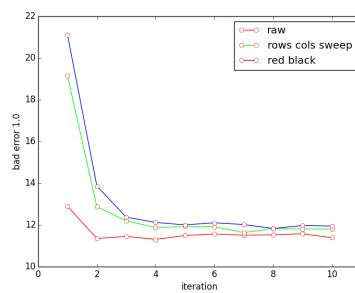


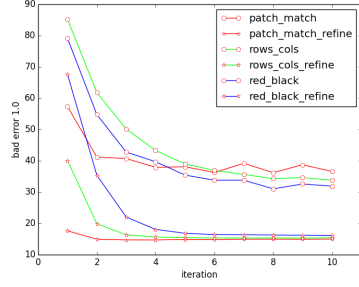
(a) left disparity without and with refine.



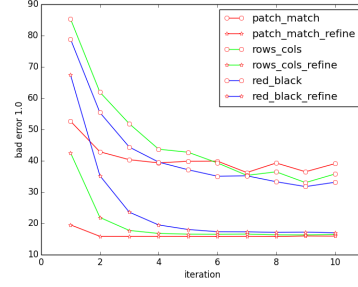
(b) right disparity without and with refine.

图 1: PatchMatch 中不同传播方式对 bad error rate 的影响

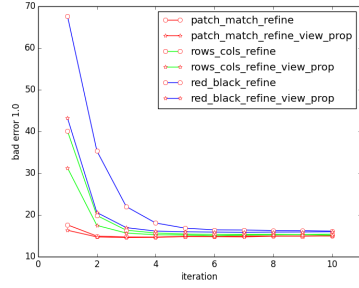
图 1显示了使用不同传播方式下的 bad error 1.0 rate, 红线最优、绿线次之、蓝线最差。



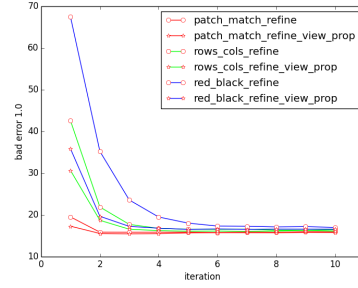
(a) left disparity without and with refine.



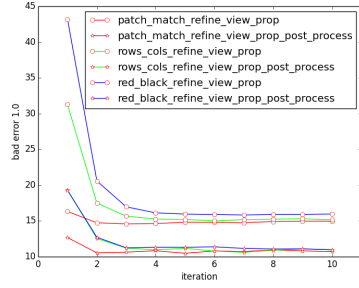
(b) right disparity without and with refine.



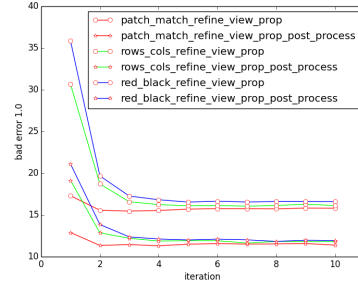
(c) left disparity without and with view propagation.



(d) right disparity without and with view propagation.



(e) left disparity without and with post-processing.



(f) right disparity without and with post-processing.

图 2: PatchMatch 中不同流程对 bad error rate 的影响

对 PatchMatch 算法中各个流程的对 bad error rate 的影响进行了实验。

图 (a)(b) 对比了是否具有 plane refinement, 在多次迭代之后的 bad error rate。可见, 是否具有 plane refinement, 对 bad error rate 的影响很

大。

图 (c)(d) 对比了是否具有 view propagation, 在多次迭代之后的 bad error rate。对于原生的 PatchMatch 方法, view propagation 的作用不大, 而对于使用其它传播方式的两种变种 PatchMatch 方法来说, view propagation 在前几次迭代具有显著的作用, 不过在多次迭代之后, 就没有作用了。这是因为变种 PatchMatch 方法的传播方式为了满足并行的要求, 存在着局限性(什么局限性??), 使得传播效果不是很好, 所以此时 view propagation 起到了修补的作用, 所以在前几次迭代能够显著减低 bad error rate; 而在几次传播之后, 进行了充分的传播之后, 此时 view propagation 就没什么作用了。

图 (e)(f) 对比了是否具有 post-processing, 在多次迭代之后的 bad error rate。毫无疑问, post-processing 也是能够显著降低 bad error rate 的, 正如大部分论文所说的那样。

纵观全局, 在所有的图示中都能观察到, 红线最优、绿线次之、蓝线最差, 这是其中的传播方式所造成的影响。而且 view propagation 起到的作用, 则是取决于使用了何种传播方式以及以及总迭代次数。

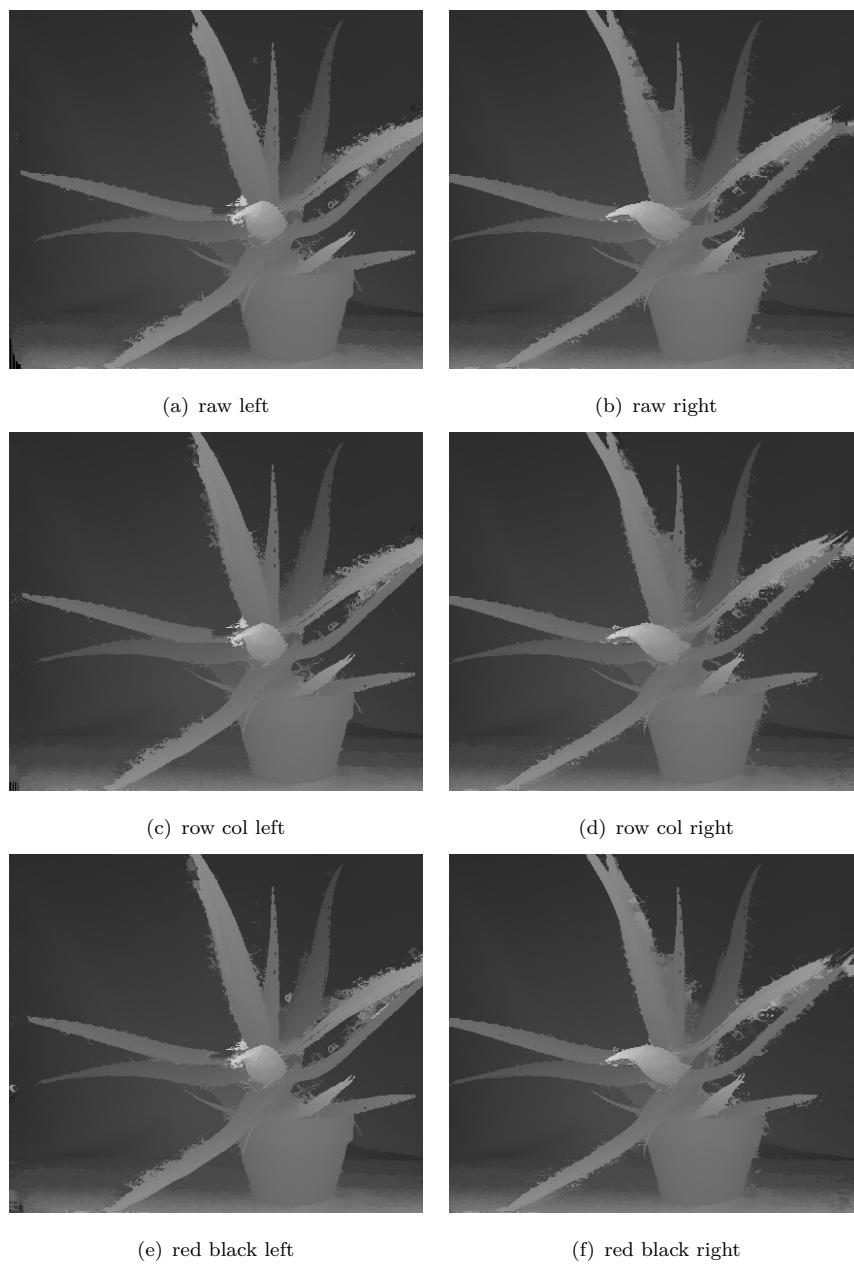


图 3: 不同传播方式下的视差图比较

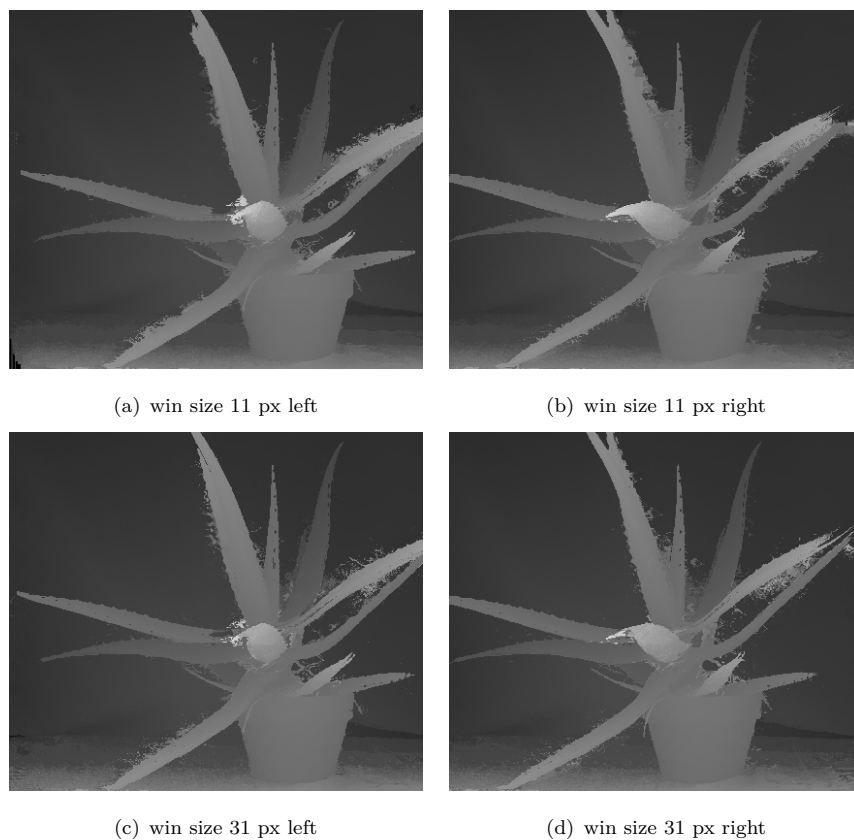


图 4: 不同窗口大小下的视差图比较

图 3显示了使用不同传播方式得到的视差图的比较结果，可以看到，几乎没什么区别。

图 4显示了原生的 PatchMatch 方法使用不同的窗口大小得到的视差图后的比较结果，可以看到，大窗口得到的视差图在视觉上明显更优于小窗口得到的视差图。而在 bad error rate 上，前者略微比后者优秀。