# A Universal Camera Model, Calibration and Applications

By Bingcheng (Arthur) Yang

May Solutions LLC
www.MaySolutionsLLC.com
ayang@MaySolutionsLLC.com

## Abstract and Outline

V0.99 2018-11-10

A camera consists of a CMOS/CCD image sensor and a lens. The lens optics makes the sensor generate an image for the scene and at the same time causes distortions to the image. Geometry distortions should be calibrated and be corrected to get a less distorted images. *Pixel Ray Vector Matrix* (PRVM) is a universal camera math model that is suitable for any cameras to describe the distortion errors and can be used to fully eliminate the distortions. The geometry property of all camera pixels can be described by the PRVM accurately, no matter the camera has a regular lens or irregular lens optics that cannot be modeled by any existing conventional math models. Compare to all other conventional camera modes, the major differences and advances of PRVM model is that the pixels in a camera are not treated as scalars anymore, but vectors. In PRVM, each pixel ray is represented by a unit vector in space (usually in the World Coordinate System, WCS). The position and the sensing direction of the pixel ray vector can be calibrated independently and accurately with reference light sources at 2 known WCS locations.

The *Intensity Property* (or photoelectric property) of each pixel can also be calibrated. It will be linearized individually and normalized all together globally. An intensity property equation for each pixel can be generated. An *Intensity Matrix* (IM) can be configured.

The PRVM matrix indicates the locations and the sensing directions of the pixels in the WCS. The IM matrix makes all these pixels have the same optical-electrical response properties. These 2 matrices make each sensor pixel to be an independent measurement unit. Depend on the application requirements, an arbitrary *Virtual Camera* can be created. An *Inverse Mapping Matrix* (IMM) can be derived for the Virtual Camera from the PRVM. All the *Virtual Pixel Ray Vectors* will be distributed in the WCS evenly or at any other expected patterns. PRVM and IM can be merged together to form an *Extended PRVM* or a *Generalized PRVM.*

*Pinhole Model* is successfully and widely used in camera modeling and applications. But with this model, the geometry errors could be big and may not be evenly distributed across the sensor or the images. The distortion errors could be up to a few or even tens pixels when a low cost lens is used, or especially toward the edge areas of the sensor. This model may not be applicable to cameras with special lenses (such as telecentric lenses). In comparison, the PRVM is model-less and assumption free. Each sensor pixel ray is independent to that of any other pixels. The residual errors are so small that they are mostly negligible. The error properties are statistically equal for all sensor pixels. Moreover, it can be converted to pinhole model or any other models easily. This makes the PRVM model be fully backward compatible with all previous software and applications. No hardware modifications are required to upgrade and migrate current existing systems and applications to use the PRVM model. To make any low cost Commercial Off-The-Shelf (COTS) camera to be industrial measurement level, all we need is only an LCD flat panel monitor and a linear stage (optional).

## Calibration Principle

It might be not feasible or impossible to illuminate an individual sensor pixel *pure physically* with an individual light source at known locations. But it could be doable and practical to do that *math-physically*. An LCD monitor with encoded *Phase Shifting Fringe Pattern* (PSFP) images can do the work perfectly. Fig. 1 to 3 depict the principle. Among them, the 3 fringe images displayed by the LCD and captured by the CMOS camera are all 8-bit monochrome grayscale. When introducing the basic principles, the color fringe images are only intended to more intuitively describe the phase relationship between the 3 monochromatic fringe images. In practical use, for color LCDs and color CMOS cameras, they should all be treated as monochrome.

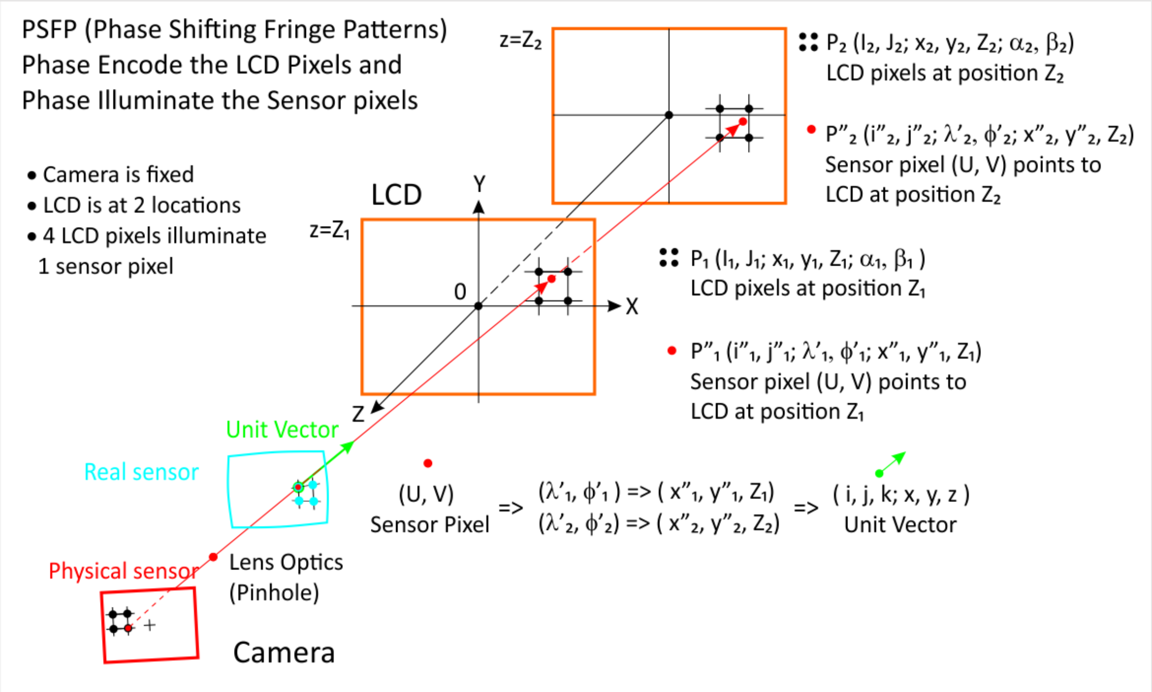
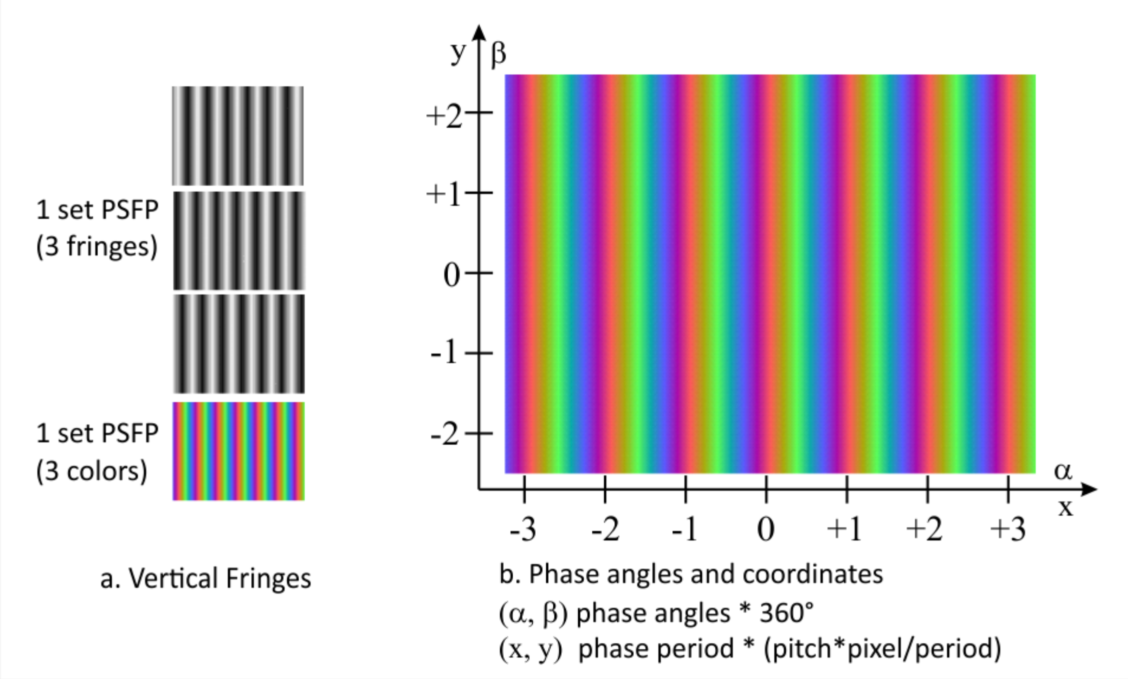


Fig. 1 The Universal Camera Model, mechanism of camera calibration, an LCD with PSFP on a linear stage. The location of the camera is fixed. The locations of the LCD can be at 2 or more.

In a set of PSFP, there are 3 phase shifting fringe images. The phase angle differences between them are 120° each other. Sensor pixels are *Phase Illuminated* all together via lens optics by the *Phase Encoded* LCD panel. Owe to the lens optics, the main contributors to a sensor pixel's intensity are the 4 adjacent LCD pixels through where the pixel ray vector passes. Both the 3 intensities and the 1 phase angle of the sensor pixel are *Physically Interpolated* via the lens optics by the 4 LCD pixels. An LCD monitor has only limited number of pixels. They provide discrete light sources and phase angles if they are encoded by the PSFP. It is the lens optics which did the *Physical Interpolation* for the sensor pixels. The PSFP concept and the physical interpolation phenomenon of the optics work together to make the LCD panel function as if that the camera were illuminated by a code modulated active lighting panel with *continuously distributed* phase angles and with *evenly distributed* phase brightness.

The essence of the principle and technology is: LCD and CMOS have extremely high pixel position accuracy. With the mathematical features of the PSFP, the exact space relationship between 1 CMOS pixel and the 1 equivalent LCD pixel is established via the unsatisfactory lens optics. The equivalent LCD pixel was physically interpolated by the 4 LCD pixels at where the CMOS pixel ray passes. By combining all the data obtained by the LCD at the 2 locations, the resulting distorted images, and the information of the desired zero-distortion image, a series of related matrices can be created. By resampling the image with the IMM thus calculated, the fidelity of the output image can be achieved with (even beyond) the level of LCD and CMOS positional accuracy. The characteristics of the PSFP also suppress the adverse effects that may be caused by non-uniform of both the LCD brightness characteristics and the CMOS photoelectric response characteristics. Thanks to these features of the PSFP, we have ingeniously avoided the geometric and physical (optical and electrical) defects of the lens, LCD and CMOS, and fully utilized the positional accuracy of LCD and CMOS, turns the *Physical Lens* into a *Mathematical Lens*.



a. Vertical Fringes

b. Phase angles and coordinates

Fig. 2 PSFP (Phase Shifting Fringe Patterns), vertical

## Camera model

In a camera, the sensor pixels are evenly distributed on the sensor chip at a few micron meter interval (pixel pitch) with very high pitch accuracy. The lens optics makes the pixels sense the scene light intensities in space. At the same time, the non-ideally made physical lens causes geometry distortions on the images. Conventional pinhole models try to model the geometry error property with up to a few tens of parameters. But the cameras with modern lenses (such as smartphone lenses) or with some special lenses (such as telecentric lens) may not meet the model's basic assumptions and requirements. This makes the math models to be not applicable or the errors are too big.

To determine each pixel ray in space, we need at least 2 reference points that are on the ray path and with accurately known coordinates in the WCS. A pixel ray vector can be formed by the 2 reference points. All these pixel ray vectors can be put together to form a matrix, the Pixel Ray Vector Matrix (PRVM).

## Calibration Panel

To calibrate each camera pixel ray accurately and independently, the calibration panel should be able to provide enough known reference points with enough accuracy. The density (or the resolution) of the reference points should be similar to (or higher than) that of the camera. The accuracy of the reference points should be subpixel level, or 1/10 of the pixel pitch or better, on the panel plane. In this way, all the camera pixels' geometry error property can be calibrated accurately and independently. In comparison, conventional calibration panels use chessboard or planar boards with dots on them. They can provide only limited number (tens to hundreds) of reference points with lower accuracy (pixel level). Only these limited number of camera pixels have reference coordinates to generate the parameters for the selected camera model. The model errors could be up to a few pixels or more at the sensor's edge and corner areas.

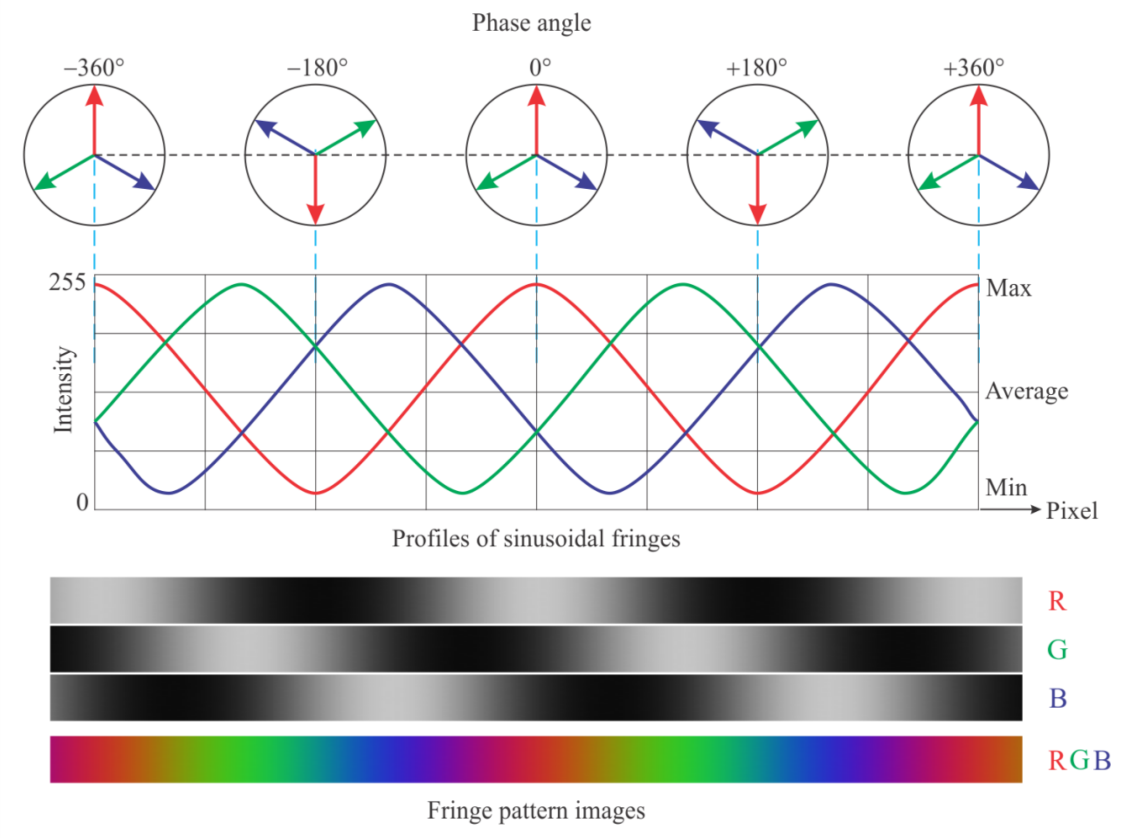


Fig. 3 PSFP (Phase Shifting Fringe Patterns) generation, profiles and images

An LCD flat panel monitor is perfect to be a calibration panel for both the *Geometry Property Calibration* and the *Intensity Property Calibration*. Similar to that of a camera sensor chip, the pixels of an LCD monitor are evenly distributed on the LCD panel with very high pitch accuracy (~ 1/20 of the pitch distance or better).

1. High density. It can provide high density reference points. An 8K LCD monitor, with 7,680 x 4,320 pixels, can provide 33 millions of reference points. A low cost 1080p LCD monitor, with 1,920 x 1,080 pixels, or 2 millions of reference points, is good enough to calibrate cameras with a few mega pixels.
2. High accuracy. The pixels pitch accuracy of an LCD is high enough to be the reference points for the cameras to be calibrated.

3. Even distribution. It can be used as an even brightness distributed light source to calibrate the camera's intensity property.

### Calibration mechanism

The PSFP images can be used to separate each pixel of the LCD monitor from all other pixels. Displaying one or more sets of PSFP images on an LCD monitor can make each pixel to be *phase encoded* uniquely and the phase angles be evenly (or linearly) distributed across the whole panel. The fringes will be arranged vertically and horizontally to separate the pixels in x-direction and y-direction respectively. The intensity profiles of the fringes usually are sinusoidal. When we generate the fringe pattern images and display them on the LCD monitor, each pixel's location (I, J), the coordinate (x, y) and the fringe phase angle (α, β) in x- and y- directions are all known. The pixel pitch of the monitor can be got from the monitor's specifications or can be measured directly.

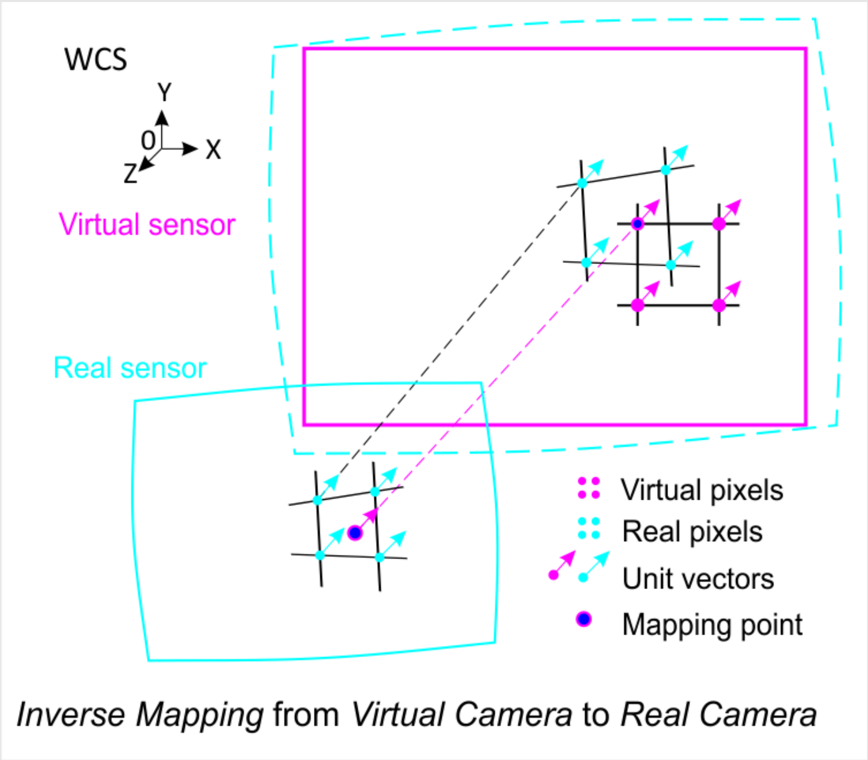


Fig. 4 Create a Virtual Camera. Inverse Mapping to get the IMM (Inverse Mapping Matrix)

The LCD monitor can be put on a linear stage. The axis of the linear stage should be perpendicular to the LCD panel plane (or in other words, be parallel to the normal vector of the LCD panel). In this way, at each linear stage's Z-locations, (x, y, Z), (I, J, Z), and (α, β, Z) of each pixel are all known.

The cameras capture the images for all the displayed PSFP. From each set of (3 vertical + 3 horizontal) fringe phase encoded images, for each sensor pixel at (U, V), the phase angles (λ, ϕ) can be determined directly by the phase calculation formula. The correspondent coordinates (x', y') are proportional to the phase angles (λ, ϕ). At the 2 LCD monitor Z-locations $Z_1$ and $Z_2$, for each camera pixel (U, V), we get 2 coordinates, $P_1'(x_1', y_1', Z_1)$ and $P_2'(x_2', y_2', Z_2)$. The 2 points in space form a ray vector that can be represented by a unit vector (i, j, k) and a point (x, y, z) on the ray. All these pixel points can be projected on to an arbitrary plane in WCS, such as z = 0. All these unit vectors form the PRVM matrix. The camera and sensor described by a PRVM can be called as a *Real Camera* and *Real Sensor*. The actual sensor can be called *Physical Sensor*.

A set of fringe patterns has 3 images. If the intensity profile of a fringe line is distributed as a sine wave, the 3 fringes can be generated by 3 sinusoidal equations with 120° phase differences between them each other. Denote the 3 intensities as ($I_0$, $I_1$, $I_2$), or (r, g, b), the 3 functions are:

$$I_0 = r = (Max+Min)/2 + (Max-Min)/2 * \cos(\theta) \qquad (1)$$

$$I_1 = g = (Max+Min)/2 + (Max-Min)/2 * \cos(\theta - 120°) \qquad (2)$$

$$I_2 = b = (Max+Min)/2 + (Max-Min)/2 * \cos(\theta + 120°) \qquad (3)$$

$$\theta = n*\Delta\theta,\ \ \Delta\theta = 360°/(6k) = 60°/k;\ \ k = 1,2,3,\ldots;\ n = 0,1,2,\ldots,6k-1;$$

$$0 \le Min < Max \le 255$$

The number of pixels in a fringe period should be the multiple of 6 to make the period covers integer pixels. Thus in a fringe period, the number of pixels will be (*6k*). A whole period will actually occupy (*6k+1*) pixels, with the first intensity and the last intensity be equal. The number of intensities will be (*3k+1*), where *k = 1,2,3,…* The typical value is *k=10.* For LCD with 32-bit color depth, the bit-depth for each color/channel is 8, or 256 gray levels, or [0, 255].

On the other hand, if the 3 values of the fringe intensity at a sensor pixel are known, its phase angle can be determined by the formula:

$$\theta = \mathrm{atan2}(\sqrt{3} * (g-b),\ 2*r-g-b\ ),\ \ \theta \in (-180°, 180°] \qquad (4)$$

*Intensity Calibration* (to get the photoelectric property, or sensor-to-monitor response curve) should be done before *Geometry Calibration* (to get the sensor pixel ray vectors in WCS space). Sometimes the response curve can be adjusted to be linear by adjusting in conjunction the *Gamma Values* of the graphics card, the monitor and the camera. The phase angles and the correspondent coordinates will not be evenly (or linearly) distributed on the LCD panel plane if the response curve is not linear, or the nonlinear errors are not compensated during the creation of the PSFP. The errors will be distributed periodically at the *fringe space frequency* on the LCD panel. In this case, linearization and error compensation approaches and algorithms can be employed.

Typical example: LCD resolution 1,920x1,080, dot pitch 0.22mm. Choose *k* = 10. Then the fringe occupies (*6k*) = 60 pixels per period; the gray value of each period has (*3k+1*) = 31 pixels. The full screen has 1,920/60 = 32 periods in the horizontal direction and 1,080/60 = 18 periods in the vertical direction. The phase difference between two adjacent pixels on the LCD，*phase pitch*, is 360°/(*6k*) = 6°. After CMOS obtains three grayscale images, at each pixel, a phase angle can be calculated with its three grayscale values. Divide it by the phase pitch of the LCD (6° in this example) to directly obtain the corresponding LCD pixel (*Physically Interpolated* position). Multiply it by the dot pitch of the LCD (0.22 mm in this example) to obtain the coordinate.

A brief description of the process: The CMOS camera captures the 3 grayscale fringe images displayed by the LCD. For each pixel of CMOS, one phase value is calculated from 3 gray values, divided by the LCD phase pitch (6°), multiplied by the LCD dot pitch (0.22mm), and the x-coordinate of the pixel vector in space is obtained. This is in the X-direction. And the same operation is performed in the Y-direction to obtain the y-coordinate. When the LCD is placed in another Z-position, the x-coordinate and y-coordinate of the second point that the ray passes can be obtained. Combining the two points' coordinates obtained by the LCD in two Z-positions, a space vector can be constructed. It is to be noticed that all of the above fringe images are generalized monochrome. It means that the optical spectrum characteristics for the 3 PSFP images should be the same. All photos should be monochrome or be converted to monochrome before processing.

## Virtual camera

In the PRVM, the vectors are not evenly distributed in space. Any 4 adjacent pixels on the real sensor plane form an arbitrary quadrilateral. Since the vectors are not parallel each other (for telecentric camera), nor converge to a single point (the optical center in a pinhole camera), the arbitrary quadrilaterals will have different shapes on different planes that they project to. For most applications, image mapping is required to get the colors and/or the intensities in between the 4 real pixels. To make the image processing more accurate and more efficient, an error free *Virtual Camera* and *Virtual Sensor* can be created. The *Original Images* can then be converted to the distortion free *Virtual Images* before further processing. Fig 4. depicts the FOV of the real camera and that of the virtual camera.

1. The virtual sensor could be at any locations and/or orientations in the WCS.
2. The FOV of the virtual camera should be smaller than that of the real camera to make every pixel of the virtual camera non-blank.
3. Their geometry specs (such as resolution, aspect ratio, etc.) can be the same.
4. There are 2 options for the virtual camera. One is to keep the virtual camera model-less and error free. The other is to use a conventional camera model.

With above assumptions, a *Virtual PRVM* and an *Inverse Mapping Matrix* (IMM) can be generated. The pixel locations of the Virtual PRVM can be determined from the assumptions directly. For the *Virtual Unit Vectors* there are 2 options. For the model-less error-free option, all the *virtual pixel vectors* can be inverse mapped by the IMM. The Virtual PRVM keeps the same properties as that of the original PRVM. For the second option, the *virtual pixel vectors* of the Virtual PRVM has a model. For pinhole camera, all the virtual vectors will converge to a single point, the Optical Center (OC). For telecentric camera, all the virtual vectors will be parallel to each other. For the *Modeled Virtual PRVM*, the virtual vectors are not error free anymore. An *Error Vector Matrix* (EVM) can be derived from the (model-less) Virtual PRVM and the Modeled Virtual PRVM simply by the vector subtraction. An *Optical Center* can be fitted by all the Unit Vectors in the PRVM. This is also the *Virtual Optical Center* of the *Virtual Pinhole Camera*.

To generate the IMM, firstly by using the PRVM, all the real pixels will be projected to the virtual plane to get the *Projected Real Pixel* (I, J; x, y, z). Within the *Virtual Sensor Plane*, for each *Virtual Pixel* (U, V; x', y', z'), there are 4 adjacent *Projected Real Pixels* (I, J; x, y, z)x4. The 4 pixel points form an arbitrary quadrilateral (not square). Based on the relationship between the 1 virtual pixel and the 4 pixels, the locations and the coefficients for the 4 pixels can be calculated and be saved in the IMM.

In application stage, the IMM matrix is used to map the received *Original Images* to get the *Virtual Images.* The virtual images are all we need for 2D applications. For 3D applications, if the object surface is on or near the virtual sensor plane, the vector errors could be negligible. It it's far away from the virtual plane, the vector error should be compensated by using the EVM matrix. If required a new virtual sensor can be created in WCS and the new matrices can be got. The *Inverse Mapping* operation can be done simply by 4 multiplies and 4 additions with the parameters in the IMM. The computation time will be similar to that for normal image resampling.

Before the inverse mapping, the IM matrix should be used to compensate the pixel intensity errors. A possible usage of the 2 matrices, IMM and the IM, is that they can be saved into the camera and let the camera do the processing. In this way, the camera can output the distortion-free virtual images directly. If the camera has not enough processing capability, the external computer can get the matrices from the camera and do the image processing.

### Reduced-dimensional PRVM

Most consumer cameras (such as cell phone cameras, surveillance cameras, etc.) are not critical to the fidelity of imaging. In this case, we can use the reduced-dimensional PRVM. It retains all the concepts of PRVM, but it's very easy to implement: just use the camera and take 6 photos of the LCD with the PSFP. IMM can be obtained from these 6 photos. A mirror can help make the optical axis of the camera lens perpendicular to the LCD plane. After resampling the photo, the distortion will be less than 1 pixel. The optical axis of the camera, or the line of sight, should be perpendicular to the LCD’s surface. A mirror can be used to help the alignment. It can also be done without a mirror by displaying the camera video images on the LCD. Adding more components can make the whole process from the alignment to the calibration automatic.

### Summary

A set of matrices (IM, PRVM, IMM, Virtual PRVM, etc.) are established to fully describe the physical property and the geometry property of a camera in the very detail for each single pixel. Each pixel’s properties are independently calibrated and represented. The mechanism, the approach and the algorithms for the calibration are accurate and robust, simple and low cost. The PRVM is universally suitable for all types of cameras. The original image can be inverse mapped to get a distortion-free virtual image by using the IMM. The virtual images kept the best sharpness of the original images. With the IM, the intensity property of each pixel is linearized and normalized. The computational cost to get the virtual image is no more than that of a normal image resampling. The concept of the *Virtual Camera* makes a non-ideally made *Physical Camera* be a perfect *Math Camera*. A math camera (math sensor) can be reconfigured in space at anywhere and with any topology structure without producing distortion errors. Due to the nature of the virtual cameras, they can be configured to make best use of the most advanced modern computer hardware, such as CPU, GPU, SoC, etc. The distortion free virtual images can be directly used by any existing software that use conventional math models. That guaranties the best backward compatibility. By introducing the concept of the PRVM and the Math Camera, and by using the techniques of the calibration and the mapping, camera images will no longer have geometry distortions, the defects that exist all the time since the birth of the cameras. Thanks to the continuous advancement of science and technology, human beings can finally watch, record and measure this real world with zero distortion.

**Camera 2.0**, the new era with *Zero Imaging Distortion*, has begun!

# Implementation and Applications

## Implementation of Calibration

There are a lot of error sources in a camera imaging and measurement system. The errors may be caused by optics, electrical circuits or math models. All the errors will propaganda physically and mathematically to the final results. Keeping the errors minimum at each stage and at the very beginning can make a better system. Many approaches and algorithms can be used to achieve the best results by model creation, calibration, error compensation and the applications.

1. Calibration target
An LCD or other flat panel monitors can be used to do the calibration.
1.1 The LCD screen size must be big enough to cover the whole FOV of the cameras at all z-locations. For the same screen size, higher resolution is preferred.
1.2 The LCD's surface should be flat without dent, and not be curved.
1.3 Flat panels that have wider viewing angles and even brightness, such as IPS and OLED have better properties than TN, TFT or other types.
1.4 For measurement applications, the refractive index (~1.5) and the coating thickness (~0.1mm) of the LCD can be modeled and the errors should be compensated.
1.5 The LCD can be put at 3 or more z-locations. The PRVM can be fit and optimized more accurately by Gaussian filtering, NURBS model or other math means. The vectors' error noises can be reduced accordingly.
1.6 To increase the accuracy, the fringes can be have higher space frequency (shorter space periods). Fringes with lower space frequency (longer space periods) may be needed to eliminate the multiple phase issue.
1.7 To reduce the noises, when generating the fringes, it's better not to use the 2 low/high intensity ends. E.g.: keep the brightness within [20, 240], not [0, 255].
1.8 High bit-depth (10-bit or more per channel) of computer graphics card and/or LCD can be used to get better quality (more smooth) fringe images.
1.9 Image dithering and other image enhancing techniques may make the fringe patterns more smooth.
1.10 From the computer graphics card to the LCD, digital interfaces (HDMI, DP, etc.) are preferred to eliminate any transfer noises cause by the cable.

2. Linear stage
2.1 To calibrate cameras with large FOV (e.g.: 1,280x1,024mm), the LCD could be too big and heavy. In this scenario, the LCD can be kept fixed and the cameras can be put on the linear stage instead. In both scenarios, the axis of the linear stage should be aligned to be perpendicular to the monitor's surface (be parallel to the monitor surface's normal vector). The alignment can be done by the help of a first surface mirror and a laser pointer (or a camera).
2.2 The calibration can be done automatically if the linear stage is power driven and can be controlled by a computer. The calibration takes only a few minutes or even a few seconds.
2.3 For DIY or when a linear stage is not available, the monitor can be hold on hands to move around in front of the cameras. The usage of the monitor is similar to that of a chessboard or a planar board with round dots on it. The difference is that the monitor must cover all the sensor FOV. Be sure to make the monitor stable at each location/orientation when a set of fringe patterns are displayed, and the images are captured by the cameras and be saved to storage. The algorithms to determine the pixel ray vectors are different than that with linear stage.

3. Camera
3.1 If a camera can support high bit-depth image output, the highest bit-depth images should be used.
3.2 If the camera can support *External Trigger/Sync Signal*, it should be used to sync with the monitor. This will reduce the image noises greatly. A sync circuitry may be needed to ensure that the exposure time of an image is only happened within one monitor fringe frame (not crossing 2 frames).
3.3 Image averaging may be applied during image capturing to further reduce the noises.
3.4 The images captured by the camera should not be saturated, nor be too dark. The camera parameters (such as exposure time, gain, etc.) can be adjusted accordingly.
3.5 The focus and the aperture of the camera lens should be adjusted and locked before the geometry calibration. The lens aperture can also be adjusted slightly after calibration but not recommended. The camera should not be touched during and after the calibration to avoid changing the geometry structure between the lens and the sensor, or between the 2 cameras of the stereo camera pair.
3.6 Camera *Intensity Calibration* (monitor-to-camera property) should be done before the Camera *Geometry Calibration*. The monitor can be put at locations that it faces straight toward the camera and the camera is out of focus on the monitor. A respond curve or a Look Up Table can be generated after the calibration.

## Virtual stereo camera

Most 3D scanner or a 3D-DIC (Digital Imaging Correlation) system use a pair of *Stereo Pinhole Camera*. They can be formed by 2 cameras, or by one single-camera with a split-lens. To achieve better 3D results, shift-lenses or other optical structures may be used to make the stereo cameras always have the same focus property for all the matched pixels and eliminate the perspective distortions. The 2 cameras should be aligned to share the maximum Common FOV or have the maximum coverage within the whole 3D DOF volume.

Two *Pinhole-modeled Virtual Cameras* can be created with the 2 PRVM and 2 Virtual PRVM can be generated. The *Pinhole-model Virtual Stereo Camera* will be configured in such a way that the *Virtual Stereo Sensors* will be created on a *Rectified Common FOV Plane*. The Plane and every row of the Virtual Pixels will be parallel to the *Baseline Vector* that is formed by the 2 fitted *Optical Centers* of the 2 *Real Cameras*. On the *Common FOV*, the pixel locations of the 2 *Virtual Sensors* are exactly the overlapped. But the directions of the unit vectors of the 2 virtual cameras are different. They converge to the 2 Optical Centers respectively. The *Pinhole-Model Virtual PRVM*, the IMM and the *Error Vector Matrix* (EVM) for the 2 virtual cameras can be created accordingly. The original images from the 2 cameras will be mapped to the *Rectified Common FOV Plane* by using their IMM for further processing.

The *Rectified Virtual Stereo Camera Pair* make the epipolar geometry from 2-D matching to 1-D matching. Computers with new generation CPU and GPU can benefit greatly from this 1-D matching feature. For 3D scanner, this can make the pixel matching and 3D reconstruction have highest efficiency and accuracy. For 2D/3D-DIC systems, when doing the subset selection, matching and correlation, for the squared subset, one line of the first camera will match exactly only the same line of the second camera. There's no need to do epipolar processing repeatedly for the same pixels. Furthermore, for shift-lens 3D-DIC systems, there's no perspective errors. Finally, among all the possible combinations of sensors and lenses, the very best configuration is the single-camera with a split-lens that has also the shift lens features at the same time. In a 3D system, the optical property of the shift lens is that the focus planes of the 2 cameras are coincide. At all other parallel planes within the whole depth of field, the 2 matched pixels have exactly the same optical focus properties. For single-sensor shift/split-lens 3D system, all the matched pixels and/or the subsets have exactly the same properties: Optical-to-Electrical, Electronics, Optics, etc. For best results, the *Error Vector Matrix* (EVM) can be used to compensate the errors when the object surfaces are far away off from the *Rectified Common FOV Plane*.

Usually the 2 cameras should be calibrated together to make the 2 cameras share the same WCS during calibration. They can also be calibrated separately. In this case, each camera can be set to face straight forward to the monitor during the calibration to get the best results for all the pixels. After the 2 cameras being calibrated, there should be additional registration to make the stereo camera pairs to be within the same WCS.

If the images that were mapped on to the *Rectified Common FOV Plane* are to be used for existing 3D scanner or 3D-DIC software, the correspondent pinhole camera models with proper intrinsic and extrinsic parameters should be established. In the pinhole models, the error related coefficients will all be 0 since the virtual images are distortion free.

### Color camera

For a color camera that can output Bayer images, the exact physical location of each pixel is fixed and its ray vector can be calibrated. Alternatively, the Bayer image can also be decoded from the color image in case it cannot be output directly.

### Smartphone camera

The lenses of smartphones are not manufactured with conventional technology. They may not meet the basic requirements and assumptions of the conventional pinhole models. The errors could be too big. The assumption free PRVM can be used without any problems.

If a Reduced-dimensional PRVM is used, the smartphone can generate the PSFP, send it to the LCD, and take photos. A *Mobile App* can make this process automatic. The alignment process (to make the camera's optical axis be perpendicular to the LCD's surface plane) can be half automatic or fully automatic (with the help of a 2-axis rotary stage or gimbal). Image Fusion of an expected crosshair and the captured crosshair, and displaying it on the LCD dynamically will help to do the alignment accurately and quickly. The whole process can be fully automatic.

### Special lens camera

Cameras with special lenses usually do not meet the basic requirements or assumptions of the conventional math models. These lenses include telecentric lens, tilt-shift lens, Scheimpflug principle lens, etc. For all these lenses, the PRVM can be used perfectly.

* * *

## Terms

- *Physical Sensor*. It’s the actual CMOS/CCD sensor with all the pixels evenly distributed on the *Physical Sensor Plane*.

- *Real Camera.* It’s the camera described by a *PRVM* (the whole set of the *pixel unit vectors)*. Where the locations of the *Real Pixels* of the *Real Sensor* are all on the same *Real Sensor Plane,* but are NOT evenly distributed. The adjacent 4 pixels form an *Arbitrary Quadrilateral*. The *Real Pixel Vectors* are distributed not as ideally as expected. E.g.: They are not converged to a single point (the Optical Center, as for a pinhole camera), nor be parallel to each other (as for a telecentric camera). Hence the shape of the *Arbitrary Quadrilateral* will change when they are projected to a plane, even if the plane is parallel to the real sensor plane.

- *Virtual Camera*. It’s the camera described by a *Virtual PRVM,* and may or may not be accompanied by an *Error Vector Matrix* (EVM). Where the *Virtual Pixels* of the *Virtual Sensor* are evenly distributed on the *Virtual Sensor Plane.* The adjacent 4 pixels form a square. Depend on the options, the *Virtual Pixel Vectors* are either model-less or with models (pinhole, telecentric, etc.). For *Model-less Virtual PRVM*, the *Virtual Pixel Vectors* keep the properties of the PRVM of the *Real Camera*. For *Modeled Virtual PRVM*, the *Virtual Pixel Vectors* are either converged to a single point (as for a pinhole camera), or be parallel to each other (as for a telecentric camera). An *Error Vector Matrix* (EVM) will be generated to describe the vector errors when *Modeled Virtual PRVM* are used.