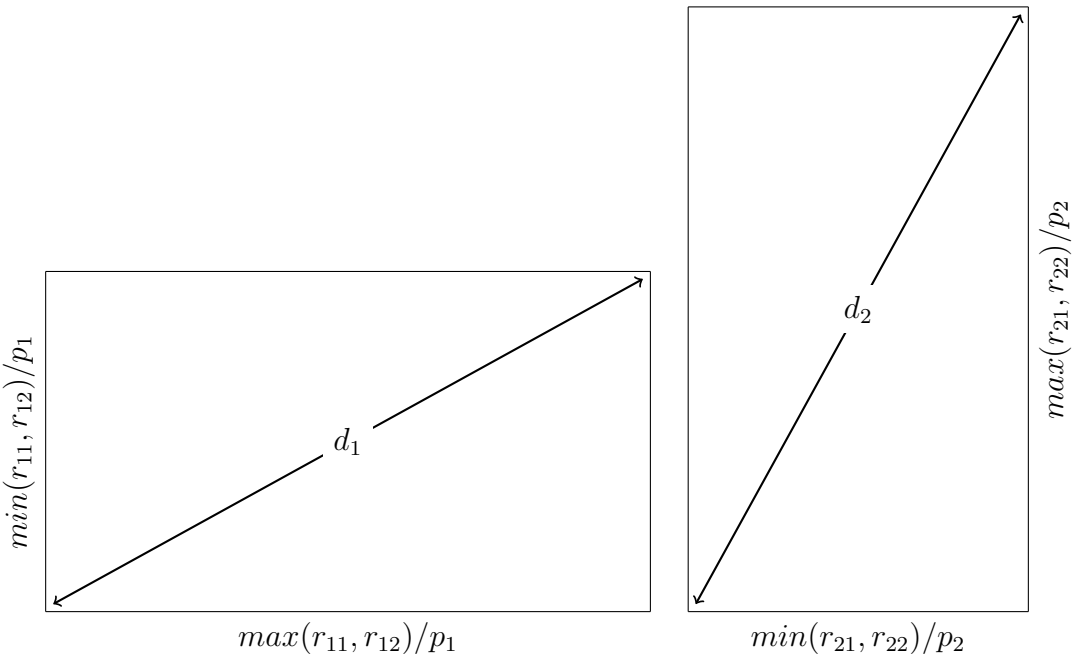
Determination of Optimal Matching Monitor Dimensions



We note that $p_i = \sqrt{r_{i1}^2 + r_{i2}^2}/d_i$, and that $r_{ij}/a_{ij} = k_i$ where:
d_i = length of the major diagonal of monitor i in inches.
r_{ij} = defining resolutions of monitor i (e.g. 1920 by 1080, $r_{i1} = 1920$, $r_{i2} = 1080$).
p_i = pixels per inch of monitor i.
a_{ij} = aspect ratio of a monitor i (e.g. 16 by 9, $a_{i1} = 16$, $a_{i2} = 9$).
k_i = a constant used to relate resolution to aspect ratio.

As a result, the ideal match for a given monitor can be found by specifying either the resolution (1) or diagonal measure (2) of a secondary monitor, as shown:

$$
\begin{aligned}
p_1 &= p_2 \\
\frac{\sqrt{r_{11}^2 + r_{12}^2}}{d_1} &= \frac{\sqrt{r_{21}^2 + r_{22}^2}}{d_2} \\
\frac{d_1 \cdot \sqrt{r_{21}^2 + r_{22}^2}}{\sqrt{r_{11}^2 + r_{12}^2}} &= d_2 \qquad (1) \\
d_1 \cdot \sqrt{r_{21}^2 + r_{22}^2} &= d_2 \cdot \sqrt{r_{11}^2 + r_{12}^2} \\
\sqrt{r_{21}^2 + r_{22}^2} &= \frac{d_2 \cdot \sqrt{r_{11}^2 + r_{12}^2}}{d_1} \\
r_{21}^2 + r_{22}^2 &= \frac{d_2^2 \cdot (r_{11}^2 + r_{12}^2)}{d_1^2} \qquad (2)
\end{aligned}
$$

When provided with the aspect ratio of the secondary monitor, (2) can be utilized to provide the exact resolution of the secondary monitor, instead of the squared sum of its resolution.

$$
\begin{aligned}
\begin{bmatrix} r_{21} \\ r_{22} \end{bmatrix} &= k_i \cdot \begin{bmatrix} a_{21} \\ a_{22} \end{bmatrix} \\
\begin{bmatrix} r_{21} \\ r_{22} \end{bmatrix} &= \sqrt{k_i^2} \cdot \begin{bmatrix} a_{21} \\ a_{22} \end{bmatrix} \\
\begin{bmatrix} r_{21} \\ r_{22} \end{bmatrix} &= \sqrt{k_i^2 \cdot 1} \cdot \begin{bmatrix} a_{21} \\ a_{22} \end{bmatrix} \\
\begin{bmatrix} r_{21} \\ r_{22} \end{bmatrix} &= \sqrt{k_i^2 \cdot \frac{a_{21}^2 + a_{22}^2}{a_{21}^2 + a_{22}^2}} \cdot \begin{bmatrix} a_{21} \\ a_{22} \end{bmatrix} \\
\begin{bmatrix} r_{21} \\ r_{22} \end{bmatrix} &= \sqrt{\frac{k_i^2 \cdot (a_{21}^2 + a_{22}^2)}{a_{21}^2 + a_{22}^2}} \cdot \begin{bmatrix} a_{21} \\ a_{22} \end{bmatrix} \\
\begin{bmatrix} r_{21} \\ r_{22} \end{bmatrix} &= \sqrt{\frac{k_i^2 \cdot a_{21}^2 + k_i^2 \cdot a_{22}^2}{a_{21}^2 + a_{22}^2}} \cdot \begin{bmatrix} a_{21} \\ a_{22} \end{bmatrix} \\
\begin{bmatrix} r_{21} \\ r_{22} \end{bmatrix} &= \sqrt{\frac{(k_i \cdot a_{21})^2 + (k_i \cdot a_{22})^2}{a_{21}^2 + a_{22}^2}} \cdot \begin{bmatrix} a_{21} \\ a_{22} \end{bmatrix} \\
\begin{bmatrix} r_{21} \\ r_{22} \end{bmatrix} &= \sqrt{\frac{r_{21}^2 + r_{22}^2}{a_{21}^2 + a_{22}^2}} \cdot \begin{bmatrix} a_{21} \\ a_{22} \end{bmatrix} \\
\begin{bmatrix} r_{21} \\ r_{22} \end{bmatrix} &= \sqrt{r_{21}^2 + r_{22}^2 \cdot \frac{1}{a_{21}^2 + a_{22}^2}} \cdot \begin{bmatrix} a_{21} \\ a_{22} \end{bmatrix} \\
\begin{bmatrix} r_{21} \\ r_{22} \end{bmatrix} &= \sqrt{\frac{d_2^2 \cdot (r_{11}^2 + r_{12}^2)}{d_1^2} \cdot \frac{1}{a_{21}^2 + a_{22}^2}} \cdot \begin{bmatrix} a_{21} \\ a_{22} \end{bmatrix}
\end{aligned}
\tag{3}
$$

As an example, the following table shows results for an ASUS VN248H-P (23.8”, 1920x1080).

Resolution	**Aspect Ratio**	**Length of Main Diagonal**	**Pixels per Inch Disparity**
1920x1080	16:9	24”	0.7713260399447819
2048x1152	16:9	25”	1.4315811301375163
2176x1224	16:9	27”	0.0914164195490201
2304x1296	16:9	28”	1.8511824958674765
2432x1368	16:9	30”	0.4525112767676092
2560x1440	16:9	31”	2.1895706940368030
2560x1440	16:9	31.5”	0.6856231466175870
2560x1440	16:9	32”	0.7713260399447819
3840x2160	16:9	47.5”	0.1948613153544727
2432x1520	16:10	31”	0.0452453048187920
1880x1504	5:4	26”	0.0399026402703840
2240x1792	5:4	31”	0.0236519427314335
1888x1416	4:3	25.5”	0.0101051855307475

```
def best_match(odiag, ores, mdiag, masp=None, verbose=False):
    mdiag, mres, masp = match_monitor(odiag, ores, mdiag, masp=masp)
    oppi = calculate_ppi(odiag, ores)
    odim = calculate_side_length(odiag, ores)
    diff = float('inf')
    bmon = None
    for _mres in standard_resolutions(masp):
        _mppi = calculate_ppi(mdiag, _mres)
        if diff > abs(oppi - _mppi):
            diff = abs(oppi - _mppi)
            bmon = (mdiag, _mres, masp, diff, (diff*odim[0], diff*odim[1]))
            if verbose:
                print bmon
        else:
            return bmon

def match_monitor(odiag, ores, mdiag=None, mres=None, masp=None):
    if not any([mdiag, mres]):
        return None
    if not ((mdiag and isinstance(mdiag, (int, float)))
             or (mres and len(mres) == 2)
             or isinstance(odiag, (int, float))
             or len(ores) == 2):
        return None
    oppi = calculate_ppi(odiag, ores)
    if mdiag:
        if masp and len(masp) == 2:
            oasp = masp
        else:
            oasp = simplify_fraction(ores[0], ores[1])
        mres = float(mdiag**2*(ores[0]**2 + ores[1]**2)) / odiag**2
        mres = (mres*(1.0 / (oasp[0]**2 + oasp[1]**2)))**0.5
        mres = (mres*oasp[0], mres*oasp[1])
    else:
        mdiag = float(odiag*(mres[0]**2 + mres[1]**2)**0.5)
        mdiag = mdiag / (ores[0]**2 + ores[1]**2)**0.5
        oasp = simplify_fraction(mres[0], mres[1])
    return (mdiag, mres, oasp)

def calculate_ppi(diag, res):
    if (isinstance(diag, (int, float))
        and isinstance(res, (list, tuple))
        and len(res) > 1):
        return (res[0]**2 + res[1]**2)**(0.5) / float(diag)
    else:
        return None
```

```
def calculate_side_length(diag, asp):
    if (isinstance(diag, (int, float))
        and isinstance(asp, (list, tuple))
        and len(asp) > 1):
        asp = simplify_fraction(asp[0], asp[1])
        red = float(diag) / (asp[0]**2 + asp[1]**2)**0.5
        return (asp[0]*red, asp[1]*red)
    else:
        return None

def standard_resolutions(asp, limit=float('inf')):
    if not (isinstance(asp, (list, tuple))
            and len(asp) > 1):
        yield None
    x, y = asp[0]*8, asp[1]*8
    _x, _y = x, y
    while limit:
        yield (_x, _y)
        _x, _y = (_x + x, _y + y)
        limit -= 1

def simplify_fraction(n, d):
    if d == 0:
        return None
    c = _gcd(n, d)
    return (n / c, d / c)

def _gcd(a, b):
    while b:
        a, b = b, a % b
    return a
```