Capture optimization for composite images

Henry Dietz and Dillon Abney;
University of Kentucky; Lexington, Kentucky

Abstract

A composite image is an image created by combining portions of multiple separately-captured images. Stitching of captures of tiled portions of a larger scene can be used to produce a single composite image (a panorama) with a wider view angle and higher total resolution. Image stacking is a different type of compositing, in which the scene is not changing significantly across captures, but camera parameters might be systematically varied. Focus stacking can extend the depth of field, aperture stacking can implement apodization shaping the out-of-focus point spread function, and noise and motion reduction can be accomplished even using the same camera parameters for each capture to be stacked. These and other compositing methods are well known and commonly used, but the same fixed pattern is commonly used for ordering of captures and choice of capture parameters. This paper examines the problem of static, pseudo-static, or dynamic determination of the optimal capture parameters and ordering.

Introduction

Electronic image sensors have been steadily improving. Not only have high-end camera sensors dramatically improved in nearly every way, but cameras with small, but decent quality, sensors can be purchased for as little as a few dollars. However, many photographic tasks still cannot be cost-effectively accomplished using just a single exposure. The current work examines the problem of ordering the component captures within creation of a composite image. Toward this goal, it is useful to divide composite images into two broad categories.

In the first group, each technique produces a **spatially-tiled composite** image. Significant physical or logical motion occurs between exposures so that different areas within the scene are sensed, and it is the sequencing of those movements that matters most:

- **Panoramas**: the entire camera is moved, ideally by rotations about a nodal point[1], in order to produce a stitched composite image with a wider view angle and higher resolution
- **Scans**: the sensor is moved in the image plane, typically within a flat-bed scanner or a large-format camera[2][3], to produce a stitched image simulating what a larger sensor would have captured
- **Electronic shuttering** and **single-pixel cameras**[4]: the sensor is not physically moved, but the set of pixel locations being sampled is varied over time; although electronic shuttering is usually described as "rolling" from one edge to the opposite edge of the sensor, other sampling sequences are possible[5]

The second group of composites primarily concern not selection of the region to actively sense, but sequencing of capture parameters for sequential exposures covering roughly the same area. There are many variations of **temporal composites**, but the key characteristic is that the sequencing decisions involve parameters other than selection of which area within the scene is being sensed.

- **Bracketing**: this is a sort of degenerate compositing in which the "best" of multiple captures, with either some exposure parameter varied or variations due to unstable grip of the camera or changes in the scene, is selected
- **Stacks**: the sensor and lens are held in approximately the same position over multiple exposures which are precisely aligned and then combined; this stacking may be for:
  - **HDR**: High Dynamic Range imaging, combining multiple images to extend the light-to-dark range of scene brightnesses recorded[6]
  - **Noise reduction**: combining multiple images to improve the SNR (Signal to Noise Ratio) by averaging[7]
  - **Super-resolution**: combining multiple images aligned at the sub-pixel level to synthesize higher resolution[8]
  - **Blur reduction**: combining the least blurry portions from multiple images to construct a complete image in which there is less motion blur from moving scene content
  - **Focus**: combining the sharpest portions from multiple images to construct a complete image in which there is an extended depth of field[6]; making an "all in focus" image
  - **Apodization** or **Aperture**: combining multiple images captured with different aperture settings to approximate apodization in which the out-of-focus point spread function (OOF PSF) is bright in the middle and smoothly darkens toward the edges (this is what Minolta called "Smooth Trans Focus emulation"[9])

The current work is thus divided into two main sections: one about optimally ordering captures for spatial tiling and second about sequencing temporal composite captures.

Walk Ordering for Spatial Tilings

The optimal ordering of component captures for a spatial tiling is the problem of finding a walk order, which is a fundamentally difficult problem. If the optimality metric is minimizing total distance traveled or time spent traveling, visiting each spatial tile position once is essentially the definition of the classical "Traveling Salesman problem" (TSP)[10]. However, the optimality is really driven by two separate constraints, only the first of which is a classical TSP:

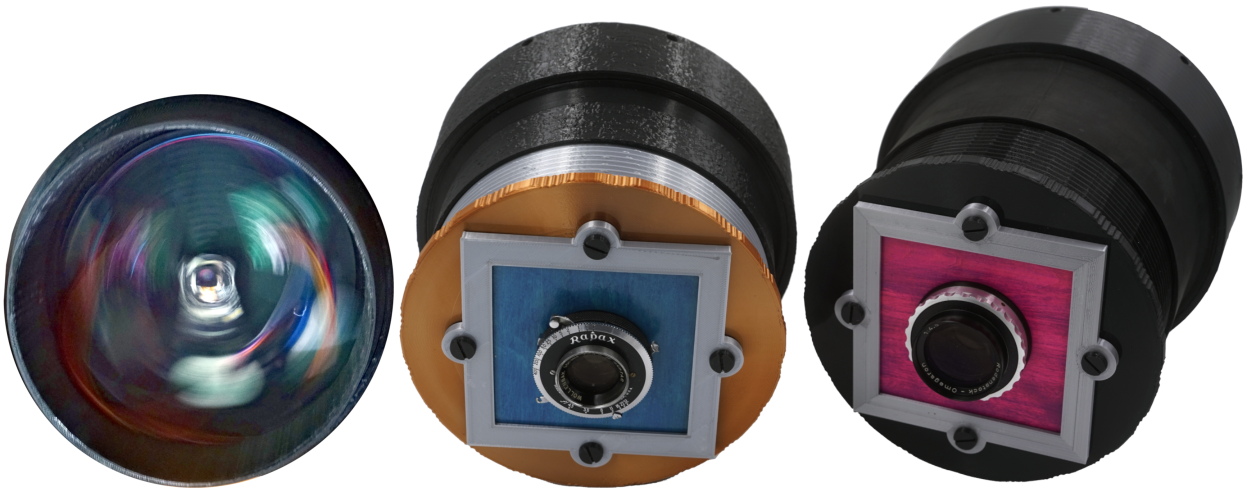


Figure 1. *Lafodis (LArge FOrmat DIgital Scanning) cameras use stepper-driven 3D-printed herringbone-gears for angle-radius motion control*

- The efficiency of implementing the walk order using the hardware motion abilities; for example, the time taken to move from one position to another is often a function of the sample position coordinate differences: how long does it take stepper motors to move from one position to the next in the walk order?
- The spatio-temporal consistency of the scene; many scenes being scanned are not entirely static, but contain elements that change over time such that spatially adjacent samples that are temporally distant (captured at significantly different times) may result in inconsistencies in the stitched image

Suppose that the scene is completely static. Using a standard X-Y motion drive system, the time to move from one position to the next can be trivially minimized by using any walk order in which only a unit step is taken to move from one position to the next. While a conventional raster scan does not meet this constraint, a bidirectional raster scan does, and so do many space-filling curves. The obvious argument against such a scan order is that some mechanical motion systems suffer "play" which would cause inaccurate positioning each time travel direction is reversed, but this play can be compensated for either mechanically (so-called "zero-backlash" systems) or computationally. An optimal static walk order easily can be pre-computed.

Even more novel motion-control systems easily yield optimal static walk orders. For example, Lafodis[3], shown in Figure 1, is a very inexpensive LArge FOrmat DIgital Scanning camera that can capture 160mm-diameter images with up to 2.6GP resolution, or 4x5" at around 1GP, by scanning the image plane with a 2MP ESP32-CAM sensor. Lafodis takes advantage of 360° angle $\times$ 100mm radius motion control to fit the drive system within the camera without loss of scannable area; X-Y drive would have required increasing the camera volume by roughly a factor of $2\times$ to make clear space for the drive system.

A spiral walk order would be fairly efficient for using Lafodis to capture a completely static scene, however, it would not necessarily be optimal. This is because the rotational step rate of Lafodis is dependent on the angle moved through rather than the distance, so the positioning rate in the innermost portion of the spiral would be limited by angular step rate. One time-optimal pattern would be a many-pointed star-like walk order making most steps in the radius axis rather than angle changes. In practice, the ESP32-CAM captures are slow enough that the motion control delay often does not dominate scan time, and even simulating a bidirectional X-Y raster scan is practical.

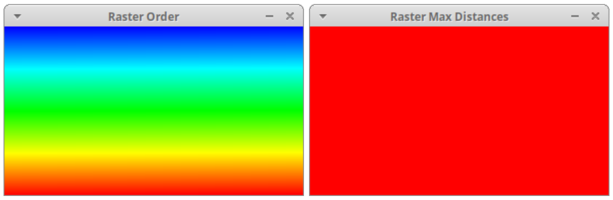
Spatio-Temporal Properties of Static Walks

By far the more complex and interesting case is when scene content may be changing during the scan. Let us begin by assuming we are interested in pre-computing a static walk order that minimizes probability of significant scene object motion artifacts. Let us further assume that sampling any one tile position takes 1 unit of time independent of the travel distance to that position – essentially assuming that the image capture is significantly slower than the motion control system.

To minimize these artifacts, the walk order used should have the property that the temporal distance between sampling spatially neighboring positions should be as small as possible. For example, a sample location within a rectangular grid of samples to be composited has eight immediate spatial neighbors. If that location is sampled at time t, the neighbors could be sampled within a window as small as $t \pm 4\delta$ where δ is the time taken for one sample, making it highly unlikely that the scene content will have changed enough to cause a visual discontinuity in the stitched composite image. If one of neighbors occurs at $t + 10000\delta$, the probability of a scene-object-motion discontinuity is much higher. Thus, the probability of an objectionable motion artifact is dominated by the maximum temporal distance to sampling of a spatial neighbor.

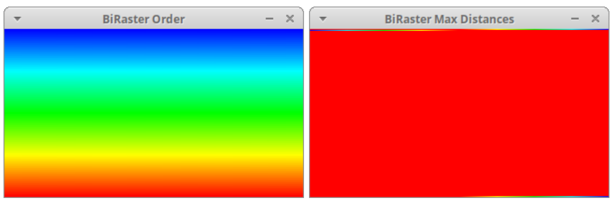
A program was written to construct and evaluate walk orders. For this paper, the examples will be limited to a 320×180 tile scan – a modest number with the popular $16:9$ aspect ratio.

Raster scan patterns are by far the most commonly used, yet they do not have very favorable temporal properties. Figure 2 describes the walk order for a standard X-Y raster. In the left image of the figure, each pixel represents one tile, and the colors are assigned based on position of the tile in the walk order, going from blue (first) to red (last). In the right image, the color represents the maximum temporal distance to sampling of an immediate neigh-



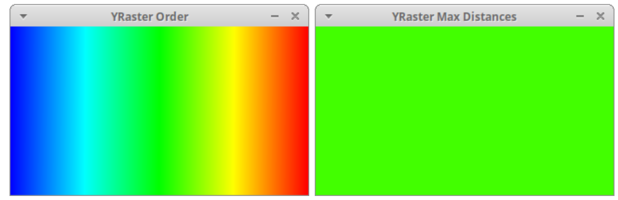

```
for (y=0; y<YMAX; ++y) for (x=0; x<XMAX; ++x)
  sample(x,y);
```

Figure 2. *X,Y raster scan order; avg. distance 240, median 320*



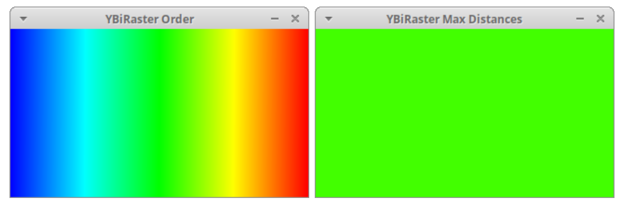

```
for (y=0; y<YMAX; ++y) for (x=0; x<XMAX; ++x)
  if (y&1) sample(XMAX-1-x,y); else sample(x,y);
```

Figure 3. *X,Y biraster scan order; avg. distance 240, median 162*




```
for (x=0; x<XMAX; ++x) for (y=0; y<YMAX; ++y)
  sample(x,y);
```

Figure 4. *Y,X raster scan order; avg. distance 135, median 180*




```
for (x=0; x<XMAX; ++x) for (y=0; y<YMAX; ++y)
  if (x&1) sample(x,YMAX-1-y); else sample(x,y);
```

Figure 5. *Y,X biraster scan order; avg. distance 135, median 180*

bor tile, scaled from blue (1) to red (for any delay greater than the maximum dimension of the raster, 320 in this case): colors closer to red thus indicate tiles likely to suffer scene object motion discontinuities. Simplified code implementing the scan order is also given.

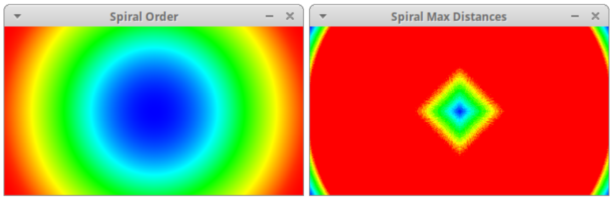
Figure 3 shows that, although the median temporal distance is improved over a unidirectional raster scan by using a bidirectional raster scan, each tile is still very likely to suffer from motion artifacts. Given that the Y dimension is smaller than X, it is not surprising that turning the raster walk pattern 90°, making the scan Y-X, helps by about the ratio between `XMAX` and `YMAX`. Figures 4 and 5 show that both unidirectional and bidirectional rasters have significantly better temporal locality – but the green in the distance images is only about twice as good as the red in the images for X-Y rasters.

Another walk order often discussed, and mentioned in the context of Lafodis above, is a spiral scan. A spiral walk order can be created efficiently by applying Bresenham's algorithm for efficiently drawing a circle[11] for a sequence of increasing radii, as described in Figure 6. The resulting pattern may be interesting, but only a tiny fraction of the pixels at the center have all spatial neighbors also temporally near. The "onion" walk order[12] is closely related, but as shown in Figure 7, is formed by a sequence of concentric square walks. It has poorer temporal locality properties than the spiral. Note that both of these patterns naturally fill areas of shapes that are larger than the rectangular array of tiles to sample; the `mustvisit(a,b)` function tracks this by returning false if position `(a,b)` is either out of bounds or has been sampled before.

A variety of space-filling curves have been studied and used to produce memory layouts for computer data structures with desirable locality properties; they can serve the same purpose here. The Z-order, also known as the Lebesgue or Morton curve[13], is easily constructed by interleaving the bits of X and Y coordinates and visiting X,Y positions in the order of the interleaved values. As should be immediately evident from Figure 8, the Z curve walk yields dramatically better temporal locality of spatial neighbors for most tiles, with poor locality tiles in a grid pattern. However, the Z-order curves are based on filling square areas with sides that are a power of 2 in length. Rather than simply clipping the curve, the Z-order coordinates can be scaled to fit the rectangular tile arrangement. We call this walk order "DivZ," as shown in Figure 9, and overall locality is slightly improved over the clipped Z-order walk.

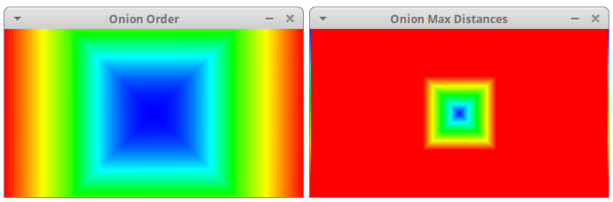
Like the unidirectional rasters, a Z order walk "hops around" – the next tile sampled is not always a spatial neighbor of the current tile. Not all space-filling curves have that unfortunate behavior. For example, the Hilbert curve does not[14], and can be efficiently computed as described in Hacker's Delight[15]. As shown in Figure 10, the Hilbert order always advances to a spatial neighbor of the current tile – provided the space being filled is a square with power-of-2 sides. This does minimize physical motion delay for X-Y drive systems, and it makes very low temporal differences more common, but the worst-case temporal differences are significantly larger than for the Z order. Scaling the space to produce a "DivHilbert" walk order, as shown in Figure 11, avoids a few large hops where the Hilbert curve is clipped (the bright red spots at the edges of Figure 10), but seems to make locality slightly worse overall.

In summary, of the patterns considered, it appears that one of the space-filling curve walk orders would be the best choice. However, taking the actual image capture and scan positioning times into account can result in other orderings being slightly better. In fact, the program constructed to perform the above tests implements a Genetic Algorithm[16] that attempts to evolve somewhat better orderings for the cost function given. The GA's initial




```
let cx=XMAX/2, cy=YMAX/2;
for (r=0; r<=sqrt(XMAX*XMAX+YMAX*YMAX); ++r)
  for (each (x,y) on a circle of radius r)
    if (mustvisit(cx+x,cy+y)) sample(cx+x,cy+y);
```

Figure 6. *Spiral scan order; avg. distance 298, median 305*




```
let cx=XMAX/2, cy=YMAX/2;
for (s=1; s<=max(XMAX,YMAX); s+=2)
  for (each (x,y) on a square of side s)
    if (mustvisit(cx+x,cy+y)) sample(cx+x,cy+y);
```

Figure 7. *Onion scan order; avg. distance 319, median 346*

population of potential solutions includes all the above orderings as well as randomly-generated orders. The improvements made were generally small, but some improvement was obtained for every case tried.

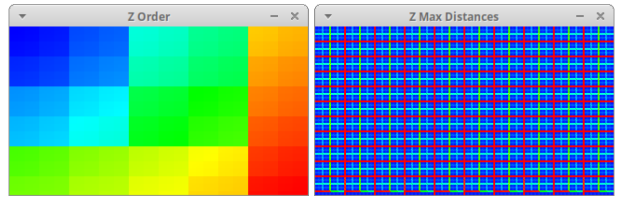
Pseudo-Static Scene-Dependent Walk Orders

In addition to purely static walk orders, it also is possible to use a small amount of scene overview or summary information to drive creation of a static order designed specifically for the current scene. This summary information can be obtained either by a sparse/lower-resolution scan, or by using a secondary camera to capture the full view with lower quality.

A portion of the scene which lacks detail is inherently less sensitive to changes in the scene. For example, tiles within a featureless blue sky have virtually no penalty associated with poor correlation of spatial and temporal locality. The same would be true of a scene region for which a sequence of two or more quick summary captures revealed no signs of scene object motion. Tiles in such regions can be omitted from the walk order computation and optimization, to be added-in using whatever order is convenient at the end of the performance-critical walk. It may even be feasible for low-detail tiles to be skipped entirely, substituting the corresponding portion of a summary image instead of sampling the tile.

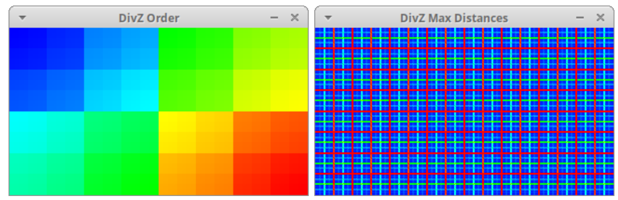
Dynamic Walk Orders

Beyond making a pseudo-static scene-dependent walk order, it is possible to dynamically change the walk order during capture. The goal is to detect inconsistency in real time and to schedule re-



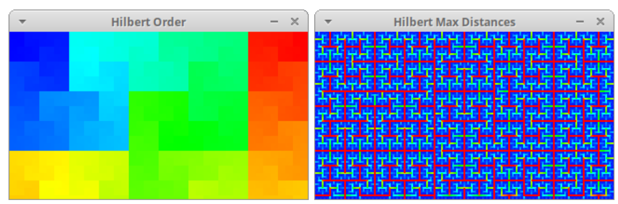

```
let b= smallest power of 2 >= max(XMAX,YMAX);
for ((x,y) in b*b Z order)
  if (mustvisit(x,y)) sample(x,y);
```

Figure 8. *Z curve scan order; avg. distance 198, median 15.2*



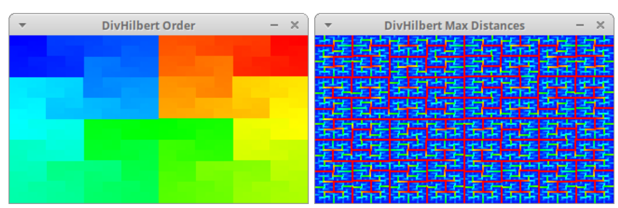

```
let b= smallest power of 2 >= max(XMAX,YMAX);
for ((x,y) in b*b Z order)
  let divx=(x*XMAX)/b, divy=(y*YMAX)/b;
  if (mustvisit(divx,divy)) sample(divx,divy);
```

Figure 9. *DivZ curve scan order; avg. distance 169, median 15.1*




```
let b= smallest power of 2 >= max(XMAX,YMAX);
for ((x,y) in b*b Hilbert order)
  if (mustvisit(x,y)) sample(x,y);
```

Figure 10. *Hilbert curve scan order; avg. distance 241, median 13.3*




```
let b= smallest power of 2 >= max(XMAX,YMAX);
for ((x,y) in b*b Hilbert order)
  let divx=(x*XMAX)/b, divy=(y*YMAX)/b;
  if (mustvisit(divx,divy)) sample(divx,divy);
```

Figure 11. *DivHilbert curve scan order; avg. distance 213, median 14.8*

sampling of the offending tiles.

The key to doing this the ability to incrementally add each tile image to an approximate stitch of the composite image – in

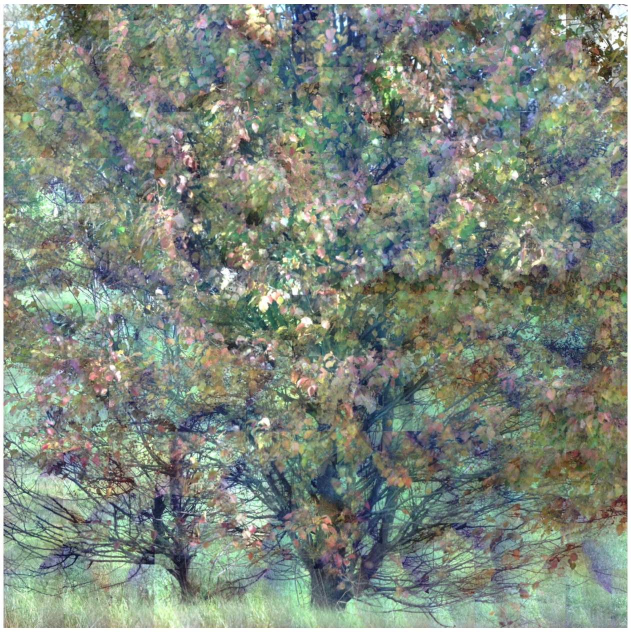


Figure 12. *Lafodis large format digital scanning cameras*

real time. High-quality stitching generally requires global adjustment of tile positions, but with approximate tile positioning it is possible to create a stitch good enough to discover where scene changes have been sufficient to create a motion artifact in the final stitched image. Senscape[17] provides a simple mechanism for real-time stitching based on modeling of uncertainty in fused sensor data, and we have implemented this for Lafodis, with an example real-time stitch given in Figure 12. A low-certainty tile placement should trigger scheduling of an additional sampling of that tile... which might in turn trigger resampling of that tile's neighbors if confidences drop for them too.

The tile re-samples could be given priority over continuing with the existing scan order, however, scanning is not a very fast process in comparison to the speed with which potential walk orders can be evaluated. Even use of a GA at runtime to modify the remaining walk order is feasible because the scheduling simply has to be fast enough to keep-up with the physical motion and sampling rates. When it is time to sample another tile, the GA can be paused and the current best choice of next tile accepted.

Temporal Composites

For temporal composites, although there is an ordering problem, in most cases the order of samplings is either unimportant or a good ordering is obvious. Ordering is not particularly important for stacking noise reduction, super-resolution, and blur reduction because the separate samplings are in some sense all equivalent. For bracketing or stacking HDR, focus, or apodization, the ordering is significant only in that the exposures with the most detail to integrate should be given priority and grouped together.

Consider apodization stacking. A crude example of how apodization stacking gives bokeh a smoother look is given in Figure 13; this example uses visibly too-large aperture steps, and the seven-bladed aperture shape is somewhat obvious, but it makes clear how the stacked exposures create a softer edge to the OOF

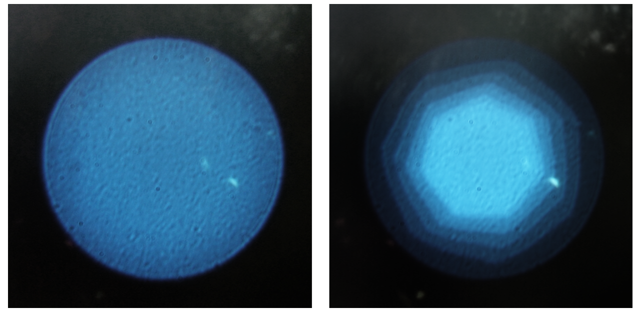


Figure 13. *Example apodization composite transformation of OOF PSF*

PSF. A better apodization composite using an $f/2.0$ lens might combine exposures at $f/2.0$, $f/2.2$, $f/2.5$, $f/2.8$, $f/3.2$, $f/3.5$, and $f/4.0$. The shutter speeds will be faster for wider apertures, and less will be sharply in focus. Thus, most scene detail is determined by the smaller apertures (larger f/numbers), and it would make sense to use a sample order that goes from $f/4.0$ to $f/2.0$ so that the exposures that contribute the least detail happen longest after the moment at which the exposure was triggered.

The more interesting decision for most temporal composites is not the ordering of a fixed set of samplings, but the pseudo-static or purely dynamic determination of the samplings to be ordered. For example, how many exposures at what apertures would be needed to generate a particular apodization transformation? The answer depends not only on the desired apodization and lens characteristics, but also on how far out of focus specular highlights in the scene are. Similarly, in focus stacking to obtain an all-in-focus image, both the minimum number of images captured and the choice of focus distances for them are highly scene dependent. As a simpler example, the number of samplings required for bracketing to select a non-camera-shake-blurred image really depends on how blurry the samples are: the sequence can stop as soon as a sharp capture has been made.

In sum, most types of temporal composites can benefit from using quick, approximate, dynamic evaluations of the samples as they are acquired in real time.

Conclusion

Although composite images are becoming increasingly common, the vast majority still use static sample orders that are not tuned to maximize the quality of the final image. For spatial tilings, we have shown that even careful selection of a static ordering can dramatically reduce the probability of serious spatio-temporal artifacts. Further, principles for creating pseudo-static and dynamic walk orderings were defined – including the concept of using one or more summary captures to formulate scene-specific priorities. It was further argued that many of the same concepts apply for temporal composites, however, the emphasis should be placed not on ordering, but on determining the number of samples and parameters for each.

It is possible to combine both spatial tilings and temporal compositing, for example, performing HDR scans. Perhaps most interesting is the fact that emerging sensor technologies can be thought of as inherently doing precisely that, treating each pixel as a tile, but without the need for physical motion to implement a spatial walk order. For example, TDCI (Time Domain Continuous Imaging)[18] naturally produces estimates for pixel values that can be translated into priorities from which a walk order for sampling pixel values can be determined: pixels that are unlikely to change value can be sampled less frequently, thus reducing the bandwidth required for readout and processing. Perhaps applying this same approach to readout of new SPAD-based or JOT-based sensors will produce important benefits?

References

[1] "GigaPan EPIC Pro User Guide," http://gigapan.com/, (accessed 1/31/2022)

[2] Rencay. http://www.rencay.com/, (accessed 1/31/2022)

[3] Henry Dietz and Paul Eberhart. "An Ultra-Low-Cost Large-Format Wireless IoT Camera," Electronic Imaging, Imaging Sensors and Systems, pp. 70-1-70-7(7), 2021 DOI: 10.2352/ISSN.2470-1173.2021.7.ISS-070

[4] Marco F. Duarte, Mark A. Davenport, Dharmpal Takhar, Jason N. Laska, Ting Sun, Kevin F. Kelly, and Richard G. Baraniuk. "Single-pixel imaging via compressive sampling," IEEE signal processing magazine 25, no. 2 (2008): 83-91.

[5] E. Artyomov, Y. Rivenson, G. Levi and O. Yadid-Pecht. "Morton (Z) scan based real-time variable resolution CMOS image sensor," IEEE Transactions on Circuits and Systems for Video Technology, vol. 15, no. 7, pp. 947-952, July 2005 DOI: 10.1109/TCSVT.2005.848338

[6] Qinchun Qian, Bahadir K. Gunturk, and Aziz U. Batur. "Joint focus stacking and high dynamic range imaging," Proc. SPIE 8660, Digital Photography IX, 866004 (4 February 2013) DOI: 10.1117/12.2003369

[7] Kalin Atanassov, James Nash, Sergio Goma, Vikas Ramachandra, and Hasib Siddiqui. "Temporal image stacking for noise reduction and dynamic range improvement," Proc. SPIE 8667, Multimedia Content and Mobile Devices, 86671P (7 March 2013) DOI: 10.1117/12.2008559

[8] Jing Tian and Kai-Kuang Ma. "A survey on super-resolution imaging," SIViP 5, pp. 329-342, 2011 DOI: 10.1007/s11760-010-0204-6

[9] "Custom 25-2 STF mode," in Minolta Dynax 7 Maxxum 7 Instruction Manual, Minolta Co., Ltd., pp. 180-181, 2000

[10] Gregory Gutin and Abraham P. Punnen, eds. "The traveling salesman problem and its variations," Vol. 12. Springer Science & Business Media, 2006

[11] Jack Bresenham. "A linear algorithm for incremental digital display of circular arcs," Communications of the ACM, Volume 20, Issue 2, Feb. 1977 pp 100-106, DOI: 10.1145/359423.359432

[12] P. Xu, C. Nguyen and S. Tirthapura. "Onion Curve: A Space Filling Curve with Near-Optimal Clustering," 2018 IEEE 34th International Conference on Data Engineering (ICDE), Paris, 2018, pp. 1236-1239, 2018 DOI: 10.1109/ICDE.2018.00119

[13] Guy M. Morton. "A computer oriented geodetic data base and a new technique in file sequencing," International Business Machines Company New York, 1966

[14] B. Moon, H. V. Jagadish, C. Faloutsos and J. H. Saltz. "Analysis of the clustering properties of the Hilbert space-filling curve," in IEEE Transactions on Knowledge and Data Engineering, vol. 13, no. 1, pp. 124-141, Jan.-Feb. 2001 DOI: 10.1109/69.908985

[15] Henry S. Warren, Jr. "Hacker's Delight Second Edition," Addison-Wesley, New York, 2013

[16] Darrel Whitley, "A genetic algorithm tutorial," Statistics and Computing, Volume 4, Issue 2, pp. 65-85, 1994 DOI: 10.1007/BF00175354

[17] Henry Dietz and Paul Eberhart. "Senscape: modeling and presentation of uncertainty in fused sensor data live image streams," Society for Imaging Science and Technology, Electronic Imaging, Computational Imaging XVIII, pp. 392-1-392-7(7), 2020 DOI: 10.2352/ISSN.2470-1173.2020.14.COIMG-392

[18] Henry Gordon Dietz. "Frameless, time domain continuous image capture," Electronic Imaging 2014, Proceedings Volume 9022, Image Sensors and Imaging Systems 2014 DOI: 10.1117/12.2040016