In any case, getting black color with slow machines is problem.
>I could try it on our 8 bit screens but I don't know how to
>render pixels with X in constant time. I recall our double buffer
>has other image color and one b&w -- that doesn't help either.
>Maybe I should dump photos to screen with low level code; how?

A few years ago a friend and I took some 256 grey-level photos from a 1 bit Mac Plus screen using this method. Displaying all 256 levels synchronized to the 60Hz display took about 10 seconds. After experimenting with different aperture settings and screen brightnesses we found a range that worked well, giving respectable contrast. The quality of the images was pretty good. There were no visible contrast bands.

To minimize the exposure time the display program built 255 different 1 bit frames. The first contained a dot only for pixels that had value 255, the second only for pixels that had value 254, etc. These frames were stored using a sparse data structure that was very fast to 'or' onto the screen in sequence. Creating these frames sometimes took 5-10 minutes on that old Mac, but the camera shutter was closed during that time anyway. And yes, we wrote directly to the screen memory. Mea culpa.

Our biggest problem was that small images were displayed in the top left corner of the screen instead of the center. It took an extra week to have the film developed and printed, because the processors took the trouble to manually move the all images into the center of the print. Who'd have guessed?