Model-Based SLAM

Acronyms and Terms used

- 3D: 3-dimensional.
- Albedo: The reflectance properties of a surface. How much light is emitted in each direction relative to incoming light. The most complete version is the bidirectional reflectance distribution function.
- COP: Center of projection.
- DOF: Degree of freedom. Pixel location in an image has 2 degrees of freedom. Position in 3D has 3 degrees of freedom. Orientation in 3D also has 3 degrees of freedom. DOFs: Degrees of freedom.
- FOR: Frame of reference. A 6-DOF location and orientation, perhaps together with a mapping (perhaps nonlinear) from one 3D space into another.
- FOV: Field of view. Refers to the angle of regard of a camera.
- Fovea: The central portion of the human field of view, where there is higher resolution vision than in the periphery.
- IMU: Inertial measurement unit. This might include measurements that are not strictly inertial in nature, such as magnetometer data, GPS measurements, and radio-based measurements.
- Mutex: Mutual exclusion primitive. Used to guarantee that only one thread of execution obtains access to an object at the same time.
- On-Center, Off-Surround: Description of the receptive field of neurons in the early visual system. This can be modeled as a Difference-of-Gaussians (DoG) filter, which has a positive Gaussian with narrower standard deviation concentric with a negative Gaussian with a wider standard deviation. It provides a comparison of local brightness with surround brightness, enhancing both edge and local-extremum detection.
- Saccade: Rapid motion of the eye's gaze between points of fixation.
- SLAM: Simultaneous Localization And Mapping: For a mobile camera platform, both determining the 3D environment in which it is moving and navigating that environment.
- w.r.t.: With respect to.

Approach

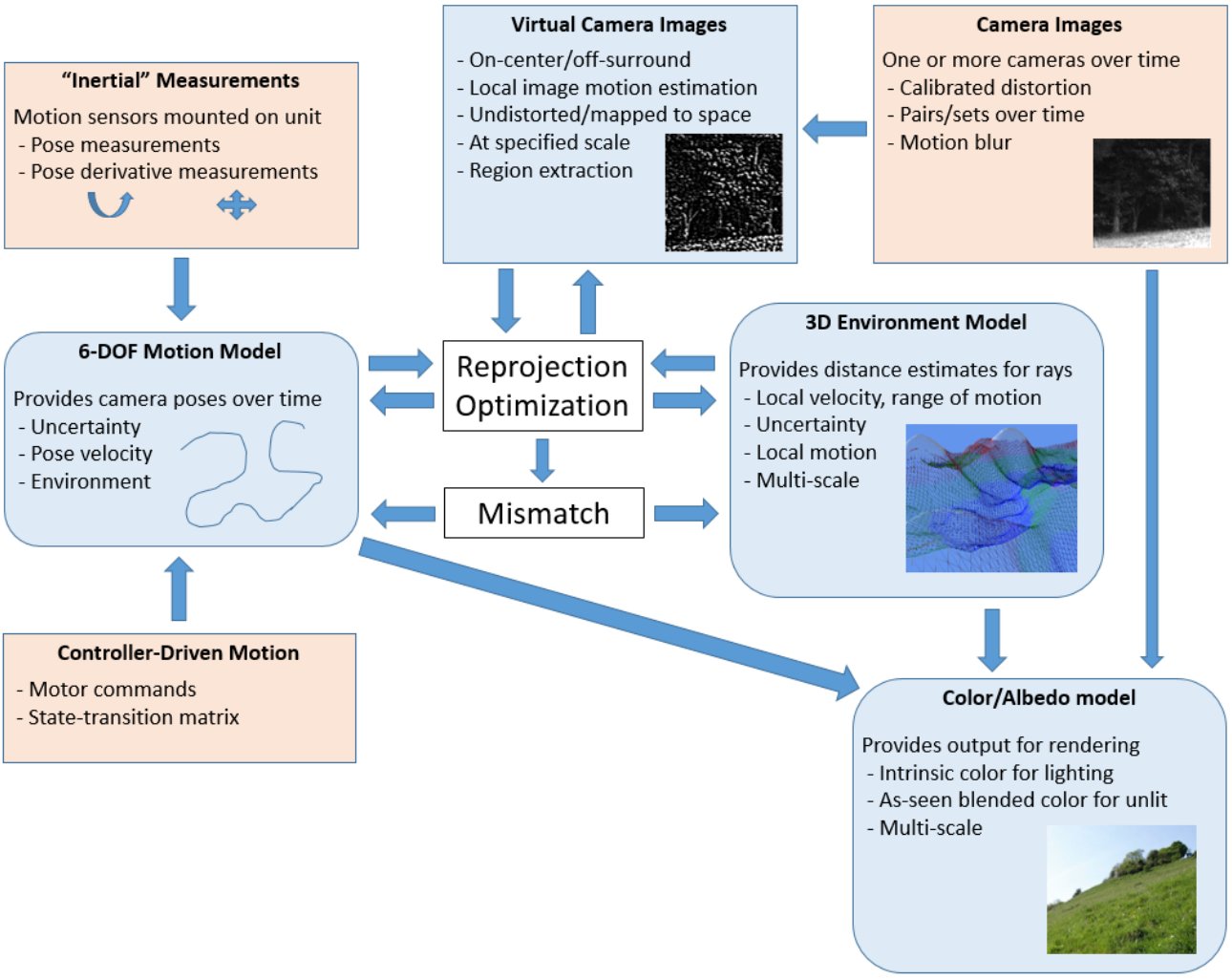
The basic approach is to construct a 3D model of the environment in which a set of one or more cameras might be moving and to use that model to estimate motion of the cameras and other sensors w.r.t. the model. The color and/or albedo of locations on the model might also be estimated, perhaps as a function of viewing direction; this intrinsic property determination might be done using a separate approach from the geometry determination. Both the geometry and the color of the model might be estimated at multiple scales. The model estimates might be improved over time based on multiple camera measurements, improved estimates of motion, and other data.

Model Initialization: The model might be initialized in a region of space using cross-correlation on sets or pairs of images or subsets of images (perhaps along with IMU-based camera-motion estimates taken around the same

time as the images) to estimate the radial distance of image locations from the camera's COP. These images might have center-surround processing applied to reduce the impact of noise, to reduce the impact of overall light level changes, and to highlight local differences in the image. Camera calibration might enable the determination of the direction of these locations in the camera FOR, producing a 3D location estimate for each image location w.r.t. the camera FOR.

The model for a region might also be initialized based on extrapolation from nearby model geometry. It might also be initialized based on interpolation between nearby or distant model geometry. It might also be initialized based on prior measurements or other prior knowledge of the geometry in the region including aerial photographs, survey data, previous LIDAR data, GIS information, and knowledge of the probability distributions of known objects (buildings, vehicles, animals, vegetation) in the region. It might also be initialized based on recent information from LIDAR, ultrasonic, or other time-of-flight or interference techniques.

Each location in the model might include estimates of motion, color, or albedo for that location. Each location might include error estimates (estimates of the accuracy of the estimates) or statistical distributions for all estimated quantities. Each quantity might be estimated by a robust estimator (median filter, non-uniform diffusion, bilinear filters, etc.).



Motion Estimation: Measurements of the IMU values and their integration over time might produce an initial estimate of the relative motion of the cameras between images. They might also produce estimates of the real-world position and orientation of the cameras when each image was acquired.

Model/Motion Optimization: Once a model is available in a region, it (or its reprojection into the camera FOR based on the motion model and camera distortion) might be compared against a recent camera image (or a subset thereof) to determine which regions in the image are well explained by the model and which are not. The model might also be compared against LIDAR, ultrasonic, or other data. The model might be maintained so long as there is not strong evidence against it (potential occlusion and loss of view might not be considered such evidence).

Adjustments to the 3D model and motion estimates might be made, producing new reprojections that improve the overall image/model match. This might be done iteratively or statically (using bundle adjustment, Kalman filters, or other approaches) to jointly or separately improve both the model and the motion estimates.

The model might also be optimized by fitting to prior measurements of the region or other prior knowledge of the geometry.

Model and motion estimation might be accelerated by determining which image regions are not well-explained (or are less well-explained) by the model and concentrating optimization on the model associated with those regions. Also, the use of a subset of regions that are well explained (perhaps widely-separated regions) might enable more rapid optimization of motion estimation. Estimates of the uncertainty in model regions might be used to concentrate calculations on those regions that need improvement while also identifying regions that are trustworthy for use in optimizing other system parameters and estimates. Sensor noise estimates (perhaps based on local image intensity) might be used to provide estimates of the likelihood of particular measurements given the model, and that uncertainty apportioned out to the various model and motion estimates.

Scale: All of the above approaches might be taken at multiple image and/or model scales. In this context, scale has at least three meanings: **spatial scale** of the distance between locations under consideration, **blur level** of the image or model locations being examined, and **detail level** of the model or image locations being examined near locations under consideration. @todo Insert diagram.

Insights and Inspiration

The human visual system operates in the context of an inertial understanding of local viewpoint changes, usually with two eyes at known disparity, in a perceptual system whose (quite significant) distortions have been modeled based on observations through these same systems.

The human visual system performs center-surround filtering on images early in the visual pipeline, which reduces the impact of overall scene brightness on local texture feature strength, highlighting edges and local bright/dark regions.

The human visual system operates based on a strong model of the world, using that model to fill in gaps in perception across most of the field unless there is a conflict. Information that does not match the model draws attention (and/or saccades) to update the model.

The human perceptual system appears to use maximum-likelihood estimation to resolve conflicting information (inconsistent over time or between modalities).

The human visual system has two separate paths for shape determination and color determination. After the 3D shape and 2D image regions are determined by the "where" system, the color of each region and object is pasted in by the "what" system.

The human visual system operates in massive parallel at multiple scales simultaneously.

The human visual system does not attend to the vast majority of information coming in at any given moment.

The human visual system operates without a common clock, with different scales operating at different speeds.

The human visual system keeps track of a small number of specific moving objects together with group/bundling of sets of objects that share common statistical behavior.

The human visual system performs most shape determination outside of the fovea, so at a level of blur. The fovea region is too small to do large-scale shape determination. There are two large-scale images (stereo vision) available for most people. Blur (model scale) decreases further from the center of attention. The fovea is drawn to nearby regions one at a time to obtain necessary detail. **Consider** doing most shape determination at coarse scale, using optical blur or pixel summation to avoid aliasing.

Earlier approaches attempt to search for individual features or feature matches. The current approach is designed (but not limited) to operate in an environment that is feature-dense across scale at most locations that need to be tracked. It is designed (but not limited) to operate in an environment where the cameras are moving, and where their motion is measured and/or estimated.

Estimation of the egocentric motion of the camera enables constraining the search for image-patch matches between images to 1D curves between images, reducing the search space from two spatial directions and one orientation. This can be used to greatly accelerate calculation. Using this together with a 3D model that might include error and/or uncertainty measurements can enable further reduction of calculations and can be used to select a subset of possible calculations that greatly improves the estimates.

Using a set of models, each of which is largely close to correct, enables rapid local updates of each model based on mostly-correct information from the others.

Benefits

Including measurements and modeling into a unified framework as described here enables more efficient use of computational resources, which can decrease energy use and/or decrease the time taken to produce models. This forms a "cycle of goodness" – where reduced solution time means reduced time between measurements, which means reduced change between measurements, which means reduced computational resources, which repeats the cycle.

Including error estimation and statistical motion distributions provides more-robust models, which enable more reliable planning based on the resulting models. It also enables additional focusing of computational resources on the most-likely solutions and/or reducing the largest errors, which can both improve the model and make more efficient use of computational resources.

Providing model-based estimates also increases the effectiveness of data transmission from a drone during deployment, enabling algorithms, a pilot and/or analysists to determine where the drone should acquire new data before returning to base. Providing a set of drones that cooperate to build a model could enable a set of analysts to survey an area more rapidly, enabling faster planning of responses in disasters and more effective and rapid activities in other situations. Model-based estimation also makes more efficient use of deployment time even in the absence of communications back to base.

Details

Center-surround processing: Image analysis based on center-surround processed images provides several potential improvements over analysis based on raw images, as seen in the figure below and described in the bullet points.

Figure Center-Surround: ***Left:*** Raw intensity image of trees at the edge of a field, with brightness scaled so that no pixels are saturated. ***Right:*** On-center/off-surround image with on radius of 1 pixel and off radius of 2 pixels. Local texture is more visible across both the bright and dark regions of the image, and are less dependent on overall brightness level.

- Initial center-surround filtering might be applied to all images before they are fed into the image matching, geometry-processing, or other calculations.
- This might reduce the impact of absolute lighting level on the image analysis and comparison calculations.
- This might reduce the impact of image noise for center filter sizes that are larger than a pixel.

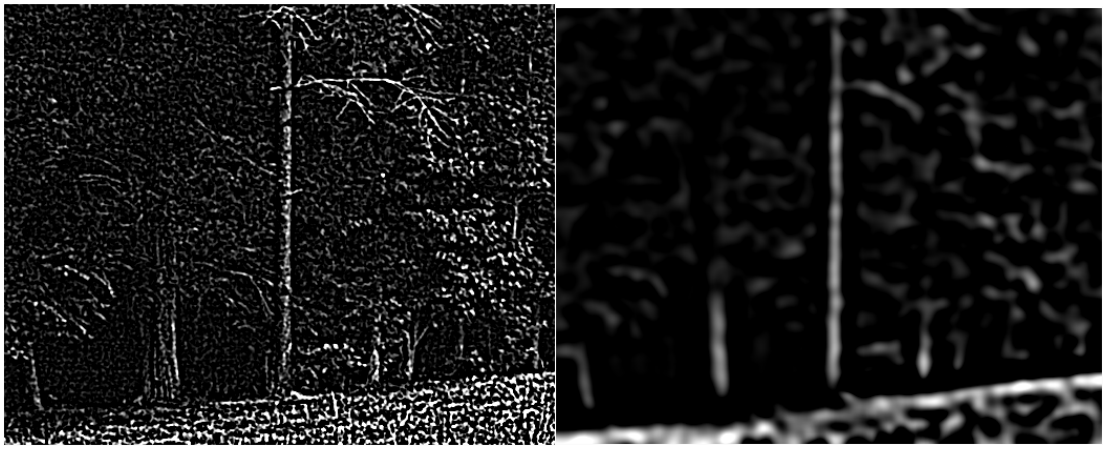


Figure Scale Adjustment: ***Left:*** On-center/off-surround image with *on* radius of 1 pixel and *off* radius of 2 pixels. ***Right:*** *On* radius of 5 pixels and *off* radius of 10 pixels. The mostly-vertical tree trunks and mostly-horizontal field boundary stand out, reducing the potential matches.

Scale-space operation: There are a large number of local texture features densely packed within images of natural scenes. Image alignment based on local shifts can get stuck in local minima due to the large number of potential matches in a region. Applying appropriate scale-space filtering and doing initial matching at coarser scale and then tracking this match to smaller scales can enable the global optimum to be achieved more efficiently than a brute-force search of all image alignments, as seen in the figure above and described in the following bullets.

- All processing might operate across two image scale dimensions: (1) Number of sample patches per image, (2) blur level of the sampled patches and across the spatial scale dimension. Patches might be selected using one of the "Pick me!" criteria described below, they might be selected uniformly, or they might be selected at random or by other approaches.
- Scale-space tracking (starting at large scale/blur, then coming in finer and refining the estimate) of image-patch matches might be used to determine detailed registration that is based on both large-scale and small-scale image features.
- Performing comparisons and calculations at a coarser scale might reduce the amount of calculation, whether the scale change is provided by computational, optical, mechanical, or other approaches.
- GPU-based or other parallelism might be used to efficiently compute coarser-scale images based on finer-scale images, aggregating metrics as appropriate (summing, averaging, Gaussian blur, range of values, statistical variation measures, etc.).
- The approach might schedule calculations across scale in images based partially or wholly on information from the image itself (such as the presence/absence of contrast or texture at a given scale).

Virtual cameras: Many of the image analysis operations can be embedded in the camera, either physically (optics, analog or digital circuits, FPGAs, DSPs or GPUs affiliated with the data stream) or virtually (as a separate, software-controlled process that has an interface providing those functions as if they were part of the actual camera), providing a substrate on which to develop robust, effective, and efficient algorithms. These cameras would report filtered data, either in place of or in addition to image data. Some of the calculations that can be

embedded, and approaches to minimize communications between tightly-coupled memory-processor subsystems, include:

- **On-center, off-surround** processing might happen on the camera. This might be done using resistor networks between the pixels, adjusting the amount of blurring by switching on and off. It might be done using digital mixtures, either on the chip or during scan-out. Two blurs of different amounts might be subtracted in analog on the sensor or this calculation might be performed digitally.
- **Image motion estimation** might happen on the camera, using local shifts and rotations of the data to locate better fits. This might be done using cross-correlation, mutual information, or other metrics.
- **Model mismatch** might be performed on a camera that has a model of expected motion that is applied before the local motion estimation is applied. This camera could report the extent of model mismatch across the image. The model mismatch may be based on pairs or sets of stored images, with no image data or features stored explicitly in the model.
- **Scale-space** processing might happen in the camera, in combination with any of the other operations.
- **Filtering** might happen on the camera, either based on thresholds relative to the extreme values or based on sorting, or based on absolute thresholds. Only values that exceed the thresholds might be transmitted to other stages or considered for further processing. Filtering might also be performed based on goodness of fit in a region, amount of contrast, or combinations of these and other metrics. This filtering might include mathematical morphology or other operations to include neighboring regions and/or to ensure closed regions for processing.
- **Model fitting** might be done on the camera, perhaps by fitting linear or higher-order functions to regions and transmitting the model and a boundary description rather than all of the values within a region. This might be done by providing structured or unstructured sets of vertices that have values which are interpolated to find values between them.

Model-based operation: The approach might include models for drone motion, camera distortion, 3D static objects, small-scale object motion, large scale object motion, environmental forces and state (wind velocity), and other things. These models can be individually or jointly optimized and improved over time based on new and historical measurements. The model updates for each might be done asynchronously and at different rates. So long as the models explain the images within acceptable tolerances, they might not be updated. When a region of an image is not well-explained by the model, this might cause an update one or more of the models. The amount of error and/or uncertainty in regions or portions of models might be used to determine which models are updated and how frequently.

- **Drone motion model:** The drone motion model might include both present state (pose, pose velocity, pose acceleration) and history of the state of the drone base and both present and historical state (angle, angular velocity) of the set of cameras attached to the drone. The drone motion model might be based on a number of things:
    - **How the drone is trying to move:** Based on commands sent to motors and model of how the drone responds (state transition matrix). The state transition matrix describes how the current state of the drone changes over time based on its earlier states and based on forces provided by its control surfaces and motors, and based on environmental state and forces.

  - **Non-imaging sensor measurements:** Based on sensor pack (accelerometer, compass, GPS, angular rate gyros and/or other sensors). This is a set of measurements that each describe a portion of the state, including pose and pose derivatives. This might also include measurement of external forces.
  - **Image measurements:** Based on image response compared to expected texture, geometry, and motion blur. This is a comparison between (1) the images expected based on the drone state and model and (2) the images seen by the set of cameras. It might take into account the expected blur due to relative motion over the exposure time of each camera.
- **3D environment model:** The 3D model of the environment that the drone is operating in might contain both stationary (static) and moving (dynamic) portions. The dynamic portions might include motion that is similar over time (water flow, etc.), small motion around a fixed location (leaves, blowing grass, etc.), and more general object motion (moving vehicles, pedestrians, etc.). The same region of space might be considered to be static at one scale and dynamic at another.
  - A Kalman filter or other estimator might adjust geometry and texture to make improve or make optimal the model estimation using camera images and measured flight path as inputs.
  - The model might estimate lateral motion at each location in the image model to enable tracking of moving water and dynamic motions, and to improve drone motion estimation and camera calibration.
  - The spatial model might use object constancy over time to enable model construction for objects viewed through trees or other partial occlusion. It might use the presence of consistent texture matches within a region of space that persist over time and over motion even in the presence of foreground elements that partially occlude from one or more viewpoints. This model might be restricted to image regions with sufficient texture contrast to discriminate between good and poor alignment.
  - This is more powerful than simple correlation and might provide a framework in which to do robust estimation of consistency between views – retrying models over a set of images rather than just pairs.
    - The use of a comparison metric that is robust to changes in color due to view-dependent lighting, such as Mutual Information, might help with this.
    - The use of only spatial occupancy information and comparing current images only against recent image sets rather than stored imagery might help, and poses little restriction because it deals with any new imagery that is being acquired.
  - The model might include classification of spatial locations into different material types (grass, trees, dirt, mud, water, cloud, sky, etc.) and the expected behavior of these materials might depend on the state of the environment (sunny, cloudy; windy speed; raining, foggy; etc.).
  - The model and its processing might be based on center-surround imagery across scales, and not on actual image intensities. The 3D model might not include local intensity information at all, but might only serve as a description of how locally-changing image from different cameras FORs should relate to one another in stored and ongoing new image streams.
- **Color/albedo model:** The intensity, color, and/or albedo of the 3D spatial model might be determined separately from the geometric model production itself. Particular raw images (before center-surround processing) might be projected onto the spatial model. They might be adjusted based on expected

lighting levels. They might be contrast enhanced. Histogram equalization, local brightness normalization, or other methods might be used to produce images that are pleasing to the viewer but which do not reflect actual directly-measured intensities. This model might be separate from the intensity values stored for the purpose of 3D geometry determination.

- This model might be reconstructed based on time and space, projecting the imagery closest in time to the desired moment that was taken from the direction that is most consistent with the desired viewing direction onto the 3D geometric model. This might not store a consistent 3D colored model, but reconstruct it on the fly from subsets of the stored data.
- This model might be reconstructed based on depth pixels, where each pixel in the reprojected image(s) is assigned an estimated depth. Interpolation might be used to fill in the regions between pixels centers.

Camera configuration: There might be a wide-FOV lens for navigation, perhaps focused at a depth different from the viewed objects to provide optical convolution to the desired pixel integration size. There might be a set of survey lenses for observation of details.

- Each camera in the set of cameras within the system might have a separate model of its characteristics, which might include: radial distortion, global and per-pixel sensitivity, global and per-pixel noise distributions, scan characteristics (progressive, interleaved, full-frame), etc.
- Different cameras in the system might have different resolutions, zoom levels, exposure times, frame rates, etc. and these might change over time for a given camera. Some cameras might include optical blurring elements, coded apertures, or other optical structuring elements to help reduce computation or increase sensitivity to desired measurements.

"Pick me!" selection of regions to work on from each set of images, or on each available computational platform. The scheduling of available resources (cameras, image processing, communications, memory) might be done in such a way as to focus them on regions that are more likely to reduce model errors. This scheduling might be based on criteria including one or more of the following. They might be used to produce masks on virtual cameras to limit data transfer and/or computations to regions of interest. The might be used to increase the priority of scheduling for a new image acquisition from the corresponding real system cameras and/or to schedule image analysis algorithms to produce data from virtual cameras.

- High model mismatch regions.
- High-noise regions.
- High-uncertainty model regions.
- High-contrast/high texture image regions.

Reprojection of information in the model from 3D space onto an image for comparison might make use of the motion estimate and 3D model to take into account one or more of the following:

- Each pixel might be shifted by amounts proportional to uncertainty in the channel being used at that location, providing multiple potential matching candidate images. Neighbors might be shifted by similar amounts to maintain consistent local variation. If we are sampling a subset of the pixels in the image, we might move the entire patch around each by the same amount.

- Estimates for values of image regions seen in only one image (perhaps because they are beyond occlusions or other borders) might be filled in based on values from nearby pixels or model values. The uncertainty of such filled-in entries might include appropriately-increasing uncertainty based on their distances (in image space or 3D space) from entries with known values.
- Each pixel or region might be reprojected based on different assumed depths of the model for regions with multiple layers to determine their most-likely layer membership.
- Pixels or regions might be adjusted based on different motion models (reprojecting the entire scene or subset based on different transforms) to determine which motion model is a better match, optimizing the expected motion.
- Dynamic-scheduling selection of which reprojections to do for each image location might be based on the estimated relative error in estimation of model and motion near that location, scheduling computations to maximize the reduction in error from each reprojection.

The **image and model encodings** used by the system might include:

- Depth pixels might be an efficient way to store and operate on from a camera-centered FOR. This would be a polar-coordinates sphere centered on the camera, with virtual camera values and estimated depth per pixel.
    - This might construct a depth estimate for each pixel in each image, which is used along with the camera FOR estimated for the whole image to form a reprojection and not maintain a separate sampled model for the world.
- A mesh of triangles might be efficient for rendering.
- Image cross-correlation might be fastest to compute in image space, which is distorted w.r.t the model. Thus, a mapping from image space to image space or from model space back to image space might be used.
- To support semi-transparent pixels (seeing through out-of-focus occluders) or screen-door transparency (seeing through branches and other partial occluders), the encoding might include one or more of the following:
    - Layers: More than one depth might be associated with a pixel. More than one triangle might be associated with a given image region. More than one image might be associated with a given image region (overlapped in depth). Image layers might include transparency per pixel.
    - Wrapped mesh: Objects might be represented by polygonal meshes that are not functions, but which cover the object as it curves in space. This could produce meshes or sets of meshes that self-occlude from certain viewpoints.
- An ego-centric sphere surrounding the drone and storing depths might be used to maintain local model state. This might be updated over time based on estimated motion, providing an updated model against which to compare new measurements. This sphere might be represented as a set of points in space, as a polygonal mesh, or as images whose pixels have depth.
- Two representations, one supporting objects (moving or not) and the other supports terrain
    - Radial from the center of a sphere
        - Handles convex objects
        - Could also be used for terrain
        - Could also handle the view from the camera center.

Local normal approximation for texture projection: For representations that support local normal estimation (including at least meshes, triangles, polygons, and pixels with depths), this local normal estimation might be used to apply appropriate projective correction to pixels from the camera as they are filled into the image information for the local model.

High dynamic range imaging over scale:

- Virtual cameras might capture images at specified exposure and blur level, with the exposure and/or blur level of consecutive images being different. This provides a set of images of the same region with different exposures, such that some images will be appropriate for brighter regions and others appropriate for darker ones.
    - Provides images at multiple exposures and scale (blur level) for each region, providing a richer description and capturing details and image variation at the appropriate level for each portion of the region and each stage of image processing.
    - Masking of regions not needed at a particular exposure and scale can be used to reduce computations by focusing them only on the regions where they are applicable.
- Images with different exposures and/or blur levels can be scheduled to be taken by real cameras according to relative priorities, relative errors, relative uncertainty, model mis-match, round-robin, or other scheduling algorithms.

Parallelism between tasks:

- Separate execution threads or processes might perform different model optimization, filtering, and other operations, each using its own virtual camera. Each might fire when the appropriate image arrives and produce an output asynchronously, triggering other steps in a longer computation. For example, a blurring thread might operate on every fourth full-resolution image (or on the latest available image each time it completes), producing blurred and perhaps downsampled images at a slower rate than the full camera input rate. Each time it completes, a model-comparison thread might execute, comparing the blurred image against whatever is the current model at the time the image is available, and using the historical pose that corresponds to the actual time that the original image was taken. Each computation might include a time stamp describing what time its calculations are based on, and all measurements or model parameters interpolated to that time to enable consistent calculations.
- Asynchronous communication between the threads might not require lock-step data transfer, but enable transfer whenever operations are scheduled or when data becomes available. This might be accomplished using semaphores or using queued image/data buffers or based on a reconstructed continuous-time model. It might be done using a “latest buffer” that is copied to and from (using a mutex or other mechanism to avoid race conditions).

Multi-perspective views:

- Two-view baseline (stereo, from integrated acceleration) image-shift matching, perhaps after removing motion blur: The motion model can be used together with the 3D world model to determine which sets of images cover the same region of space.
- The outbound flight path and inbound path might be used to provide two widely-separated views to get a broad baseline on each 3D location. A serpentine path might be taken to provide baseline in the

forward direction to avoid obstacles. A Hilbert or other curve might be used to optimize multi-drone survey paths.

- Multiple cameras with different centers of projection might be mounted on the same drone, providing simultaneous multi-viewpoint images.
- Synchronized cameras on different drones, perhaps with each other's location visible because of a biased fisheye orientation (or >180 degree viewpoint), might enable simultaneous capture of two or more perspectives on the same 3D spatial region from different viewpoints.

Motion blur:

- **Forward modeling:** The amount of blur expected in a camera image is a function of the exposure time, the motion of the camera, and the motion of object or its texture during that exposure. This might be modeled in the image rendered from the model, either using pixel-space blurring kernels or multi-pass rendering with different projections. This would enable comparison of the model to the as-measured blurred image.
- **Estimating:** An improved estimate of the underlying un-blurred texture in an image before blurring might be obtained by applying deconvolution with a kernel that locally matches the expected image motion due to camera and object texture. This would enable comparison of the deblurred image in textured model space.

Methods for estimation of depth might include:

- Depth from model match.
- Depth from cross-correlation, mutual information.
- Depth from motion blur
- Depth from focus
- Depth from echolocation. The units could chirp once per second, in synchrony and at different frequencies, to determine how far they were from each other (speed of sound delays); directional speakers might help with orientation.

Potential computational acceleration techniques:

- On-center, off-surround image acquisition might be obtained through focus adjustment/optical blurring to avoid the computational cost of doing the multiple blurs. In this case, we'd acquire images at two different blur levels and computationally subtract one from the other to produce an image like that from the center-surround neurons in the early human visual system.
- Optical blurring (focus change) might also be used to avoid the need to computationally filter the image in scale space, producing am image with high resolution sampling but low spatial-feature resolution.
- Correlation calculations might be done most rapidly in image space, which is distorted w.r.t geometry or projection space. This might be enabled by producing an example image from the model that is distorted in the same manner as the camera image is distorted. It might be supported by computing correlation in image space and then providing an undistortion mapping of differential motion from image space to linear projection space.
- Operations might go only as far in scale space as is warranted given the expected distance that image patches might move over the time between measurements (or based on the expected mismatch

between model-transformed prior measurements and current measurements), limiting the number of levels that must be computed. Excessive mismatch at the furthest level might tag a region for further processing without causing the entire image or region to be processed at additional levels. Finding optimal results at a boundary condition might likewise trigger further processing, either in a region or across the entire image.

Synchronization:

- The relative timing of measurements from the cameras, IMUs, and other instruments in a system might be known by system construction or might be estimated. The estimation might be done by finding the time alignment that minimizes the difference between the differential motions predicted from each separately. It might be done by finding the alignment that minimizes an error metric or that optimizes a criterion. It might be done by finding the time alignment that minimizes the total motion estimation error among all sensors over a measurement period. It might be done by finding the alignment that predicts the minimum energy of motion.
- Sensors in the system might operate at fixed sampling rates, or there might be variation in their rates. These variations might be estimated by separately determining the appropriate alignment at different measurement periods, which might overlap or might be disjoint.

Flexible and efficient transmission:

- **Geometry:** Terrain height fields might be encoded in a multi-scale VRPN image, stored in a manner similar to a MIPmap, with powers-of-two or other scale reduction at each level. Only a portion of each image needs to be sent for any particular request or calculation, and higher-resolution data might overwrite existing low-resolution data because it can be downsampled on the far side of the communication channel. Implicit coordinates might be used in a regular grid to reduce the amount of spatial information required to send along with the height data.
- **Color/Texture/Albedo:** The color, intensity, or other visually-relevant features of the data might be encoded as a separate channel in the same MIPmap structure, or it might be stored in a separate, spatially-registered structure. It might be stored at a higher spatial resolution than the geometry data.

Noise:

- Expected noise (largely per-pixel Poisson) might be included in all of the estimations, adjusting the expected fitness, compared against measured contrast, and included in all of the calculations and optimizations.
- Noise might especially play a role in the estimation of uncertainty and in the specifications of expected distributions of intensity within the models.

Camera calibration might be performed or improved by observing the motion of stationary features or texture patches during motion that is tracked by IMU or other means. The relative phase or latency between camera and sensor measurements might be determined by studying characteristics (motion onset, motion magnitude, relative timing) of image-patch motion and IMU motion. This might be improved by taking into account known characteristics of the 3D distribution of objects and/or color distributions. This might be done in a situation where there is a fixed bright emitter and the camera's motion is constrained to be only rotation around its

center of rotation or another fixed axis. An assumption of radial symmetry in the distortion might be used to improve and extend the calibration to pixels where the emitter is not seen.

Potential Assumptions

Environments consisting of dense feature matches together with planar (water surface, wall) or infinite-distance (sky) regions.

Distortion correction is modeled, camera motion is estimated, and camera motion is present.

The environment being measured is largely static, with most objects moving at most small distances (leaves, grass).

Additional ideas

DRONOS (Drone Operating System): Like VRPN, but for drones. Time information. Error bars. Real-time LEWOS-like measurement and control. GPS synchronized clocks. GPU image processing? Real-time distributed across the whole system. Kalman filters. Common device interfaces.

Hardware synchronization: Between all measurements down to the microsecond level. To camera shutter trigger (both beginning of integration and length of exposure). If GPS, we could get down to the microsecond between units.

Tasks for drones:

- Survey
- Deliver food
- Communicate
    - Drone operator who speaks local language to survivors
    - Remote counselors talking through Skype or cell phone to survivors
- Locate survivors
    - Drones collect audio while flying and people might listen to the audio and report when they hear people.
- Image the dead for identification or other purposes.
- Deliver paper messages/flyers, maybe custom-printed on board the drone
- Provide a cellular network
- Swap out cell phones with on-the-ground personnel (new batteries, data)
- Have the drones land in a chain to provide increased transmission range for cell phone or other communications.
- Consider dropping low-power bluetooth transmitters as environmental sensors and detectors of people (audio?). They might form a network of sensors if dropped close enough together.
- Have a drone fly far afield, then land or drop a tether to tie down and then inflate a balloon to stay aloft for a while. Or even better, just deploy a balloon on a tether with whatever repeater is needed.

Rotor on the back aiming horizontally to improve efficiency of flying forward. Shape the drone body as a wing to help with this.

Acoustic 3D scanning for robust geometry avoidance.

Transport a pack of drones via blimp and launch, then bring back the blimp. Or tie them together in a chain like geese do to try and increase their range.

Figure Images Repeated

Figure Center-Surround (left)

Figure Center-Surround (right) and Figure Scale-space operation (left)

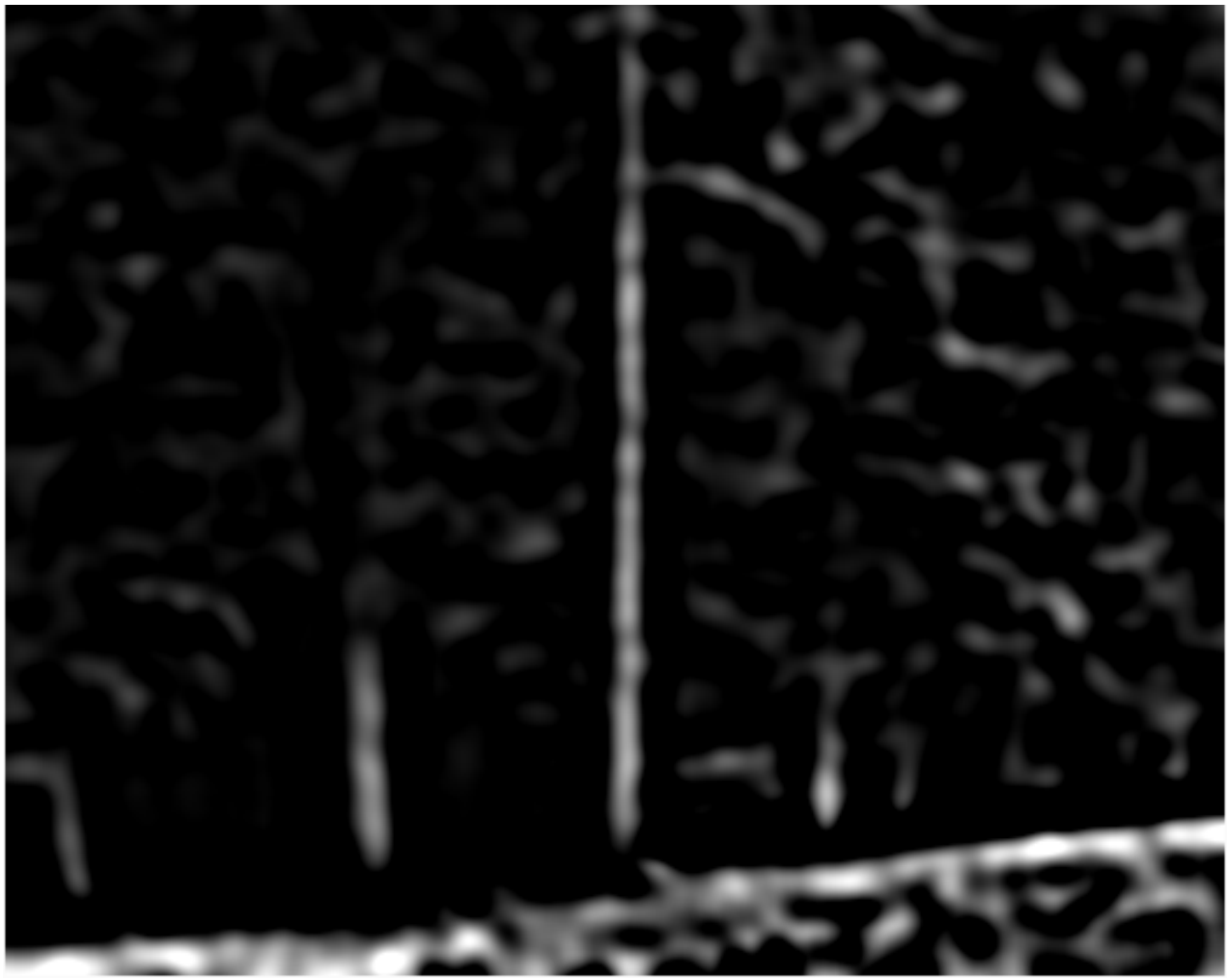


Figure Scale-space operation (right)

"Inertial" Measurements
Motion sensors mounted on unit
- Pose measurements
- Pose derivative measurements
6-DOF Motion Model
Provides camera poses over time
- Uncertainty
- Pose velocity
- Environment
Controller-Driven Motion
- Motor commands
- State-transition matrix
Virtual Camera Images
- On-center/off-surround
- Local image motion estimation
- Undistorted/mapped to space
- At specified scale
- Region extraction
Reprojection Optimization
Mismatch
3D Environment Model
Provides distance estimates for rays
- Local velocity, range of motion
- Uncertainty
- Local motion
- Multi-scale
Camera Images
One or more cameras over time
- Calibrated distortion
- Pairs/sets over time
- Motion blur
Color/Albedo model
Provides output for rendering
- Intrinsic color for lighting
- As-seen blended color for unlit
- Multi-scale