There's third way too.

With zooms in range 0..100% the new scaled image is always smaller/equal to original picture, so the memory consumptions is under O(2). I don't think this case needs further improvements, and I would keep it as it is. (actually this method is probably optimal even for zooms like 100-200%)

When the zoom is > 100%, you can try different approach, not to scale whole image, but only the current viewport of it.
In case even this is too slow, you may during panning use the previous viewport from cache, and rescale only the parts which are newly on screen.

I'm not sure how large your viewport is, but anything under 1000x1000 pixels should be well manageable by GHz CPUs, if you have the original picture in RAM.

In this way your rescaled image for >100% zooms will have fixed size, and will not grow further with bigger zooms, also with bigger zooms you will use smaller and smaller part of original picture to create the rescaled one, so the performance will grow.

Of course this is pure theory, maybe it would not perform as well as I think in real world, but I have some experience with real time graphics, and unless your view has like 2000x1000 pixels, the modern CPU + RAM should easily rescale such amount of data on the fly.

edit: cbpporter suggested pretty much the same thing I see.