LMMSE method estimates of a missing primary color sample in both horizontal and vertical directions then combines the two to reconstruct the channels, we can use any interpolation method to interpolate the missing green samples at red and blue pixel (in this case we used second-order Laplacian interpolation filter), then interpolate the missing red and blue samples at green samples.
After this step, we obtain $\bar{\Delta}_{g,r}$ by using the interpolated green value at i position minus the red value at i, and using green value at i minus the interpolated red value at i for both horizontal and vertical direction, then we obtain the error by using $\Delta_{g,r}$ (obtained by $G_i - R_i$) value minus the $\bar{\Delta}_{g,r}$.
Now we have error = $\Delta_{g,r}$ - $\bar{\Delta}_{g,r}$, which means that $\bar{\Delta}_{g,r} = \Delta_{g,r}$ - error, now we have to estimate $\Delta_{g,r}$ using $\bar{\Delta}_{g,r}$ for horizontal and vertical direction, Using LMMSE, we have the optimal value as $E[\Delta_{g,r}] + \frac{Var(\Delta_{g,r})}{Var(\Delta_{g,r})+Var(error)}(\bar{\Delta}_{g,r} - E[\Delta_{g,r}])$. And using this method, we can estimate $\Delta_{g,b}$ as well.
Now all we have left is how to get the value of$E[\Delta_{g,r}], Var(\Delta_{g,r})$ and $Var(error)$. If we assume that this random process is ergodic and stationary, $E[\Delta_{g,r}]$ can be estimated by using its neighbors, thats
$E[\Delta^i_{g,r}] = \frac{1}{2L+1}\sum_{k=1}^{2L+1} Y^s_n(k)$
, where $Y_n = [\bar{\Delta}_{g,r}(n-L)...\bar{\Delta}_{g,r}(n)...\bar{\Delta}_{g,r}(n+L)]$, and Y^s_n is the weighted average of Y_n
$Var(\Delta^i_{g,r}) = \frac{1}{2L+1}\sum_{k=1}^{2L+1}(Y^s_n(k) - E[\Delta^i_{g,r}])^2$
and $error^i = \frac{1}{2L+1}\sum_{k=1}^{2L+1}(Y^s_n(k) - y_n(k))^2$.
(Our implementation for LMMSE till Intermediate project report ends here, the next parts' code haven't been implemented yet)
Using this method, now we have two values for a single position because we have horizontal and vertical direction, let $\bar{x}_w(n)$ be the final predicted value, we have
$\bar{x}_w(n) = w_h(n) * \bar{x}_h(n) + w_v(n) * \bar{x}_v(n)$, where h and v means horizontal and vertical, $\bar{x}$ is the predicted value from last step, where $w_h(n) = \frac{Var[\bar{x}_v(n)]}{Var[\bar{x}_h(n)+Var[\bar{x}_v(n)}$ and $w_v(n) = \frac{Var[\bar{x}_h(n)]}{Var[\bar{x}_h(n)+Var[\bar{x}_v(n)}$, finally, put this all together, we have calculate the missing green values using red value at n + $\Delta_{g,r}(n)$ or using blue value at n + $\Delta_{g,b}(n)$.
For missing red (blue) samples at the blue (red) sample positions, we use the green value - the average of four green-red(green-blue) difference value of its neighbors to obtain red(blue) value.
Then the last step is to estimate the missing red/blue samples at the green sample positions, because last step, now we have the value of green-red difference values for its neighbors and we can compute the bilinear average of the green-red differences, apply the same step to obtain the final missing value.
With this, we can predict all the missing values and reconstruct the image.