Generalised Mathematical Framework for Browser Position Calculation: Coordinate Transformations Across Multiple Display Configurations

Michael R. Malloy

May 17, 2025

Abstract

This paper establishes a comprehensive mathematical framework for calculating positions in different screen sizes, browser windows, and display scaling configurations. We define a system of coordinate spaces and derive the transformations between them, enabling accurate position calculation regardless of changes in screen dimensions, viewport sizes, browser window positions, or DPI scaling factors. We present canonical forms of these transformations along with their proofs, demonstrating how they can be applied to solve practical browser positioning problems. The framework provides a unified approach to coordinate translation, allowing developers to create responsive and adaptive applications that function consistently across diverse display environments.

1 Introduction

Modern web applications must operate in a diverse range of display environments characterised by varying screen dimensions, window sizes, and scaling factors. This variability presents significant challenges for the precise positioning of elements, especially when coordinates need to be accurately translated between different reference frames.

Building upon previous work in browser position calculation, this paper extends the mathematical framework to handle transformations across different screen sizes. Our extended framework provides a unified solution for calculating positions under any combination of:

- Screen dimensions (e.g., from 2560×1440 to 1920×1080)
- Browser window positions (location of browser on screen)
- Viewport dimensions (visible area within browser)
- DPI scaling factors (ratio of physical to logical pixels)

We establish coordinate systems that represent each reference frame, derive canonical transformations between them, and prove the mathematical properties of these transformations. The framework enables both forward and inverse transformations, allowing for complete conversion of coordinates under changing display conditions.

2 Coordinate Systems

We begin by defining five fundamental coordinate systems that form the basis of our framework.

Definition 2.1 (Normalised Coordinate System)**.** The normalised coordinate system $\mathbb{N}$ represents positions as ratios of screen dimensions, independent of the actual pixel counts.

- Domain: $\mathbb{N} \subset [0, 1]^2$
- For a point $\boldsymbol{p}_n \in \mathbb{N}$, we denote $\boldsymbol{p}_n = (x_n, y_n)$ where $0 \leq x_n, y_n \leq 1$

Definition 2.2 (Screen Coordinate Systems)**.** We define two screen coordinate systems:

- $\mathbb{S}_1$: Screen coordinates in the original display configuration
- $\mathbb{S}_2$: Screen coordinates in the target display configuration

Both are physical pixel coordinate systems with origin at the top-left corner of their respective screens.

- Domain: $\mathbb{S}_i \subset \mathbb{Z}^2$ where $i \in \{1, 2\}$
- For a point $\boldsymbol{p}_{s_i} \in \mathbb{S}_i$, we denote $\boldsymbol{p}_{s_i} = (x_{s_i}, y_{s_i})$

Definition 2.3 (Browser Coordinate System)**.** The browser coordinate system $\mathbb{B}$ is the physical pixel coordinate system with origin in the upper left corner of the browser window.

- Domain: $\mathbb{B} \subset \mathbb{Z}^2$
- For a point $\boldsymbol{p}_b \in \mathbb{B}$, we denote $\boldsymbol{p}_b = (x_b, y_b)$

Definition 2.4 (Logical Coordinate System)**.** The logical coordinate system $\mathbb{L}$ is used by the browser internally after scaling the DPI.

- Domain: $\mathbb{L} \subset \mathbb{Q}^2$ where $\mathbb{Q}$ is the set of rational numbers
- For a point $\boldsymbol{p}_l \in \mathbb{L}$, we denote $\boldsymbol{p}_l = (x_l, y_l)$

3 Key Parameters

Our framework depends on several key parameters that characterise the display configurations.

Definition 3.1 (Display Configuration Parameters)**.**

- $\boldsymbol{s}_1 = (s_{w1}, s_{h1}) \in \mathbb{Z}_+^2$: Original screen dimensions in physical pixels
- $\boldsymbol{s}_2 = (s_{w2}, s_{h2}) \in \mathbb{Z}_+^2$: Target screen dimensions in physical pixels
- $\boldsymbol{b}_1 = (b_{x1}, b_{y1}) \in \mathbb{S}_1$: Browser window position in original screen coordinates
- $\boldsymbol{b}_2 = (b_{x2}, b_{y2}) \in \mathbb{S}_2$: Browser window position in target screen coordinates
- $\boldsymbol{v}_1 = (v_{w1}, v_{h1}) \in \mathbb{Z}_+^2$: Viewport dimensions in logical pixels (original configuration)
- $\boldsymbol{v}_2 = (v_{w2}, v_{h2}) \in \mathbb{Z}_+^2$: Viewport dimensions in logical pixels (target configuration)
- $\sigma_1 \in \mathbb{R}^+$: Original DPI scaling factor
- $\sigma_2 \in \mathbb{R}^+$: Target DPI scaling factor

4 Transformation Operators

We now define the transformation operators between our coordinate systems, providing their canonical forms and proving their properties.

4.1 Screen Size Transformation

We first define the transformations between different screen sizes.

Definition 4.1 (Screen-to-Normalised Transformation)**.** The screen-to-normalised transformation $T_{S_i \to N} : \mathbb{S}_i \to \mathbb{N}$ maps a point from screen coordinates to normalized coordinates:

$$T_{S_i \to N}(\boldsymbol{p}_{s_i}) = \left(\frac{x_{s_i}}{s_{wi}}, \frac{y_{s_i}}{s_{hi}} \right) = \boldsymbol{p}_n \tag{1}$$

where $i \in \{1, 2\}$ indicates the screen configuration.

Definition 4.2 (Normalised-to-Screen Transformation)**.** The normalised-to-screen transformation $T_{N \to S_i} : \mathbb{N} \to \mathbb{S}_i$ maps a point from normalised coordinates to screen coordinates:

$$T_{N \to S_i}(\boldsymbol{p}_n) = (x_n \cdot s_{wi}, y_n \cdot s_{hi}) = \boldsymbol{p}_{s_i} \tag{2}$$

where $i \in \{1, 2\}$ indicates the screen configuration.

Theorem 4.3. *The transformations $T_{S_i \to N}$ and $T_{N \to S_i}$ are linear transformations.*

Proof. We prove this for $T_{S_i \to N}$. For any $\boldsymbol{p}_{s_i}, \boldsymbol{q}_{s_i} \in \mathbb{S}_i$ and scalar $\alpha \in \mathbb{R}$:

1. $T_{S_i \to N}(\boldsymbol{p}_{s_i} + \boldsymbol{q}_{s_i}) = \left(\frac{x_{p_{s_i}} + x_{q_{s_i}}}{s_{wi}}, \frac{y_{p_{s_i}} + y_{q_{s_i}}}{s_{hi}} \right) = \left(\frac{x_{p_{s_i}}}{s_{wi}} + \frac{x_{q_{s_i}}}{s_{wi}}, \frac{y_{p_{s_i}}}{s_{hi}} + \frac{y_{q_{s_i}}}{s_{hi}} \right) = T_{S_i \to N}(\boldsymbol{p}_{s_i}) + T_{S_i \to N}(\boldsymbol{q}_{s_i})$

2. $T_{S_i \to N}(\alpha \boldsymbol{p}_{s_i}) = \left(\frac{\alpha \cdot x_{p_{s_i}}}{s_{wi}}, \frac{\alpha \cdot y_{p_{s_i}}}{s_{hi}} \right) = \alpha \cdot \left(\frac{x_{p_{s_i}}}{s_{wi}}, \frac{y_{p_{s_i}}}{s_{hi}} \right) = \alpha \cdot T_{S_i \to N}(\boldsymbol{p}_{s_i})$

Since both properties are satisfied, $T_{S_i \to N}$ is a linear transformation. The proof for $T_{N \to S_i}$ follows in a similar way.

The canonical matrix representation of $T_{S_i \to N}$ is:

$$T_{S_i \to N}(\boldsymbol{p}_{s_i}) = \begin{pmatrix} \frac{1}{s_{wi}} & 0 \\ 0 & \frac{1}{s_{hi}} \end{pmatrix} \begin{pmatrix} x_{s_i} \\ y_{s_i} \end{pmatrix} \tag{3}$$

And for $T_{N \to S_i}$:

$$T_{N \to S_i}(\boldsymbol{p}_n) = \begin{pmatrix} s_{wi} & 0 \\ 0 & s_{hi} \end{pmatrix} \begin{pmatrix} x_n \\ y_n \end{pmatrix} \tag{4}$$

$\square$

Definition 4.4 (Direct Screen-to-Screen Transformation)**.** The direct transformation $T_{S_1 \to S_2} : \mathbb{S}_1 \to \mathbb{S}_2$ maps a point from the original screen coordinates to the target screen coordinates:

$$T_{S_1 \to S_2} = T_{N \to S_2} \circ T_{S_1 \to N} \tag{5}$$

For any point $\boldsymbol{p}_{s_1} \in \mathbb{S}_1$:

$$\begin{aligned} T_{S_1 \to S_2}(\boldsymbol{p}_{s_1}) &= T_{N \to S_2}(T_{S_1 \to N}(\boldsymbol{p}_{s_1})) && (6)\\ &= T_{N \to S_2}\left(\frac{x_{s_1}}{s_{w1}}, \frac{y_{s_1}}{s_{h1}}\right) && (7)\\ &= \left(\frac{x_{s_1}}{s_{w1}} \cdot s_{w2}, \frac{y_{s_1}}{s_{h1}} \cdot s_{h2}\right) && (8)\\ &= \left(x_{s_1} \cdot \frac{s_{w2}}{s_{w1}}, y_{s_1} \cdot \frac{s_{h2}}{s_{h1}}\right) && (9)\\ &= (x_{s_1} \cdot \alpha_x, y_{s_1} \cdot \alpha_y) && (10) \end{aligned}$$

where $\alpha_x = \frac{s_{w2}}{s_{w1}}$ and $\alpha_y = \frac{s_{h2}}{s_{h1}}$ are the scaling factors for width and height, respectively.

Corollary 4.5. *The transformation $T_{S_1 \to S_2}$ is a linear transformation.*

Proof. As the composition of two linear transformations ($T_{S_1 \to N}$ and $T_{N \to S_2}$), $T_{S_1 \to S_2}$ is also a linear transformation.

The canonical matrix representation is:

$$T_{S_1 \to S_2}(\boldsymbol{p}_{s_1}) = \begin{pmatrix} \alpha_x & 0 \\ 0 & \alpha_y \end{pmatrix} \begin{pmatrix} x_{s_1} \\ y_{s_1} \end{pmatrix} \tag{11}$$

where $\alpha_x = \frac{s_{w2}}{s_{w1}}$ and $\alpha_y = \frac{s_{h2}}{s_{h1}}$. □

4.2 Browser Window Transformations

Next, we define the transformations related to the position of the browser window.

Definition 4.6 (Screen-to-Browser Transformation)**.** The screen-to-browser transformation $T_{S_i \to B_i} : \mathbb{S}_i \to \mathbb{B}_i$ maps a point from screen coordinates to browser coordinates in configuration i:

$$T_{S_i \to B_i}(\boldsymbol{p}_{s_i}) = \boldsymbol{p}_{s_i} - \boldsymbol{b}_i = (x_{s_i} - b_{xi}, y_{s_i} - b_{yi}) = \boldsymbol{p}_{b_i} \tag{12}$$

Theorem 4.7. *The transformation $T_{S_i \to B_i}$ is an affine transformation.*

Proof. This transformation represents a translation of the vector $-\boldsymbol{b}_i$. Although it does not preserve the origin and is therefore not linear, it is an affine transformation.

The canonical form of an affine transformation is $T(\boldsymbol{x}) = A\boldsymbol{x} + \boldsymbol{c}$, where A is a linear transformation and $\boldsymbol{c}$ is a constant vector. For $T_{S_i \to B_i}$, we have:

$$T_{S_i \to B_i}(\boldsymbol{p}_{s_i}) = I\boldsymbol{p}_{s_i} - \boldsymbol{b}_i \tag{13}$$

where I is the identity matrix and $\boldsymbol{c} = -\boldsymbol{b}_i$ is the translation vector. □

4.3 DPI Scaling Transformations

Now we define transformations related to DPI scaling.

Definition 4.8 (Browser-to-Logical Transformation)**.** The browser-to-logical transformation $T_{B_i \to L_i} : \mathbb{B}_i \to \mathbb{L}_i$ maps a point from browser coordinates to logical coordinates in configuration i:

$$T_{B_i \to L_i}(\boldsymbol{p}_{b_i}) = \frac{1}{\sigma_i}\boldsymbol{p}_{b_i} = \left(\frac{x_{b_i}}{\sigma_i}, \frac{y_{b_i}}{\sigma_i}\right) = \boldsymbol{p}_{l_i} \tag{14}$$

Theorem 4.9. *The transformation $T_{B_i \to L_i}$ is a linear transformation.*

Proof. This transformation represents a uniform scaling by factor $\frac{1}{\sigma_i}$. It preserves both vector addition and scalar multiplication, making it a linear transformation.

The canonical matrix representation is:

$$T_{B_i \to L_i}(\boldsymbol{p}_{b_i}) = \begin{pmatrix} \frac{1}{\sigma_i} & 0 \\ 0 & \frac{1}{\sigma_i} \end{pmatrix} \begin{pmatrix} x_{b_i} \\ y_{b_i} \end{pmatrix} \tag{15}$$

□

4.4 Composite Transformations

We now define composite transformations that combine multiple individual transformations.

Definition 4.10 (Screen-to-Logical Transformation)**.** The screen-to-logical transformation $T_{S_i \to L_i} : \mathbb{S}_i \to \mathbb{L}_i$ maps a point from screen coordinates to logical coordinates in configuration i:

$$T_{S_i \to L_i} = T_{B_i \to L_i} \circ T_{S_i \to B_i} \tag{16}$$

For any point $\boldsymbol{p}_{s_i} \in \mathbb{S}_i$:

$$
\begin{aligned}
T_{S_i \to L_i}(\boldsymbol{p}_{s_i}) &= T_{B_i \to L_i}(T_{S_i \to B_i}(\boldsymbol{p}_{s_i})) && (17)\\
&= T_{B_i \to L_i}(\boldsymbol{p}_{s_i} - \boldsymbol{b}_i) && (18)\\
&= \frac{1}{\sigma_i}(\boldsymbol{p}_{s_i} - \boldsymbol{b}_i) && (19)\\
&= \frac{\boldsymbol{p}_{s_i} - \boldsymbol{b}_i}{\sigma_i} && (20)\\
&= \left(\frac{x_{s_i} - b_{xi}}{\sigma_i}, \frac{y_{s_i} - b_{yi}}{\sigma_i}\right) && (21)
\end{aligned}
$$

Theorem 4.11. *The transformation $T_{S_i \to L_i}$ is an affine transformation.*

Proof. As the composition of an affine transformation ($T_{S_i \to B_i}$) and a linear transformation ($T_{B_i \to L_i}$), $T_{S_i \to L_i}$ is an affine transformation.

The canonical form is:

$$
T_{S_i \to L_i}(\boldsymbol{p}_{s_i}) = A\boldsymbol{p}_{s_i} + \boldsymbol{c} \tag{22}
$$

where $A = \frac{1}{\sigma_i} I$ (a uniform scaling matrix) and $\boldsymbol{c} = -\frac{\boldsymbol{b}_i}{\sigma_i}$ (a translation vector). $\square$

Definition 4.12 (Complete Transformation)**.** The complete transformation $T_{S_1 \to L_2} : \mathbb{S}_1 \to \mathbb{L}_2$ maps a point from original screen coordinates to target logical coordinates:

$$
T_{S_1 \to L_2} = T_{S_2 \to L_2} \circ T_{S_1 \to S_2} \tag{23}
$$

For any point $\boldsymbol{p}_{s_1} \in \mathbb{S}_1$:

$$
\begin{aligned}
T_{S_1 \to L_2}(\boldsymbol{p}_{s_1}) &= T_{S_2 \to L_2}(T_{S_1 \to S_2}(\boldsymbol{p}_{s_1})) && (24)\\
&= T_{S_2 \to L_2}(x_{s_1} \cdot \alpha_x, y_{s_1} \cdot \alpha_y) && (25)\\
&= \frac{(x_{s_1} \cdot \alpha_x, y_{s_1} \cdot \alpha_y) - \boldsymbol{b}_2}{\sigma_2} && (26)\\
&= \left(\frac{x_{s_1} \cdot \alpha_x - b_{x2}}{\sigma_2}, \frac{y_{s_1} \cdot \alpha_y - b_{y2}}{\sigma_2}\right) && (27)
\end{aligned}
$$

Theorem 4.13. *The transformation $T_{S_1 \to L_2}$ is an affine transformation.*

Proof. As the composition of a linear transformation ($T_{S_1 \to S_2}$) and an affine transformation ($T_{S_2 \to L_2}$), $T_{S_1 \to L_2}$ is an affine transformation.

The canonical form is:

$$
T_{S_1 \to L_2}(\boldsymbol{p}_{s_1}) = A\boldsymbol{p}_{s_1} + \boldsymbol{c} \tag{28}
$$

where:

$$A = \begin{pmatrix} \frac{\alpha_x}{\sigma_2} & 0 \\ 0 & \frac{\alpha_y}{\sigma_2} \end{pmatrix} \tag{29}$$

and $\boldsymbol{c} = -\frac{\boldsymbol{b}_2}{\sigma_2}$. □

5 Invertibility of Transformations

An important property of our transformations is their invertibility, which allows us to move between coordinate systems in both directions.

Theorem 5.1. *All defined transformations are invertible, with the following inverse transformations:*

$$T^{-1}_{S_i \to N} = T_{N \to S_i} \tag{30}$$

$$T^{-1}_{S_i \to B_i}(\boldsymbol{p}_{b_i}) = \boldsymbol{p}_{b_i} + \boldsymbol{b}_i \tag{31}$$

$$T^{-1}_{B_i \to L_i}(\boldsymbol{p}_{l_i}) = \sigma_i \cdot \boldsymbol{p}_{l_i} \tag{32}$$

$$T^{-1}_{S_i \to L_i}(\boldsymbol{p}_{l_i}) = \sigma_i \cdot \boldsymbol{p}_{l_i} + \boldsymbol{b}_i \tag{33}$$

$$T^{-1}_{S_1 \to S_2}(\boldsymbol{p}_{s_2}) = \left(\frac{x_{s_2}}{\alpha_x}, \frac{y_{s_2}}{\alpha_y} \right) \tag{34}$$

$$T^{-1}_{S_1 \to L_2}(\boldsymbol{p}_{l_2}) = \left(\frac{\sigma_2 \cdot x_{l_2} + b_{x2}}{\alpha_x}, \frac{\sigma_2 \cdot y_{l_2} + b_{y2}}{\alpha_y} \right) \tag{35}$$

Proof. We prove invertibility by showing that the composition of each transformation with its inverse yields the identity transformation.

For $T_{S_i \to N}$ and $T_{N \to S_i}$:

$$T_{N \to S_i}(T_{S_i \to N}(\boldsymbol{p}_{s_i})) = T_{N \to S_i} \left(\frac{x_{s_i}}{s_{wi}}, \frac{y_{s_i}}{s_{hi}} \right) \tag{36}$$

$$= \left(\frac{x_{s_i}}{s_{wi}} \cdot s_{wi}, \frac{y_{s_i}}{s_{hi}} \cdot s_{hi} \right) \tag{37}$$

$$= (x_{s_i}, y_{s_i}) \tag{38}$$

$$= \boldsymbol{p}_{s_i} \tag{39}$$

Similar proofs can be constructed for each of the other inverse transformations, showing that in each case, the composition with the original transformation yields the identity transformation. □

This invertibility ensures that we can transform coordinates in any direction between our defined coordinate systems, a critical property for a complete coordinate transformation framework.

6 Application to Position Calculation

Now we apply our mathematical framework to the practical problem of calculating positions across different display configurations.

Theorem 6.1. *Given a point $\boldsymbol{p}_{s_1} = (x_{s_1}, y_{s_1})$ in the original screen coordinates, its representation in target logical coordinates $\boldsymbol{p}_{l_2}$ is:*

$$\boldsymbol{p}_{l_2} = \left(\frac{x_{s_1} \cdot \alpha_x - b_{x2}}{\sigma_2}, \frac{y_{s_1} \cdot \alpha_y - b_{y2}}{\sigma_2}\right) \tag{40}$$

where $\alpha_x = \frac{s_{w2}}{s_{w1}}$ and $\alpha_y = \frac{s_{h2}}{s_{h1}}$.

Proof. This follows directly from the definition of the complete transformation $T_{S_1 \to L_2}$. □

Theorem 6.2. *Given a point $\boldsymbol{p}_{l_2} = (x_{l_2}, y_{l_2})$ in the target logical coordinates, its representation in original screen coordinates $\boldsymbol{p}_{s_1}$ is:*

$$\boldsymbol{p}_{s_1} = \left(\frac{\sigma_2 \cdot x_{l_2} + b_{x2}}{\alpha_x}, \frac{\sigma_2 \cdot y_{l_2} + b_{y2}}{\alpha_y}\right) \tag{41}$$

where $\alpha_x = \frac{s_{w2}}{s_{w1}}$ and $\alpha_y = \frac{s_{h2}}{s_{h1}}$.

Proof. This follows from the inverse of the complete transformation $T^{-1}_{S_1 \to L_2}$ as defined in Theorem 5.1. □

7 Edge Detection Across Configurations

Finally, we apply our framework to determine the edges of the browser window in different screen configurations.

Theorem 7.1. *Given the browser window position $\boldsymbol{b}_i = (b_{xi}, b_{yi})$ in screen coordinates, viewport dimensions $\boldsymbol{v}_i = (v_{wi}, v_{hi})$ in logical pixels, and DPI scaling factor σ_i, the browser window's edges in screen coordinates in configuration i are:*

$$\textit{Top edge: } y = b_{yi} \tag{42}$$

$$\textit{Left edge: } x = b_{xi} \tag{43}$$

$$\textit{Right edge: } x = b_{xi} + \sigma_i \cdot v_{wi} \tag{44}$$

$$\textit{Bottom edge: } y = b_{yi} + \sigma_i \cdot v_{hi} \tag{45}$$

Theorem 7.2. *The browser window edges in the original screen coordinates can be mapped to the target screen coordinates as:*

$$\text{Top edge: } y = b_{y1} \cdot \alpha_y \tag{46}$$
$$\text{Left edge: } x = b_{x1} \cdot \alpha_x \tag{47}$$
$$\text{Right edge: } x = (b_{x1} + \sigma_1 \cdot v_{w1}) \cdot \alpha_x \tag{48}$$
$$\text{Bottom edge: } y = (b_{y1} + \sigma_1 \cdot v_{h1}) \cdot \alpha_y \tag{49}$$

Proof. This follows from applying the screen transformation $T_{S_1 \to S_2}$ to each edge equation from Theorem 7.1. □

8 Application to the Specific Problem

We now apply our extended framework to the original problem with additional screen size variation.

8.1 Original Parameters

- Original screen resolution: $\boldsymbol{s}_1 = (2560, 1440)$ pixels
- Viewport dimensions: $\boldsymbol{v}_1 = (2000, 1000)$ logical pixels
- DPI scaling factor: $\sigma_1 = 2$
- Target absolute position: $\boldsymbol{p}_{s_1} = (2065, 539)$ in original screen coordinates

8.2 Target Parameters

- Target screen resolution: $\boldsymbol{s}_2 = (1920, 1080)$ pixels
- Viewport dimensions: $\boldsymbol{v}_2 = (1800, 900)$ logical pixels
- DPI scaling factor: $\sigma_2 = 1.5$

8.3 Scaling Factors

$$\alpha_x = \frac{s_{w2}}{s_{w1}} = \frac{1920}{2560} = 0.75 \tag{50}$$
$$\alpha_y = \frac{s_{h2}}{s_{h1}} = \frac{1080}{1440} = 0.75 \tag{51}$$

8.4 Position Transformation

Assuming the browser window positions are $\boldsymbol{b}_1 = (100, 50)$ and $\boldsymbol{b}_2 = (75, 37.5)$ (scaled appropriately), the target position in the new screen coordinates is:

$$\begin{aligned}
\boldsymbol{p}_{s_2} &= T_{S_1 \to S_2}(\boldsymbol{p}_{s_1}) && (52)\\
&= (2065 \cdot 0.75, 539 \cdot 0.75) && (53)\\
&= (1548.75, 404.25) && (54)
\end{aligned}$$

And in the new logical coordinates:

$$\begin{aligned}
\boldsymbol{p}_{l_2} &= T_{S_1 \to L_2}(\boldsymbol{p}_{s_1}) && (55)\\
&= \left(\frac{2065 \cdot 0.75 - 75}{1.5}, \frac{539 \cdot 0.75 - 37.5}{1.5}\right) && (56)\\
&= \left(\frac{1548.75 - 75}{1.5}, \frac{404.25 - 37.5}{1.5}\right) && (57)\\
&= (982.5, 244.5) && (58)
\end{aligned}$$

8.5 Browser Window Edges

Original browser window edges:

$$\begin{aligned}
\text{Top edge: } & y = 50 && (59)\\
\text{Left edge: } & x = 100 && (60)\\
\text{Right edge: } & x = 100 + 2 \cdot 2000 = 4100 && (61)\\
\text{Bottom edge: } & y = 50 + 2 \cdot 1000 = 2050 && (62)
\end{aligned}$$

Transformed browser window edges:

$$\begin{aligned}
\text{Top edge: } & y = 50 \cdot 0.75 = 37.5 && (63)\\
\text{Left edge: } & x = 100 \cdot 0.75 = 75 && (64)\\
\text{Right edge: } & x = 4100 \cdot 0.75 = 3075 && (65)\\
\text{Bottom edge: } & y = 2050 \cdot 0.75 = 1537.5 && (66)
\end{aligned}$$

9 Conclusion

We have established a comprehensive mathematical framework for calculating positions across different screen sizes, browser windows, and display scaling configurations. Our framework provides a unified approach to coordinate translation, allowing for accurate positioning regardless of changes in the display environment.

The key contributions of this paper include:

- Definition of five coordinate systems (normalised, two screen systems, browser, and logical) that represent different reference frames
- Derivation of canonical transformations between these coordinate systems, including their inverse transformations
- Proof of the mathematical properties of these transformations, particularly their linearity/affinity and invertibility
- Application of the framework to practical problems such as position calculation and edge detection across different display configurations

This framework enables developers to create responsive and adaptive applications that work consistently in diverse display environments, addressing the practical challenges of modern web development.