The Computational Array Camera

Dan Lelescu
Chief Imaging Scientist

Pelican Imaging Corporation

September 23, 2014
Today we snap more photos every 2 minutes than humanity did as a whole in the 1800s

[IDC, Technorati]
Modern camera evolution

Current consumer camera

Some “computational” features can be added w/o HW modifications (e.g., HDR, video super-resolution, generating panoramas)

The theoretical plenoptic camera captures all information at a point in space

Practical, lower-dimensionality computational camera instantiations
R&D scope for computational imaging

- **Plenoptic image acquisition**
  - Camera design, calibration, synchronization
  - Space/time sampling, optimal sampling (aliasing?)
  - Typically, huge amount of data are generated

- **Plenoptic processing**
  - Reconstruction of imaged scene data, plenoptic representations for specific purposes, feature generation and associated apps (e.g., depth map and usage)
  - Coding (for storage, transmission, display)
  - Formats

- **Plenoptic signal communication**
  - Transport issues (e.g., error resilience) specific to this domain
  - Bandwidth!

- **Rendering/displays, printing**
  - Display devices (to take advantage of new imaging capability)
  - 3D printing
Outline

- The plenoptic function
- Computational cameras as codecs
- The Pelican Imaging array camera
The plenoptic function and its parameterizations
The plenoptic function

- The *plenoptic function* was introduced formally in [Adelson 1991].
  - Describes all light information collected at a point in space-time

- The plenoptic function is originally a 7D function,

\[ f(V_x, V_y, V_z, \Theta, \Phi, \lambda, t) \]

where

- \( V_x, V_y, V_z \) viewpoint coords.
- \( \Theta, \Phi \) ray direction
- \( \lambda \) wavelength
- \( t \) time

- By fixing various parameters in the plenoptic function, one obtains more restrictive representations.
Of particular interest: 4D Parameterization of Light Field

- Integral photography [Lippmann 1908]

- Light fields are 4D parameterizations of the plenoptic function
  - Light Fields [Levoy 1996] and Lumigraphs [Gortler 1996]: a ray is indexed by its intersection with two parallel planes.

  **Two-plane parameterization**

  ![Diagram](attachment:diagram.png)

  - Assumption of space free of occluders (to reduce from 5D to 4D); six pairs of planes surrounding the convex hull of the object being imaged
4D Light Field capture

- Spatio-angular capture, whether
  - of the main lens image, using a microlens array (like a relay-lens system) near sensor
  - of the scene, using lens arrays

[Levoy 1996]

[Ng 2005]

© Pelican Imaging 2014. All rights reserved.
Brief overview of computational cameras*

* Extensive literature available, this is a sparse sampling

Credit: http://www.instructables.com/id/ DIY-Camera-Array-1-Computational-Photography-Prim/
Computational camera as codecs

- Optics and/or camera structure (e.g., case of arrays) “encode” the imaged scene in various ways

- Typically, the closely-adapted digital processing “decodes” the information to produce the desired features of the computational camera

( As usual, an image/video codec may be inserted between the two, esp. given the volume of data that may be generated).
Computational camera codecs (contd.)

- Aspects of such devices can just as well be cast in the language of information theory

- E.g.,
  - what constitute “good” views of the scene?
    - Viewpoint entropy [Vasquez 2001],
      \[ I = -\sum_{i=1}^{n} \frac{A_i}{A_t} \log \frac{A_i}{A_t}, \]
      where \( n \) is the number of facets of objects seen in the scene, \( A_i \) is the projected area of face \( i \) over the sphere centered at viewpoint \( A_t \) is the total area of the sphere
  - how “efficient” is the information transfer across acquisition & processing
  - efficient source coding of generated data, e.g., MPEG-4 Part10 predictive Multiple View Coding (MVC), or “just-in-time” (JIT)-decode representations (e.g., [Lelescu 2004])
The “encoding” of acquisition: Approaches [1]

- **Object Side Coding**
  - Involves an optical element attached to a conventional lens
  - Examples include:
    - Catadioptric Lenses (Lens + mirrors) [Chahl 1997, Baker 1999, Lelescu 2002]
    - Bi-prism Stereo [Lee 1998]

- **Pupil Side Coding**
  - Involves an optical element attached to the pupil plane of conventional lens
  - Examples include:
    - Cubic Phase Plates [Dowski 1995]
    - Coded Aperture [Levin 2007]
The “encoding” of acquisition: Approaches [2]

- **Focal Plane Coding**
  - Involves an optical element placed close to the sensor/detector
  - Examples include:
    - Pixel-wise control of exposure [Nayar 2003]
    - Use of microlens arrays [Adelson 1992], [Ng 2005], [Lumsdaine 2009], [Georgiev 2010],
    - Attenuation masks [Veeraraghavan 2007]

- **Illumination Coding**
  - Spatial or temporal control of flash to code captured images
  - Examples include:
    - Robust 3D using space-time stereo [Zhang 2003]
    - High speed 3D reconstruction using structured light, e.g., [Gong 2010]
    - Kinect [Microsoft]
The “encoding” of acquisition: Approaches [3]

- Camera clusters and arrays
- No optical coding need be involved, but “coding” occurs due to information capture across individual cameras
  - Additional coding may involve high-frequency scene information captured in phase-offset aliased array images
- Examples include:
  - Multi-baseline stereo [Okutomi 1993]
  - TOMBO array [Tanida 2001]
  - Flexible Camera Arrays [Nomura 2007]
  - Stanford Camera Array [Wilburn 2005]
  - Pelican Imaging Camera Array [Venkataraman 2008]
The encoding of acquisition:
A few category examples
Object Side Coding

E.g.,

- Bi-prism stereo [Lee 1998]
- Catadioptric omnidirectional capture and processing [Lelescu 2002]
- Extended depth of field (EDOF) through wavefront coding, e.g., [Dowski 1995]
  - A standard optics is modified by a phase mask
  - The phase mask alters the wavefront such that point-spread function does not change appreciably

- Phase-mask optics “coupled” with a deconvolution process enable a large-DoF image recovery, since the blur kernel is largely invariant with distance, e.g., on-sensor EDOF solution [Lelescu 2009].
Patterned occluder within the aperture of the camera
- Creates a coded aperture
- The aperture filter can now discriminate between depths

Recover the scale of the blur which allows one to
- Determine the depth (since the scale of the blur is dependent on depth)
- Recover the image by inverting the blur at each depth level

**Figure 3:** Left: Top, a standard Canon 50mm f/1.8 lens with the aperture partially closed. Bottom, the resulting blur pattern. The intersecting aperture blades give the pentagonal shape, while the small ripples are due to diffraction. Right: Top, the same model of lens but with our filter inserted into the aperture. Bottom, the resulting blur pattern, which allows recovery of both image and depth.
Focal Plane Coding [Adelson 1992]

By placing a lenticular array close to the sensor plane of the main lens, the resulting ‘plenoptic’ camera provides depth cues.

**FIGURE 1.** In a conventional camera, only a 2-D image is captured at the sensor plane. Because of this, it is impossible to tell whether the point being imaged is further from or nearer to the image plane.

**FIGURE 2.** In a plenoptic camera, an array of microlenses is used to sample the angular information of light rays. When the object is out-of-focus point, a blurred spot is formed on the microlens array, but depending on the incident angle of the light, different pixels will be illuminated.
Focal Plane Coding (contd.)

- Spatio-angular sampling using a microlens array: Plenoptic camera [Ng 2005]; Focused plenoptic camera [Lumsdaine 2009], [Georgiev 2010]
  - Differences in focusing the main lens image and the microlenses → differences in reconstruction and render resolution

- For example, in plenoptic camera [Ng 2005]
  - Image: integrate within microlens sub-images
  - Refocusing the image:
    - refocusing = summing windows extracted from several microlenses

© Pelican Imaging 2014. All rights reserved.
Camera clusters – Virtualized Reality [Rander 1997]

- A Gantry (or Dome) is built to house cameras at different points of view
- The cameras capture multiple points of view
- Synthesize intermediate views from positions on the gantry, or from points inside the convex hull of the gantry
What can an array camera do?

Features
- Small form factor (very thin, e.g., 3.5mm) computational camera
- Restore higher resolution imagery from low-resolution input – super-resolution (SR) – a balanced angular vs. spatial resolution (in 4D)
- Virtual viewpoint (whether native res., or further super-resolved)
- Dynamic focus; post-capture refocus/synthetic aperture; re-lighting, etc.
- Natively co-located (RGBZ) depth map
  - Consumer depth-driven applications, depending on design
- Video from an LF camera, can use depth features for applications

The balancing of strengths in the multi-feature “star-graph” is part of design constraints. Some trade-offs have to be made (no free lunch)

Camera instantiations can be built, with different combination of features and trade-offs.
Computational camera design typically more complex than traditional camera

Level 1: proof of concept design/simulations, more limited, controlled-condition testing

Level 2: physical emulation or build, and more extensive testing, but not "consumer-grade", e.g.,
- small number of cameras built, may use manual or per-image/class tuning
- manufacturing tolerances

Level 3: full-fledged camera module, meant for field operation, e.g.,
- large numbers of cameras built, extensive testing
- robustness is paramount, manufacturing tolerances
- stable adaptive tuning to practically uncontrolled imaging conditions
- (self-diagnosis/correction in the field)
Building computational cameras (contd.)

- New HW challenges for an array camera, e.g.,
  - Performance and tolerances of components
  - New composite metrics, and tolerances for the array
  - Alignment techniques

- Critical to design jointly the Encoder (acquisition HW) and Decoder (digital processing)
  - Approach/algorithms/assumptions that will function within design constraints, and achieve desired functionality
  - Develop solutions from classes of advanced statistical signal processing approaches (esp. able to account for modeling/characterization uncertainties)
What does the array camera “encode”?  

- Geometry and intensity information in 4D (u,v,s,t):  
  - **Depth information** (disparity, in image space)  
    - Decode: Geometric registration and parallax detection  
  - High frequency information above sensor Nyquist (if so designed) in the form of phase-offset aliased input data \(\rightarrow\) **super-resolution decoding**  
    - Can be used (even at varying strength) to complement other features, e.g., refocus, virtual view, etc.  
  - **Dynamic range information** (exposure bracketing in array)  
    - For “single shot” HDR  
    - Decode: HDR reconstruction
Sample considerations for PiCam design

- PiCam HW ("encoder"): Optics, sensors (and module integration)

- PiCam SW ("decoder") Core processing
  - Parallax detection
  - Super-resolution

- PiCam SW applications
Encoder: Camera module structure
Encoder: Sample design considerations: Optics

- Each channel can be designed for a narrower spectral band
  - Small bandwidth – less achromatization needed, or better performance with the same effort
  - Separated color channels – each channel can be focused properly
- Small optical format reduces aberrations and influence of form errors
Example: monolithic lens array
Encoder: Sensor Design

- In the case of a Bayer-pattern, the CFA is deposited on the pixels.

- Once each focal plane is monochrome the filter can be moved from sensor to the lens!

- Benefits:
  - Cheaper lithography & material
  - Reduced pixel stack height $\rightarrow$ increased pixel MTF (less crosstalk)
Decoder: High-level core- and derived- functions

“Feature” processing
- Virtual Viewpoint
- Refocus, Relighting
- Co-located Depth

Software Processing Blocks:
- Normalization
- Parallax
- Fusion SR

Hardware Processing Blocks:
- LR Images
- Color Proc. Gamma
- Post-Processing
- MAP SR

© Pelican Imaging 2014. All rights reserved.
“Decoding” depth: Parallax detection & regularization

- **First level**: joint (multi-camera) parallax detection, multi-channel (e.g., RGB)
  - Spatial arrangement of Color Filters (cameras) very important (occlusion handling)

- **Second level**: refinement through a “visibility processing” reasoning
  - Basically, verify validity of initial result by testing the obtained geometry against array constraints

- Saves more geometry \{u,v,s,t\} information for the subsequent “uncertainty processing” (or hypothesis testing) in the MAP reconstruction

- For certain applications, a further depth –map regularization may be performed to fill in missing data.
Example: Depth map (w/ confidence map)
Decoding: Recovering resolution

The resolution is a function of multiple parameters, including:

- Optical Format of each camera in array
- Number and arrangement of cameras
- F/# (determines diffraction limit), aberrations, and resulting OTF of optics
- Pixel size (sampling rate, aliasing)
- Sensor MTF
- Super resolution factor

System_{MTF} = \text{Optics}_{OTF} \times \text{Sensor}_{MTF}

Traditional camera MTF, aliasing is undesired (OLPF used)

Array component camera MTF. Exploit aliasing to SR recover.
Important to model, characterize, or determine “degradations”:

- multiple blurs (e.g., optics, sensor)
- geometry (e.g., scene-independent distortions, scene-dependent parallax)
- Noise (both imaging, and impact of cumulative estimator noise)

Trust (to some degree) but verify:

- The processing design starts with built-in assumption of uncertainties → most appropriate statistical models adopted → toward robust functionalities
Decoding: Super-resolution reconstruction

- Leverage Bayesian philosophy
  - No “turn-key” solution; needs dedicated derivations

- Probabilistic models incorporate general, and system-specific priors
  - Optics characteristics – e.g., PSFs, geometry
  - Sensor – e.g., MTF, Noise
  - Array geometry

- A MAP (maximum a-posteriori) restoration approach provides a powerful unified framework for processing
  - Addresses uncertainty from prior stages (e.g., parallax, normalization)
  - Stabilizes solution

- Cross-channel fusion of Red/Blue channels, along with selective transfer of weighted MAP-gradients from Green
  - Could optimally be done “inside the loop”, but more expensive
“Decoder”: The Super-resolution reconstruction (contd.)

By this aliasing measure (percent aliasing & visual):
SR Factor
1140/480 = 2.4

Other measures are possible, as long as applied consistently.

Lens: F3.1
Array: 16 cams
1000×750 each
1.75μ pixels

© Pelican Imaging 2014. All rights reserved.
“Decoder”: Reconstruction animation

COLOR RECONSTRUCTED
PiCam: More examples and applications
Reconstruction

Single subarray low-res image

Super resolved image

© Pelican Imaging 2014. All rights reserved.
Reconstruction (indoor, higher noise)
Reconstruction (far)
Reconstruction, DoF/resolution comparison
Depth map + regularization (outdoor depth)

Input Image

Regularized Depth
Applications: Refocus
Applications: Re-Lighting
Applications: Point clouds (capture at 10-15cm)
Future applications: Close object scan
Summary

➢ Computational cameras
  ▪ Can provide set of unique/interesting/useful features
  ▪ Ongoing efforts to bring them to consumer

➢ Array camera
  ▪ Core functionalities:
    ▪ Provides depth
    ▪ Higher-resolution than that of individual component camera
  ▪ Form factor adapted to application domain (including very thin, mobile form-factor camera)
  ▪ With higher computational budgets, more (or increased quality) features could be offered in an even small form factor.
More information at
www.pelicanimaging.com

Thank you
References [1]

References [2]


References [3]