obtained from an image source, such as from memory coupled with the IPD. The correspondence of image pixels with the next M screen positions are not necessarily one-to-one. For example, one image pixel may cover and correspond to multiple screen positions, depending on the resolution of the image to be displayed. The obtained image is then projected by the IPD onto one or more of the M screen positions. In one embodiment, the next M screen positions are determined on a continuous basis, for example, using another sliding window with a width of M. The process terminates at block **892**. Additional process details are described below with respect to FIG. **8**B.

[0143] FIG. **8**B shows a flow diagram of one embodiment of a process of generating an image with the IPD of FIG. **1**. With reference to FIGS. **1** and **8**, the overall process of generating an image using a IPD starts at block **800** and proceeds to block **805** where tracer beam **122** pulses are projected onto screen **114**. As discussed above, tracer beam **122** may be IR pulses that are projected in parallel with combined image beam **120** onto screen **114** and subsequently detected by detector **112**. Tracer beam **122** is used to predict the next screen position on the current scan line being swept across screen **114** by scanner **110**. The process proceeds to block **810**.

[0144] At block **810**, detector **112** is used to detect tracer beam **122** pulses, and provide the raw data and/or preprocessed data associated with the pulses to processor **116**. In one embodiment, a sliding window algorithm is utilized to collected data about the N preceding pulses corresponding with the N preceding screen positions. The process proceeds to block **815**.

[0145] At block **815**, processor **116** calculates the trajectory of the current scan line to predict and/or estimate the next screen position on the current scanline for display of image pixel from memory **118**. In one embodiment, next M screen positions are predicted based on the N preceding tracer beam **122** pulses on the current scanline. In one embodiment, the determination/prediction of the next M screen positions is repeated for each of the M pixels as the current scanline sweep continues by the scanner **110**, thus implementing a sliding window algorithm both at the feedback end where data are collected about N preceding pulses and at the prediction end where M next screen positions are predicted. The sliding window at the feedback end is N pulses wide while the sliding window at the prediction end is M pixels wide. The process moves on to decision block **820**.

[0146] At decision block **820**, the process determines whether adjustment coefficients are updated and whether pixel color values, such as intensity or saturation, need adjustment before display. If so, the process proceeds to block **825** where the pixel values obtained from memory **118** are adjusted using the adjustment coefficients before proceeding to block **830**. Otherwise, the process proceeds directly to block **830**.

[0147] At block **830**, the next M screen positions on the current scanline are determined. As noted above, in one embodiment, the next M screen positions are predicted based on dual sliding windows, one at the feedback end where data about preceding N screen positions are collected, and one at the prediction end where each of the next M screen positions are determined based on the data collected about the preceding N screen positions. The process proceeds to block **835**.

[0148] At block **835**, component image beams outputted by light sources **104** are modulated according to the corresponding values of color components of the pixels of image in memory **118**, where the pixels correspond to the next screen position on the current scanline predicted at block **830**. For example, the intensity of the R (Red) component image beam may be set to the red component value of the image in memory **118** for the pixel to be displayed next on the current scanline. The process proceeds to block **840**.

[0149] At block **840**, the component image beams, for example, RGB components, are combined together to form one combined image beam **120**. In one embodiment, a prism may be used to combine the component image beam. In other embodiments other methods currently known in the art or methods to be discovered in the future may be used to combine the component image beams. The process proceeds to block **845**.

[0150] At block **845**, a scanner **110**, for example a MEMS device with a rotating mirror with two degrees of rotational freedom, for example, two orthogonal planes of rotation, reflects the combined image beam **120** as pseudorandom scanlines sweeping across screen **114**. Different methods may be used to inject randomness into the direction of the scanlines from one scan cycle to the next. These methods range from mechanical and physical means, such as imprecisely controlled off-center vibrations at microscopic level, to electronic and software means, such as random number generators. The process proceeds to block **850**.

[0151] At block **850**, detector **112** or another image detection camera optionally detects the reflection of combined image beam **120** from screen **114**. In one embodiment, combined image beam **120** is scattered off screen **114** and is refocused onto detector **112**, or the other image detection camera, using a lens **510** (see FIG. **5**). Data collected about projected pixel values and positions are used to improve the projected pixel values (the image) on the next scan cycle. The process proceeds to decision block **855**.

[0152] At decision block **855**, the data collected about projected pixel values are compared with the corresponding pixel value of the image stored in memory **118** to determine any deviations. If any deviations are detected, for example, because of screen **114** color or texture, adjustment coefficients for the pixel values of the image in memory **118** are updated at block **860** to adjust such pixel values for the next scan cycle. Otherwise, the process proceeds to block **805** and the process is repeated for the next screen position on the current scanline. As one scanline is completed, another scanline is started and the same process described above is repeated with respect to the new scanline.

[0153] It will be understood that each block of the flowchart illustration, and combinations of blocks in the flowchart illustration, can be implemented by computer program instructions. These program instructions may be provided to a processor to produce a machine, such that the instructions, which execute on the processor, create means for implementing the actions specified in the flowchart block or blocks. The computer program instructions may be executed by a processor to cause a series of operational steps to be performed by the processor to produce a computer implemented process such that the instructions, which execute on the processor to provide steps for implementing the actions specified in the flowchart block or blocks. The computer program instructions may also cause at least some of the operational steps shown in the blocks of the flowchart to be performed in parallel. Moreover, some of the steps may also be performed across more than one processor, such as might arise in a multi-processor