Part 4

Mobile metric capture and reconstruction
Computer vision and mobile applications

Digital photos (auto enhance)
Panoramic photos (autostitch)
Biometrics (fingerprints)
Face detection
HDR

Image Search
Visual Search
Landmark recognition
Augmented Reality
3D capture
VSLAM

1990 2000 2010

Material Capture
Computer vision and mobile applications

- **Mostly 2D**
  - Image enhancement
  - Image stitching
  - Image matching
  - Object detection
  - Texture classification
  - Activity recognition
  - ...

- **Mostly 3D**
  - Camera localization
  - Pose estimation
  - 3D shape recovery
  - 3D scene reconstruction
  - Material/appearance recovery
  - Augmented reality
  - ...

Applications made possible by specific features of mobile devices!

- Features
  1. Mobility
  2. Camera
  3. Active light
  4. Non-visual sensors
  5. Processing power
  6. Connectivity
  7. Display
Features (1/7): Mobility

- **Consumer**
  - Smartphones
  - Tablets

- **Embedded**
  - Autonomous driving
  - Assistive technologies

- **Specific**
  - Drones
  - Robots
Features (1/7): Mobility

• Consumer
  – Smartphones
  – Tablets

• Embedded
  – Autonomous driving
  – Assistive technologies

• Specific
  – Drones
  – Robots

On-site applications / Personal applications / Motion and/or location taken into account / Embedded solutions
Features (2/7): High-res/flexible camera

- **Common features**
  - High resolution and good color range (>12 MP, HDR)
  - Small sensors (similar to point and shoot cameras – approx. 1/3”)
  - High video resolution and frame rate (4K at 30fps)

- **Wide variety of field of views**
  - Standard, fisheye, spherical

- **Specialized embedded cameras**...
  - Better lenses and sensors...
Features (2/7): High-res/flexible camera

• **Common features**
  - High resolution and good color range (>12 MP, HDR)
  - Small sensors (similar to point and shoot cameras – approx. 1/3”)
  - High video resolution and frame rate (4K at 30fps)

• **Wide variety of field of views**
  - standard, fisheye, spherical

• **Specialized embedded cameras…**
  - Better lenses and sensors…

Visual channel is the primary one

Computational photography

Apps analyze/use snapshots or videos
Features (3/7): Active lighting

- All smartphones have a flashlight
  - LED source at fixed distance from camera
- Custom devices have integrated emitters
  - Google TANGO / Microsoft Kinect
    - Integrated depth sensor
- Leads to specialized capture procedures
Features (3/7): Active lighting

- All smartphones have a flashlight – LED source at fixed distance from camera
- Custom devices have integrated emitters – Google TANGO / Microsoft Kinect
  - Integrated depth sensor
  - Leads to specialized capture procedures

Specialized capture procedures exploiting synchronization of illumination and visual sensing

Features (4/7): Non-visual sensors

- **Absolute reference**
  - **GPS / A-GPS**
    - Mainly for outdoor applications
  - **Magnetometer**
    - Enable compass implementation
    - Often inaccurate for indoor

- **Relative reference**
  - **Accelerometer**
    - Variable accuracy (sensitive to temperature)
    - Good metric information for small scale scene
  - **Gyroscope**
    - Very good accuracy for device relative orientation

- **Synced with camera!**
Features (4/7): Non-visual sensors

- Absolute reference
  - GPS / A-GPS
  - Mainly for outdoor applications
  - Magnetometer
    - Enable compass implementation
    - Often inaccurate for indoor
  - Relative reference
    - Accelerometer
      - Variable accuracy (sensitive to temperature)
      - Good metric information for small scale scene
    - Gyroscope
      - Very good accuracy for device relative orientation
- Synced with camera!

Data fusion!

Ex. Garro et al. **Fast Metric Acquisition with Mobile Devices.** VMV 2016
Features (5/7): Processing power

• Growing performance of mobile CPU+GPU
  – *(see previous sections)*

• Capable to execute computer vision pipeline on mobile device
  – i.e. *OpenCV* for Android

• Some limitations due to power consumption
Features (5/7): Processing power

- Growing performance of mobile CPU+GPU (see previous sections)
- Capable to execute computer vision pipeline on mobile devices (i.e. OpenCV for Android)
- Some limitations due to power consumption

**On-board pre-processing or even full processing**

Features (6/7): Connectivity

- **Many connectivity options**
  - **Local area**: NFC, Bluetooth, Bluetooth Low Energy, Wi-Fi 802.11x
  - **Wide area**: Cellular wireless networks: 3G/4G/5G

- **Mobile devices can connect at local or wide area at reasonable speed**
  - Typical LTE/4G: 18 Mbps down, 9.0 Mbps up
  - Typical Wi-Fi: 54Mbps (g), 300Mbps (n), 1Gbps (ac).

- **Lo-cost -> No-Costs**
Features (6/7): Connectivity

- Many connectivity options
  - Local area: NFC, Bluetooth
    - Bluetooth Low Energy, Wi-Fi 802.11x
  - Wide area: Cellular wireless networks: 3G/4G/5G
- Mobile devices can connect at local or wide area at reasonable speed
  - Typical LTE/4G: 18 Mbps down, 9.0 Mbps up
  - Typical Wi-Fi: 54Mbps (g), 300Mbps (n), 1Gbps (ac).
- Lo-cost -> No-Costs

Load balancing (client / server)
Access to large databases (e.g., search)
Communication

Features (7/7): Display!

- Hi-res/hi-density display
  - Data presentation!
- Co-located with camera + other sensors
  - Tracking during capture!
- Touch screen
  - Co-located user-interface
  - (UI also may exploits other sensors)
Features (7/7): Display!

- Hi-res/hi-density display
  - Data presentation!
  - Co-located with camera + other sensors
    - Tracking during capture!
  - Touch screen – Co-located user interface
    - (UI also may exploits other sensors)

**Data/result presentation**

**Guided capture / Augmentation**

Wrap-up: mobile apps characterized by the exploitation of mobile device features

- Features
  1. Mobility
  2. Camera
  3. Active light
  4. Non-visual sensors
  5. Processing power
  6. Connectivity
  7. Display
Example 1

DATA FUSION FOR METRIC CAPTURE
Metric acquisition with a commodity mobile phone

• **Goal**
  – Capture 3D models with real-world measures

• **Data fusion approach**
  – Exploit synchronization of visual sensor & IMU to capture scenes in real-world units

Garro et al. [*Fast Metric Acquisition with Mobile Devices*]. VMV 2016
Structure-from-Motion + Dense reconstruction

- SfM reconstructs a point cloud from a series of images
  - 3D positions of (sparse) matched features
  - Camera positions and orientations
- Many approaches for densification
  - Pipeline showed to work at interactive rates on phones (Taskanen et al 2013)
- SCALE AMBIGUITY
Data fusion: Visual + IMU

- Use sensors synced with visual channel
  - **GPS+Magnetometer** generally not applicable
  - **IMU** returns orientation and acceleration in real world units

- **Idea**
  - track camera movement with IMU during visual capture
  - use IMU data to find out the real-world distance between SfM camera positions, resolving the scale ambiguity
Data fusion: Visual + IMU

• The accelerometer returns acceleration
• Therefore, we should be able to compute the displacement between two camera positions as

\[
x(T_1, T_2) = \left\| \int_{T_1}^{T_2} \left( v(T_1) + \int_{T_1}^{t'} a(t) \, dt \right) \, dt' \right\|
\]

• Not so easy: onboard IMU sensors are biased and noisy and SfM camera positions are sparse
Data fusion approaches (1/5)

- Match position from IMU integration with position from SfM, coping with noise/bias by extensive filtering

- Requires LONG acquisition times and LONG offline processing times

A new approach to vision-aided inertial navigation [Tardif et al 2010]
Data fusion approaches (2/5)

- **Ad-hoc online solutions taking into account IMU characteristics**
  - Segment motion in “swift movements” with large accelerations
  - Integration of IMU acceleration to derive position matched with SfM
  - Continuous process of outlier rejection and re-estimation of scale

Live metric 3D reconstruction on Mobile Phones [Tanskane et al. 2013]

\[
\arg \min_{\lambda} = \sum_{i \in I} \left\| \bar{x}_i - \lambda \bar{y}_i \right\|^2
\]

One estimate of $\lambda$ at the end of each swift movement
Estimation of scale $\lambda$ only on inlier set $I$

- **Working but motion-dependent and prone to accumulation error due to integration**
Data fusion approaches (3/5)

- **Match accelerations from IMU with accelerations from SfM**

  \[
  \min \sum_{k=1}^{m} \| a_w(t_k) - \hat{a}_w(t_k) \|^2
  \]

  Camera trajectory estimation using inertial measurements and Structure from Motion results [JungTaylor2001]

  spline parameters

- **Works off-line and assumes high-accuracy (robotics) IMU**
Data fusion approaches (4/5)

• Match accelerations from IMU with accelerations from SfM at SfM frame-rate (large baseline!)
  – **Downsample and anti-alias** IMU samples at SfM frame rate
  – Optimize scale and bias

Hand-waving away scale [Ham et al. 2014]

\[
\arg\min_{s,b} \eta \{ s \cdot \hat{A}_V + 1 \otimes b^T - DA_I R_I \}
\]

\[ D: \text{ convolutional matrix for antialiasing and downsampling IMU signal} \]

• Requires very long acquisition times due to downsampling at SfM rate
Data fusion approaches (5/5)

- Match accelerations from IMU with accelerations from SfM at IMU frame-rate (small baseline!)
  - **Upsample** SfM samples at high rate using all available visual data
  - Estimate acceleration from upsampled transforms and match them to IMU samples using robust fitting

Fast Metric Acquisition with Mobile Devices. [Garro et al. 2016]

\[
\arg\min_{s,R} \| A_c - s R A_s \|
\]

- Fast, coping with large errors and noise
Vision Module Pipeline

- Tracking features
- Video frames
- IMU samples
- Stereo frames
- Large baseline
- Small baseline
- 3D-2D correspondences
- N > M camera poses
- 3D alignment
- 3D points in a common reference system

PnP
Vision Module

- Traces Shi-Thomasi features
- When baseline is large enough
  - Estimate Essential Matrix, that is, relative camera pose between $f_0$ and $f_i$
  - Calculate a 3D point for each feature point
- Note: each pair of cameras has its own reference system
Vision Module

• Global registration
  – M point clouds
  – A subset of features is present in each point cloud
  – Use feature correspondence to align all the point cloud in the same reference system

• Cameras upsampling
  – Features are tracked for all frames
  – Use aligned point cloud and tracking position to estimate cameras for all frames with Perspective-n-Point \((\text{PnP})\)
Recovering the scale factor (1/2)

IMU accelerations

\[ A_s = \begin{pmatrix}
    a_s^x(t_0) & a_s^y(t_0) & a_s^z(t_0) \\
    \vdots & \vdots & \vdots \\
    a_s^x(t_K) & a_s^y(t_K) & a_s^z(t_K)
\end{pmatrix} \]

Camera accelerations

\[ A_c = \begin{pmatrix}
    p_c''(t_0)^T R_c(t_0) \\
    \vdots \\
    p_c''(t_K)^T R_c(t_K)
\end{pmatrix} \]

Problem to solve

\[ \argmin_s \{ \| A_c - s A_s \| \} \]
Recovering the scale factor (2/2)

- LS, gradient descent (et similia) poorly conditioned
  - Not so many data
  - Severe outliers

- Robust fitting use RANSAC approach
  - Use MLESAC robust estimator to maximize likelihood rather than just the number of inliers

- Introduce rotation matrix $R$
  - Account for orientation bias
  - Improve RANSAC performance

\[
\arg\min_{s} \{ ||A_c - sA_s|| \} \\
\arg\min_{s,R} \{ ||A_c - sRA_s|| \}
\]
## Results

<table>
<thead>
<tr>
<th>Scene Name</th>
<th>Real scale m / s.u.</th>
<th>Acquisition info</th>
<th>Our approach m / s.u.</th>
<th>Simple scaling m / s.u.</th>
</tr>
</thead>
<tbody>
<tr>
<td>3D printer</td>
<td>2.094</td>
<td>17.0</td>
<td>65</td>
<td>883</td>
</tr>
<tr>
<td>Scanner setup</td>
<td>3.565</td>
<td>9.8</td>
<td>53</td>
<td>641</td>
</tr>
<tr>
<td>Desktop</td>
<td>6.520</td>
<td>11.3</td>
<td>48</td>
<td>596</td>
</tr>
<tr>
<td>Statuettes</td>
<td>2.602</td>
<td>11.5</td>
<td>53</td>
<td>607</td>
</tr>
<tr>
<td>Office desk</td>
<td>1.977</td>
<td>30.4</td>
<td>88</td>
<td>471</td>
</tr>
<tr>
<td>Office workstation</td>
<td>3.95</td>
<td>12.3</td>
<td>37</td>
<td>1307</td>
</tr>
<tr>
<td>Ara pacis</td>
<td>1.568</td>
<td>30.07</td>
<td>77</td>
<td>1569</td>
</tr>
<tr>
<td>Workstation (Fastest)</td>
<td>0.707</td>
<td>9.9</td>
<td>34</td>
<td>1305</td>
</tr>
<tr>
<td>Desk fast motion</td>
<td>6.918</td>
<td>14.8</td>
<td>74</td>
<td>1718</td>
</tr>
</tbody>
</table>

- Median error 4% (wrt 10-15% of other STAR solutions)
Example 2

DATA FUSION AND COMMUNICATION FOR INDOOR CAPTURE
Indoor capture + presentation

• Creation and sharing of indoor digital mock-ups
  – Exploiting the capabilities of modern mobile devices

• Much interest/applications (security, location awareness, …)
  – Need to capture visual information together with room structure
Typical solutions

• Indoor capture and modeling
  – Manual modeling
  – Semi-automatic methods based on high-density data
    • Laser scanning
      – Professional but expensive, limited to specific applications
    • Multi-view stereo from photographs
      – Generally cost effective but hard to apply in the indoor environment
        » Walls poorly textured, occlusions, clutter
        » Furthermore: need for heavy MW constraints, computationally demanding


Furukawa et al. Reconstructing Building Interiors from Images. ICCV 2009
Examples using low-cost mobile devices

- **Interactive capture and mapping of indoor environment**
    - Floor corners marked via an augmented reality interface
    - Manual editing of the room and floor plan merging using the screen interface
  - Sankar and Seitz: Capturing indoor scenes with smartphones (UIST2012)
    - Corners marked on the screen during video playback
Exploiting panoramic images

- 360 degrees images are easy to capture using common devices
  - Interactive apps using IMU + GUI + automatic stitching
  - Dedicated cameras

- 360 degrees images are easy to navigate
  - Spheremaps + emerging formats video+image formats
  - VR devices for immersion

- What about analyzing them?
Finding the room structure

• Take one spheremap per room
  – Equirectangular images generated by a mobile device
    • Vertical lines aligned with the gravity vector
    • Image approx. oriented towards magnetic North
  – Eventually use IMU + Visual features for stitching

• Track user motion to identify connections between rooms
  – Use IMU + Visual Features for tracking

• Solve local + global optimization to find indoor structure
  – Multi-room environment
Finding the room structure

- Analyze spheremap to extract single room structure
  - Room model considers vertical walls
  - Extract edges and filter out regions likely far from top/bottom edges of walls
  - Find wall height
    - Voting scheme used to extract most likely wall height by maximizing pairs of matching wall-floor / wall-height edge pixels
  - Fit 2.5D room model to recovered wall edge map

- Uses specialized transform to speed-up computation

Vary $h_w$ results in a transform scaling
Finding the rooms structure

- **Iterated to map the entire floor-plan**
  - Mobile tracking of user’s direction moving between adjacent rooms creates a connected room graph
  - Doors position identification in the image by computer vision
  - Doors matching according with graph
  - Rooms displacement
  - **Global optimization of combined model**

Pintore et al. *Omnidirectional image capture on mobile devices for fast automatic generation of 2.5D indoor maps.* IEEE WACV 2016
### Results

<table>
<thead>
<tr>
<th>Scene Name</th>
<th>Features Area [m²]</th>
<th>Np</th>
<th>Area error</th>
<th>Wall length error</th>
<th>Wall height error</th>
<th>Corner angle error</th>
<th>Editing time</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>MP</td>
<td>Ours</td>
<td>MP</td>
<td>Ours</td>
<td>MagicPlan</td>
</tr>
<tr>
<td>Office H1</td>
<td>720</td>
<td>10</td>
<td>2.95%</td>
<td>1.78%</td>
<td>35 cm</td>
<td>15 cm</td>
<td>2.0 cm</td>
</tr>
<tr>
<td>Building B2</td>
<td>875</td>
<td>25</td>
<td>2.50%</td>
<td>1.54%</td>
<td>30 cm</td>
<td>7 cm</td>
<td>6.0 cm</td>
</tr>
<tr>
<td>Commercial</td>
<td>220</td>
<td>6</td>
<td>2.30%</td>
<td>1.82%</td>
<td>25 cm</td>
<td>8 cm</td>
<td>12.0 cm</td>
</tr>
<tr>
<td>Palace</td>
<td>183</td>
<td>3</td>
<td>16.86%</td>
<td>0.20%</td>
<td>94 cm</td>
<td>5 cm</td>
<td>45.0 cm</td>
</tr>
<tr>
<td>House 1</td>
<td>55</td>
<td>5</td>
<td>21.48%</td>
<td>2.10%</td>
<td>120 cm</td>
<td>16 cm</td>
<td>15.0 cm</td>
</tr>
<tr>
<td>House 2</td>
<td>64</td>
<td>7</td>
<td>28.05%</td>
<td>1.67%</td>
<td>85 cm</td>
<td>8 cm</td>
<td>18.0 cm</td>
</tr>
<tr>
<td>House 3</td>
<td>170</td>
<td>8</td>
<td>25.10%</td>
<td>2.06%</td>
<td>115 cm</td>
<td>15 cm</td>
<td>20.0 cm</td>
</tr>
</tbody>
</table>

Pintore et al. Omnidirectional image capture on mobile devices for fast automatic generation of 2.5D indoor maps. IEEE WACV 2016

- Reasonable, fast reconstruction with rough structure and visual features
Sharing the indoor model

• **Indoor model**
  – Exploration graph
    • Each node is a spheremap/room
    • edges (yellow) are transitions between adjacent rooms
    • Stored on a server (standard http Apache2)
  – Panoramic images
    • Mapped according with the graph

• **Interactive exploration**
  – Room
    • WebGL fragment shader
    • dragging to change view orientation and pinching to zoom in/out
  – Passages
    • **Real-time rendering** of the transitions between rooms
      – Exploiting geometric model stored on the server
      – Performance improvement compared to use precomputed videos
    • Suggested paths
Some results

Live demo: [http://vcg.isti.cnr.it/vasco/](http://vcg.isti.cnr.it/vasco/)

Click on the dataset on the left column to start

3D reconstruction of a 655 mq office with 19 rooms. This environment was acquired with a mobile phone (HTC One M8)

Reconstruction of a 70 rooms floor of the NHV ministry at Den Haag, Netherlands. The whole model was acquired with a Ricoh Theta S camera
Wrap-up: mobile apps characterized by the exploitation of mobile device features

- Features
  1. Mobility
  2. Camera
  3. Active light
  4. Non-visual sensors
  5. Processing power
  6. Connectivity
  7. Display
After the break: rendering!

BREAK!