/data/
- Middlebury multi-view data set: http://vision.middlebury.edu/mview/
- Efficient Large Scale Multi-View Stereo for Ultra High Resolution Image Sets: http://cvlab.epfl.ch/research/surface/emvs/
Thank you!