

# GEOMETRIC TOOLS

## Categories

[Home](#)  
[Contact Information](#)  
[Privacy Statement](#)  
  
[Books](#)  
[Downloads](#)  
[Documentation](#)

## Browse Source

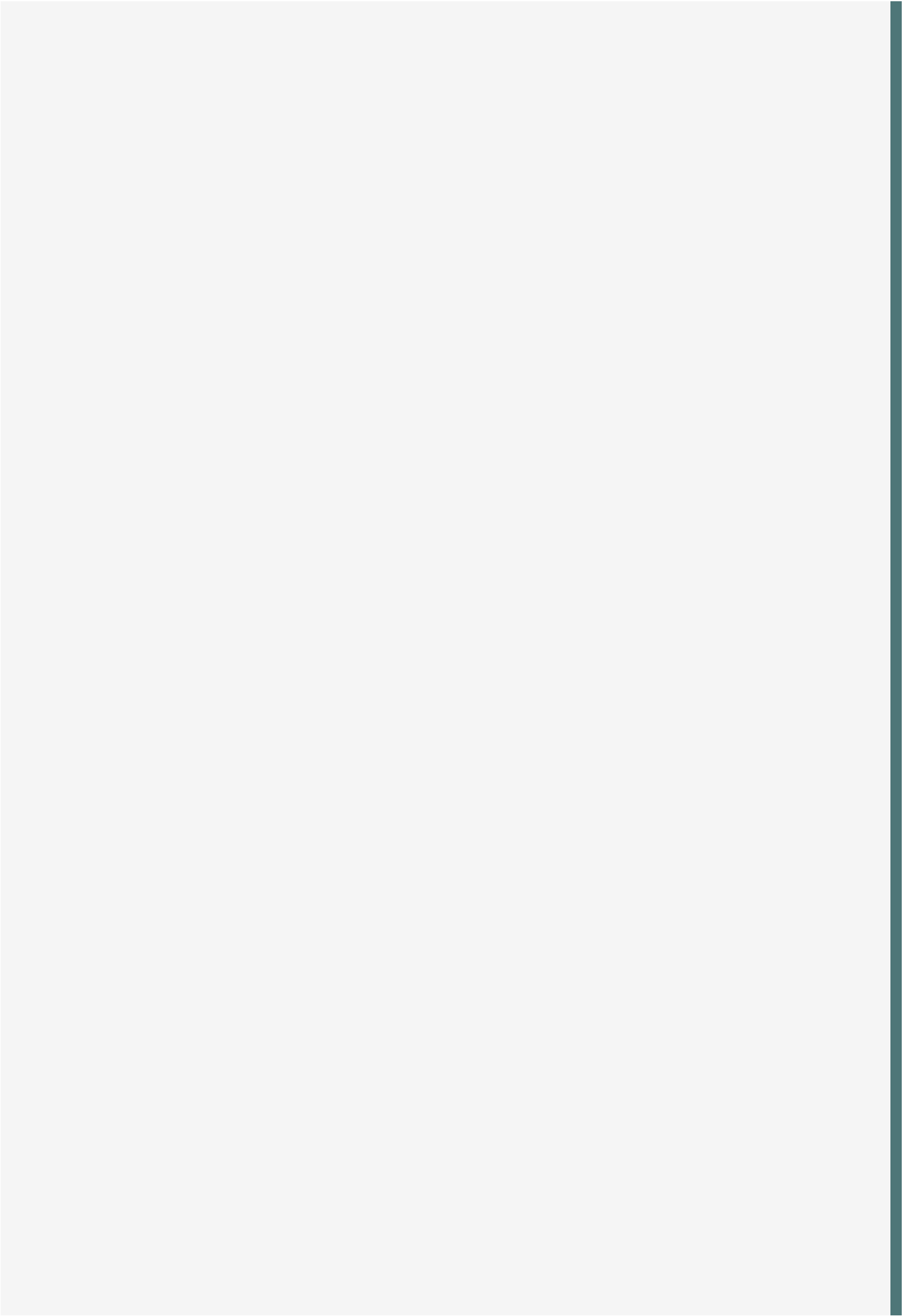
[Low Level](#)  
[Mathematics](#)  
[Graphics](#)  
[Imagics](#)  
[Physics](#)  
[Applications](#)

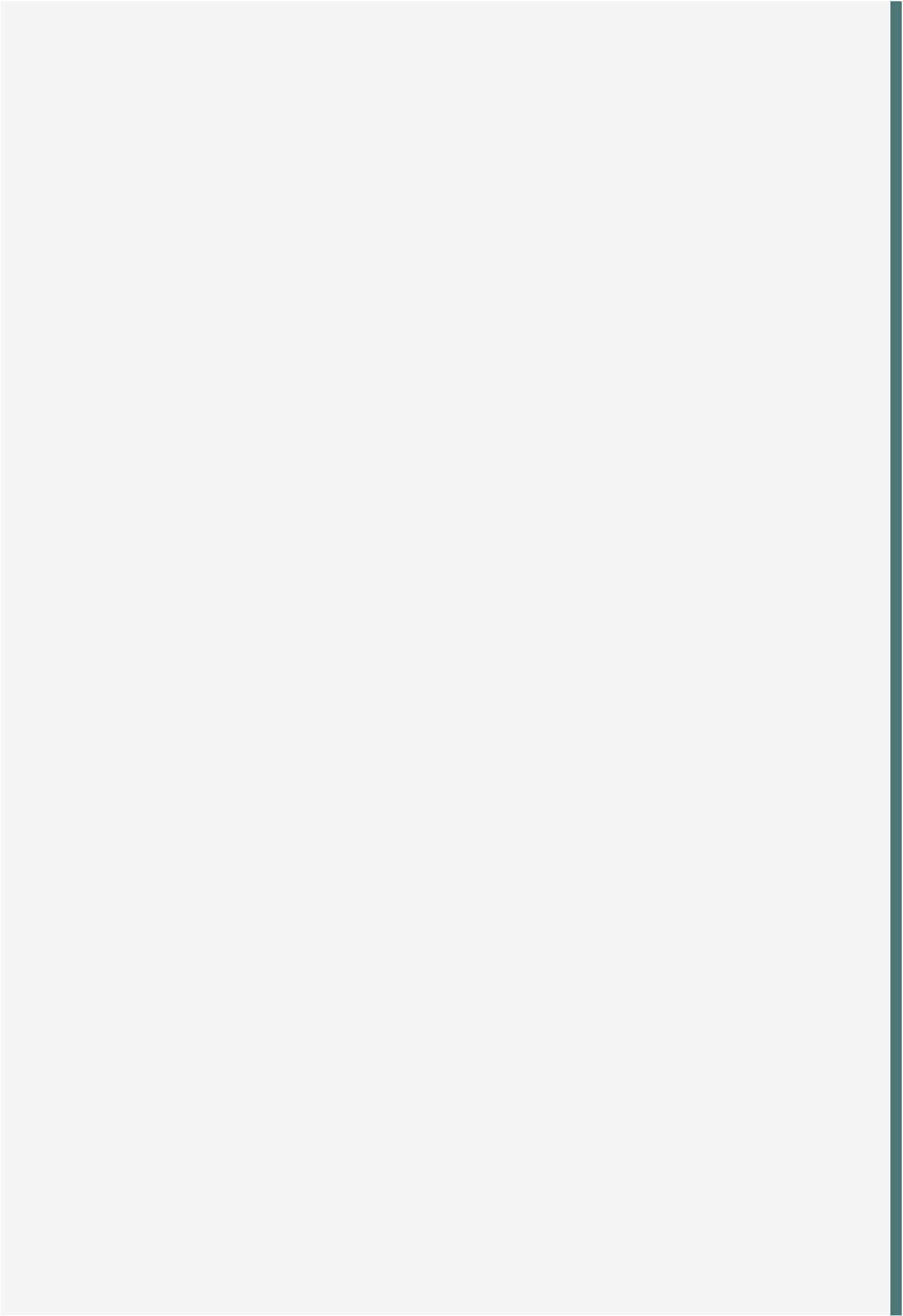
## Browse Samples

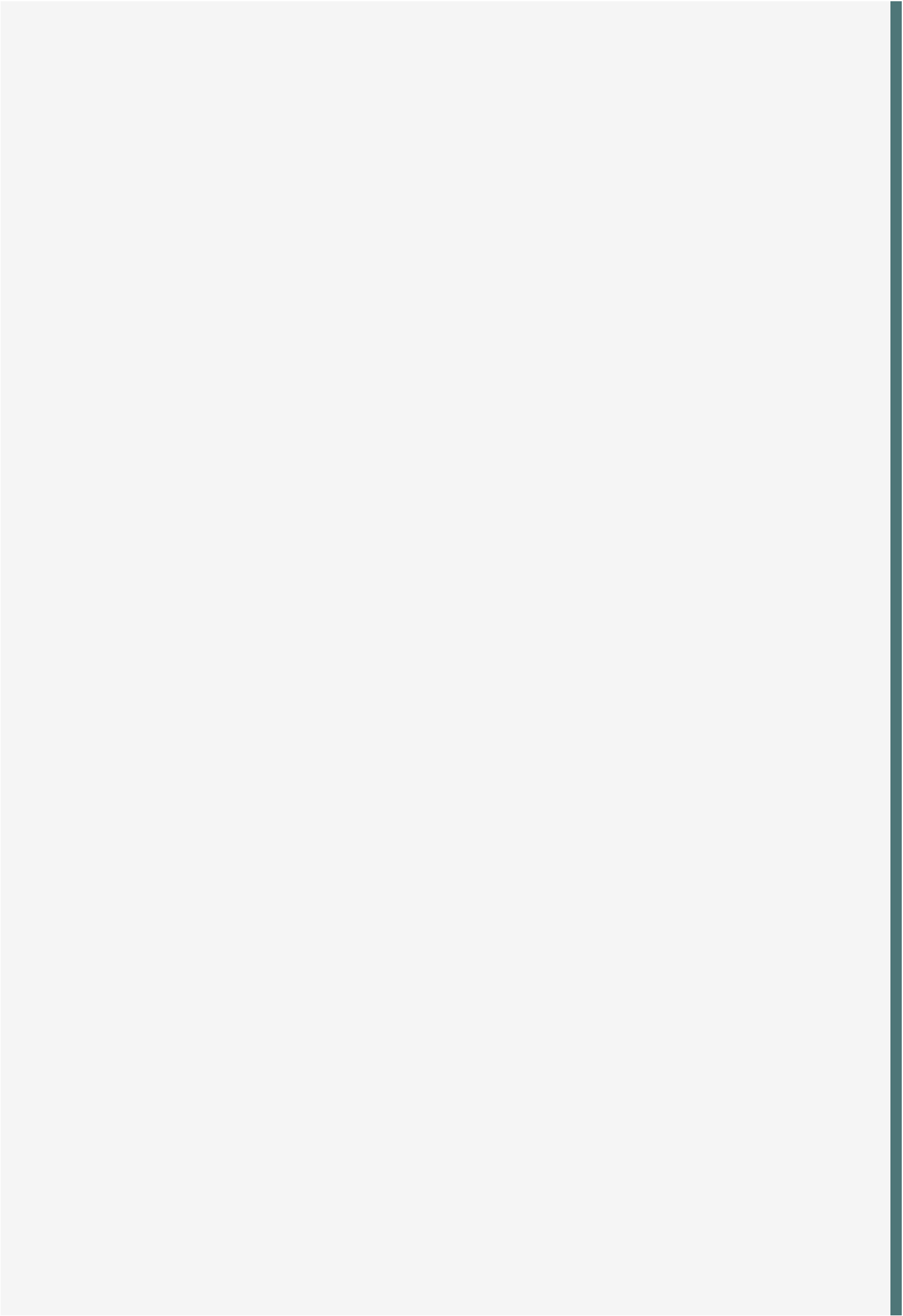
[Basics](#)  
[Geometrics](#)  
[Graphics](#)  
[Imagics](#)  
[Mathematics](#)  
[Physics](#)

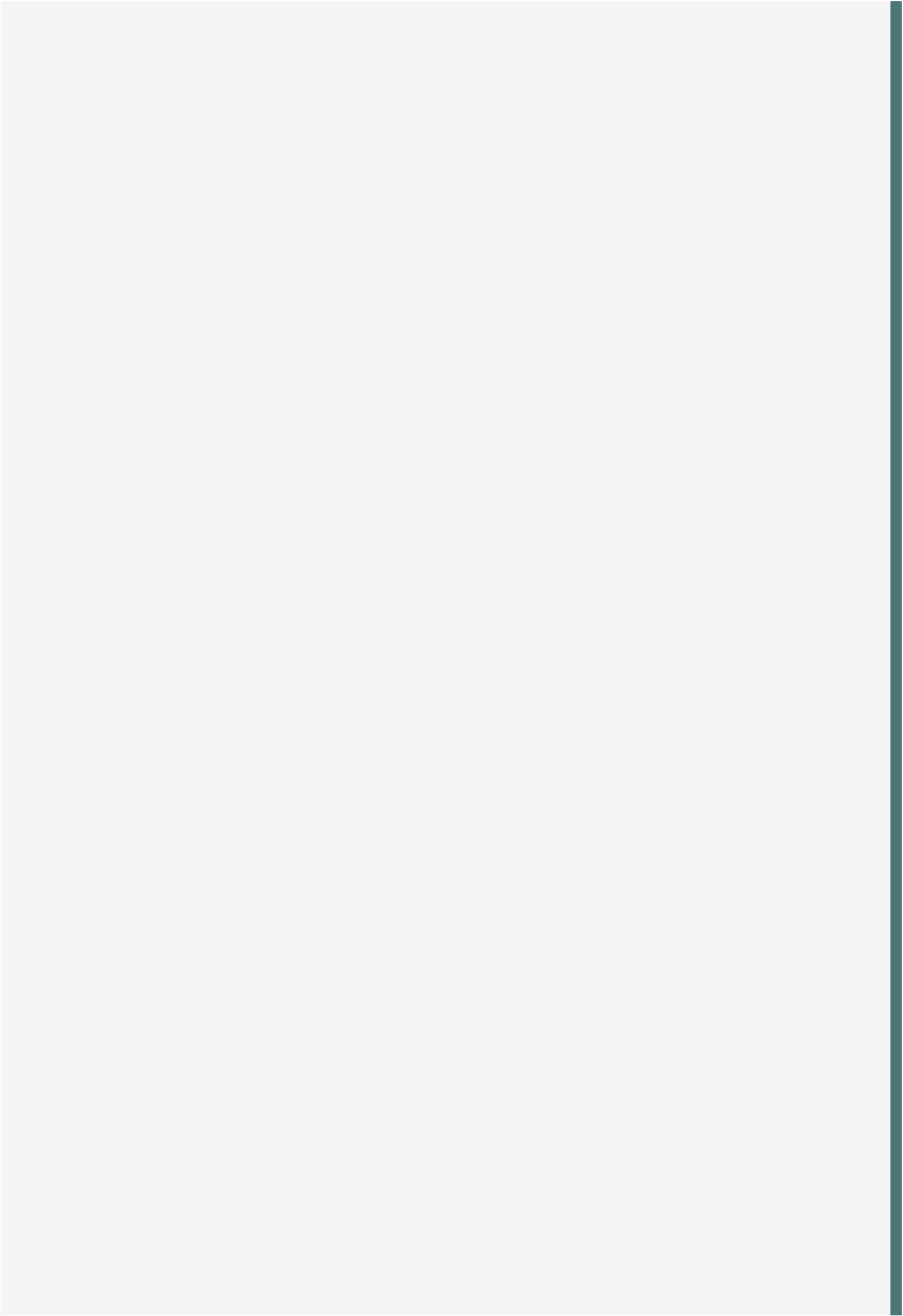
## Browse Tools

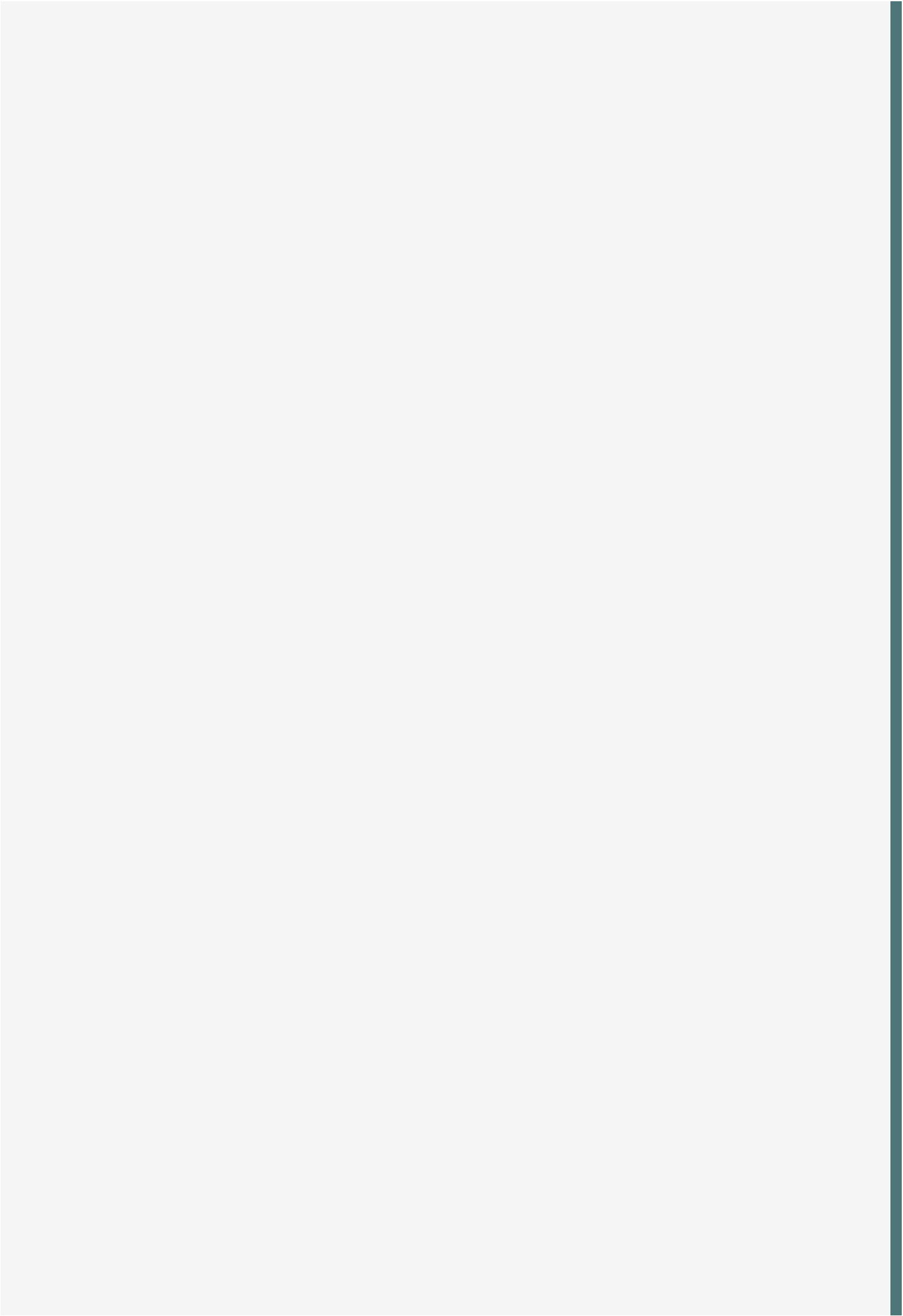
[Tools](#)

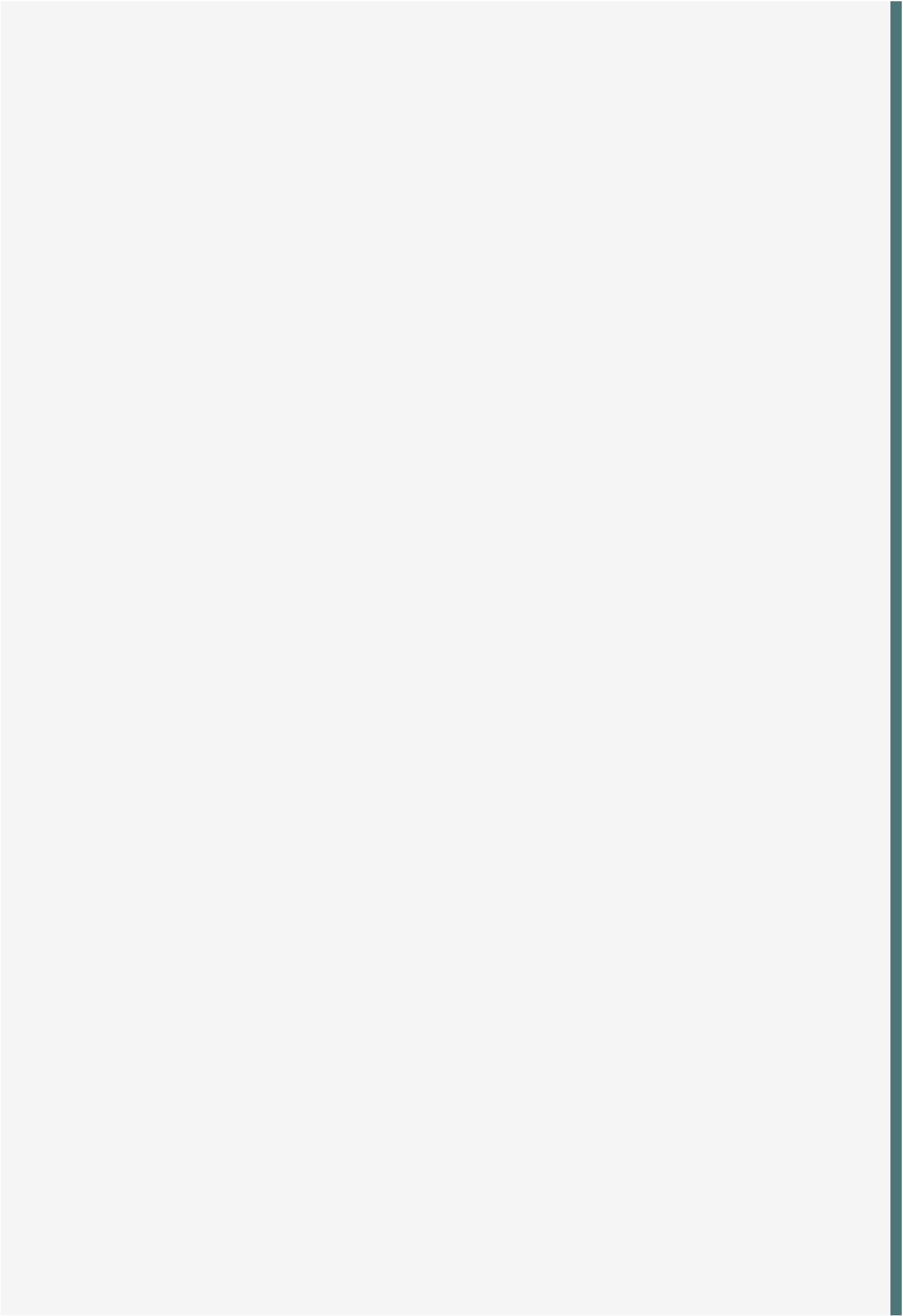


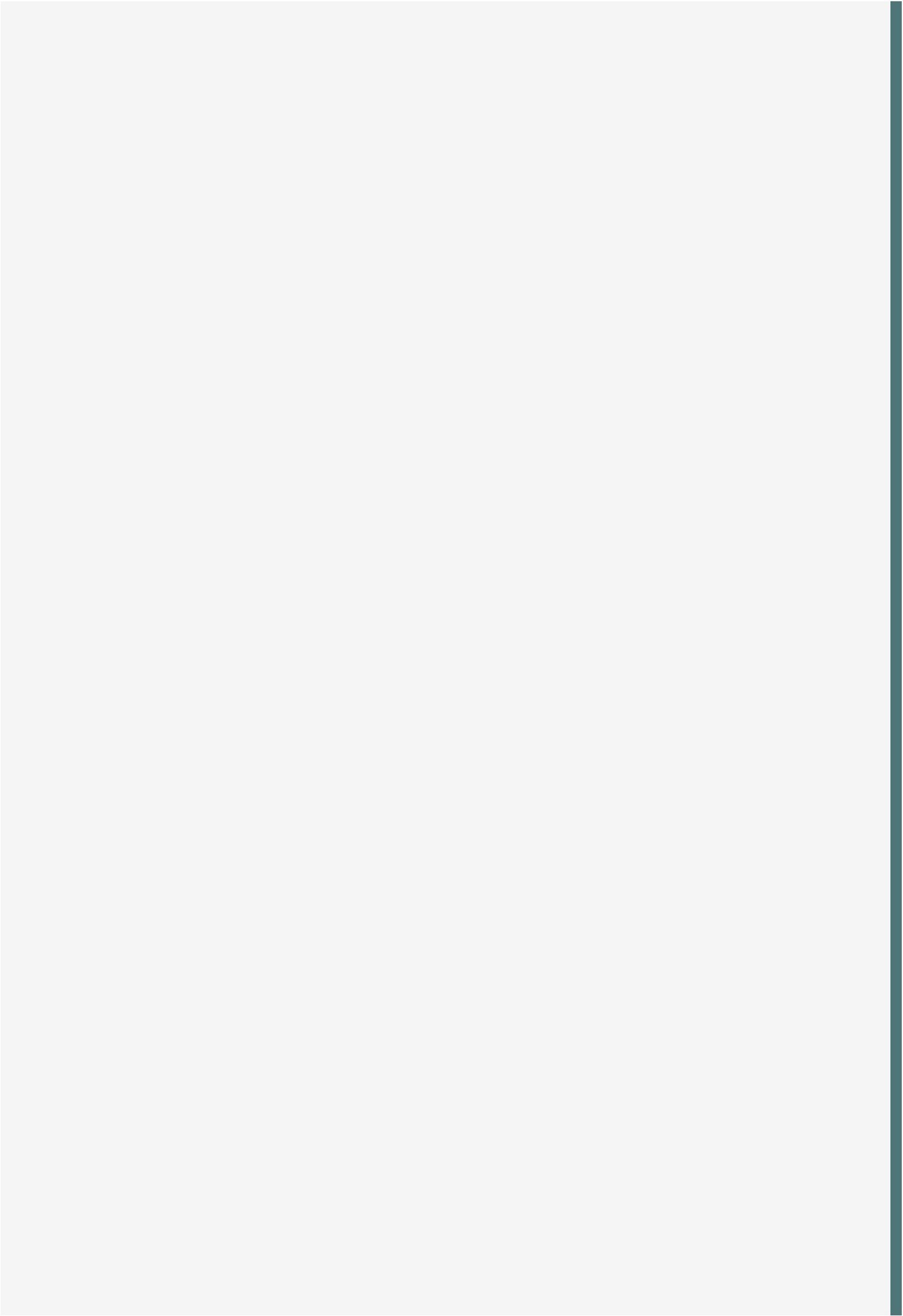




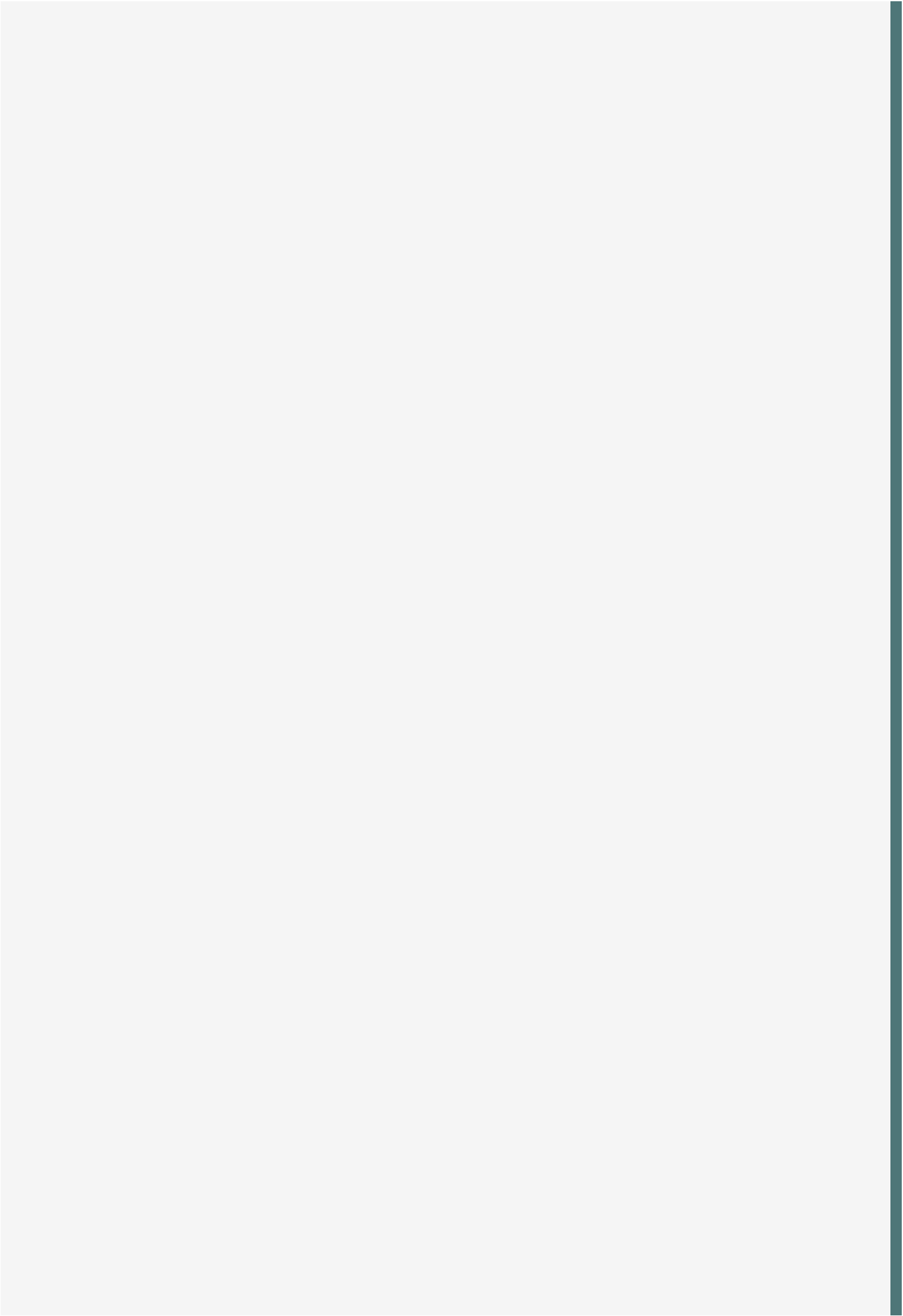


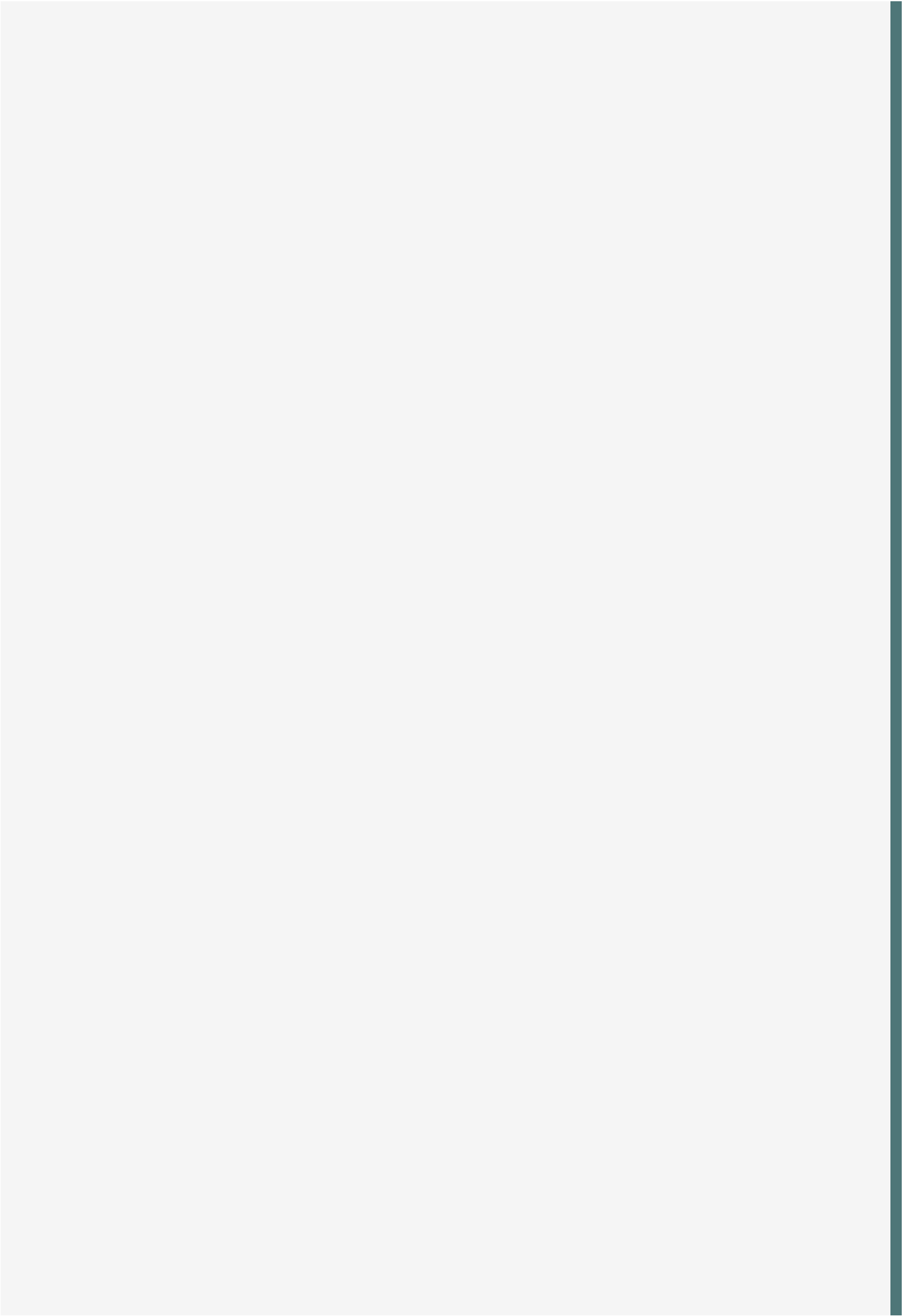


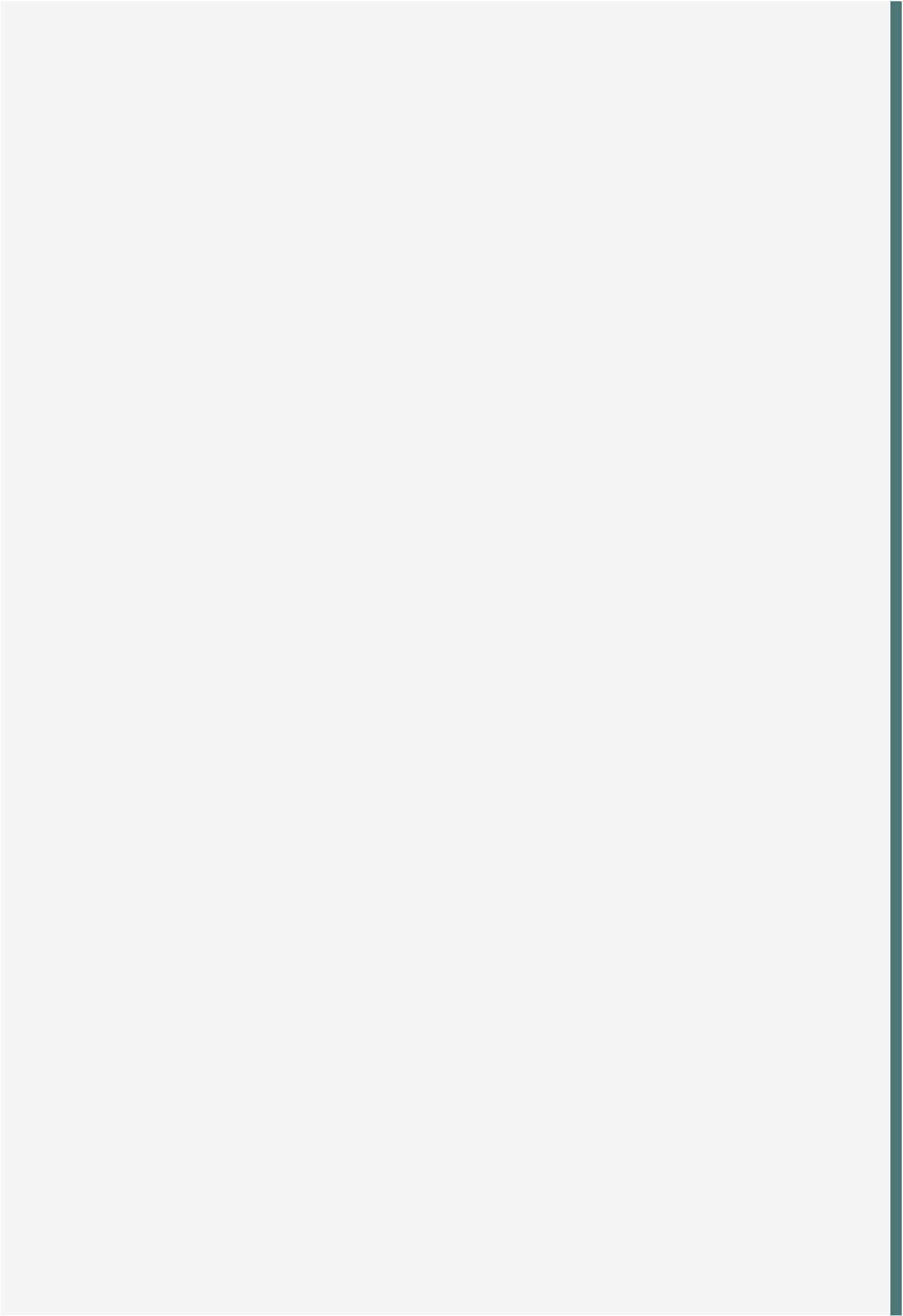












## Book Corrections for 3D Game Engine Design (2nd edition)

- [Book Corrections Organized by Date of Change](#)
- [Book Corrections Organized by Page Number](#)

## Book Corrections Organized by Date of Change

**22 January 2015, page 659.** The pseudocode

```
float sb1 = -tmpST/seg.e;
sClosest[0] = rectangle.e[0] * ((1-i1)*(2*i0-1)+i1*sb1);
sClosest[1] = rectangle.e[1] * ((1-i0)*(2*i1-1)+i0*sb1);
```

should be

```
float sb1 = tmpST/seg.e;
sClosest[0] = rectangle.e[0] * ((1-i1)*(2*i0-1)+i1*sb1);
sClosest[1] = rectangle.e[1] * (i1*(2*i0-1)+(1-i1)*sb1);
```

**27 December 2014, page 55.** The next-to-last paragraph has "Suffice it to say that my OpenGL and Direct3D renderers have been implemented to use the exact same view and projection matrices,...". The book shipped with Wild Magic 4.0 and used the [0,1] depth projection matrix (Direct3D). But this is not what the OpenGL rasterizer requires, so the source code is incorrect. The bug was fixed for Wild Magic 4.7.

**7 November 2014, page 805.** The two paragraphs after the pseudocode ("Although ... due to pointer aliasing.") are taken from the 1st edition of the book. In that edition, the decrement of the counter occurred before the increment, in which case these paragraphs apply. In the 2nd edition, the smart pointer code had been rewritten to use the increment first and the decrement second, so the paragraphs are irrelevant.

**7 November 2014, page 808.** Two of the paragraphs on the page mention calls to the "copy constructor". These should be calls to the "constructor". In the 1st edition of the book, the objects were smart pointers already, so in the code for that book, in fact the copy constructors were called.

**14 August 2014.** This is a large batch of changes thanks to the keen eye of Fabricio Valgrande Ambrosio, sent to me as an 11-page PDF with 101 potential issues.

page 90: The second line of pseudocode has **ymin = last maximum** ... but should be **ymax = last maximum** ...

pages 399-400: Page 399 has four bold **E** but should be **H**. Page 400 has one similar replacement.

page 400: Line 5 from the top of the page has **... right child of the root** ... but should be **... left child of the root** ...

page 413: In the discussion,  $\sigma = (V \cdot D)/|D|^2$  should be  $\sigma = (V \cdot D)/|D|$ .

page 423: The vector **W<sub>(0)</sub>** should be **V<sub>(0)</sub>**.

page 428: The vector **VW<sub>(0)</sub>** should be **V<sub>(0)</sub>**.

page 440: The middle line of the 3-line displayed equation at the bottom of the page has **- $\alpha_1 A_2 \times B_0$**  but should be **- $\alpha_1 A_2 \cdot B_0$** .

page 447: Figure 8.20 has 6 occurrences of **- $a_2/a_1$**  that should be **- $a_1/a_2$** .

page 448: The second displayed equation from the top has  **$c_{first} = p_0$**  ... but should be  **$c_{first} = p_1$**  ...

page 452: The pseudocode loop for Newton's method to find the contact time is missing a line of code at the end of the loop, after the if-block. The derivative needs to be updated, so the line should be **gder0 = GDer(obj0,vel0,obj1,vel1,t0);**

page 473: The last displayed equation has a division by  **$n^2 + x_v^2 + y_v^2$**  but should be the square root,  **$(n^2 + x_v^2 + y_v^2)^{1/2}$** .

page 497: The first paragraph has the phrase **... so in practice the 18n+11 is too large. For typical artist-generated models, n=64 is a reasonable choice.** This needs rewriting to **...so in practice the maximum number of bytes 299 is larger than expected. For typical artist-generated models, one expects approximately 64 bytes.**

page 499: Figure 8.39 is missing a blue dot that marks the intersection point **A<sub>1</sub>**. The edge **< V<sub>0</sub>, A<sub>1</sub> >** is drawn in black but should have been drawn in red.

page 508: In the first displayed equation of Section 9.1, the left-hand side has the acceleration  **$\ddot{X}$**  which is correct, but the last argument of the function on the right-hand side also has the acceleration. This argument needs to be the velocity  **$\dot{X}_i$** .

page 529: In the third line from the bottom, there is **(P<sub>1</sub> - P)** but should be **(P<sub>1</sub> - P<sub>0</sub>)**.

page 534: The last line of Exercise 10.1 has an equation with **y<sup>2</sup>** which should be **y<sub>2</sub>**.

page 537: The line before the second displayed equation has **V • (X-P)**. The first left parenthesis should be removed, **V • (X-P)**. The third line of the second displayed equation has **... V V<sup>T</sup> (X-P)**. A right parenthesis is missing, **... V V<sup>T</sup> (X-P)**.

page 538: The interval after the first displayed equation is **(0,π/2** and is missing a right parenthesis, **(0,π/2)**. The left-hand side of the second displayed equation has **D • (X-P)** but the vertical bar should not be there, **D • (X-P)**. The right-hand side of the third displayed equation has the same vertical bar that should be removed. The left-hand side of the third displayed equation has **... D D (X-P)** which is missing a transpose superscript, **... D D<sup>T</sup> (X-P)**.

page 550: The first line is **Replacing Equation (11.11) in Equation (11.8)...** which should be **Replacing Equations (11.10) and (11.11) in Equation (11.8)...**

page 556: The pseudocode has **basis[j][k] = c0 \* basis[j-1][k] \* flnvD0 + c1 \* basis[j-1][k+1]**; This should be **basis[j][k] = c0 \* basis[j-1][k] + c1 \* basis[j-1][k+1]**;

page 562: There are two occurrences of points using lower-case **p**, **p<sub>i</sub>** and **p<sub>i-1</sub>**. These should use upper-case **p**, **P<sub>i</sub>** and **P<sub>i-1</sub>**.

page 569: The displayed equation after Equation (11.27) has the term **δ X''(t)** (contains the second derivative) but should be **δ X'''(t)** (contains the third derivative).

page 584: The first paragraph of Section 12.6 has **[0,1]** as the domain for (u,v), but it should be the Cartesian product **[0,1]<sup>2</sup>**.

page 587: The first paragraph of Section 12.7 claims that the subdivision is discussed for cylinder, sphere, and ellipsoids, but in fact the section talks only about rectangle and triangle patches. However, if you have a tessellation of a cylinder, sphere, or ellipsoid, the subdivision algorithm for triangles applies to those meshes. I modified the text to **This section describes subdivision algorithms for rectangle and triangle patches**.

page 588: A displayed equation starts with **q<sub>i</sub>**. To be consistent with previous notation and that of the last displayed equation, this should use upper-case **q**, **Q<sub>i</sub>**.

pages 614-615: The first displayed equation on page 614 has equalities of lengths of vectors but the last term is squared radius, **r<sup>2</sup>**. This should be **r**. The same change should be made for the second displayed equation on page 615.

page 619: The first displayed equation has terms **V<sub>j</sub>** but should be **P<sub>j</sub>**.

page 620: The last displayed equation has term **X<sub>i</sub>** but should be **P<sub>i</sub>** to be consistent with the notation used previously in the section.

page 634: The second line of the first paragraph of Section 13.5.1 defines the cylinder radius using upper-case **r**, **R**, but it should be lower-case **r**, **r** because later in the paragraph a reference is made to the radius using lower case.

page 637: Line 2 has the factorization **M = R D R<sup>T</sup>**. In the subsection *Fitting Points with a Gaussian Distribution*, the factorization for **M** involves two terms written as **R<sup>T</sup> D R**. These should be modified to **M = R D R<sup>T</sup>**.

page 640: The last line of the multiline Equation (14.1) has a term **(I - D D)** but should be **(I - D D<sup>T</sup>)**

page 642: The last displayed equation starts with a column vector (s,t). The last equality in that line is missing the multiplier **1 / δ** that was included in the left-hand side of that equality.

page 650: The pseudocode has a line **t = (b1 >= 0 ? 0 : (-b1 >= a11 ? 1 : -b1/a11))**; but should be **t = (b1 >= 0 ? 0 : (-b1 >= a11 ? 1 : -b1/a11))**; [The last b-term had an extra 1-subscript.]

pages 651-652: The last displayed equation of page 651 has **b<sub>1</sub> E<sub>1</sub> + b<sub>2</sub> E<sub>2</sub>** but should be **b<sub>1</sub> E<sub>0</sub> + b<sub>2</sub> E<sub>1</sub>**. The same replacement should be made in the first displayed equation of page 652.

page 656: There is an equation **P<sub>0</sub> = P + ...** that should be **P<sub>0</sub> = C + ...**

page 659: The pseudocode comments have **Or the line and triangle are rectangle**. This should say **Or the line and rectangle are parallel**. The pseudocode does not declare loop indices **i0** and **i1** as type *int*. The pseudocode has a line **tClosest = tmpT**; but should be **tClosest = tmpLT**;

page 660: The third line of the first paragraph for Section 14.6.3 has **rectiangle** but should be **rectangle**.

page 663: The pseudocode does not declare loop index **i** as type *int*.

pages 667-668: The pseudocode does not declare loop indices **i** and **j** as type *int*.

page 669: The pseudocode does not declare loop indices **i0**, and **i1**, or **j** as type *int*.

page 673: Counting from the bottom of the page, lines -6, -7, and -9 use the word **ellipse** but should use **ellipsoid**. MY NOTE: Sections 14.13.1 and 14.13.2 were motivated by an article in Graphics Gems IV that claims the largest root of the polynomial corresponds to the closest point. As I discovered later (when trying to formulate a robust algorithm), this is not always true! See

### Distance from a Point to an Ellipse, an Ellipsoid, or a Hyperellipsoid

page 676: The last paragraph before Section 14.13.5 has the phrase **equidistance from C** but should be **equidistance from P**.

page 683: The pseudocode has a line **Find(line,object,quantity,tvalue))**; which has an extra right parenthesis, it should be **Find(line,object,quantity,tvalue)**;

page 694: Figure 15.6 has a 2-by-3 array of graphs which the pseudocode refers to by letters; for example, by Figure 15.6 (a). The graphs are not labeled. The top row of graphs from left to right are (a), (b), and (c). The bottom row of graphs from left to right are (d), (e), and (f).

page 695: The paragraph before the last block of pseudocode has **t = (e<sub>0</sub> - x<sub>p</sub>)/t<sub>d</sub>** but should be **t = (e<sub>0</sub> - x<sub>p</sub>)/x<sub>d</sub>**.

page 700: The pseudocode has **discr ≥ epsilon** but should have **discr ≥ 0**, and the comment **discr is effectively zero** should be **discr is zero**.

page 705: The statement is made that the line is parallel to the capsule axis when **|z<sub>d</sub>| = 1**. To clarify, this statement implies **x<sub>d</sub> = y<sub>d</sub> = 0**, because the beginning of the section mentions that {U,V,D} is an orthonormal set, which implies **D = (x<sub>d</sub>, y<sub>d</sub>, z<sub>d</sub>)** is a unit-length vector, which further implies that if a component has magnitude 1, the other components must be zero.

page 706: In the subsection *Line Not Parallel to Capsule Axis*, there is an equation  $x^2 + y_2 = r^2$  but should be  $x^2 + y^2 = r^2$ . In the subsection *Line Not Parallel to Capsule Axis*, the first displayed equation has right-hand side  $a_2t^2 + 2a_1t + a_0$  but should be  $a^2t^2 + 2a_1t + a_0$ .

page 709: The right-hand side of the first displayed equation has the term  $D^T(2AP+B)$  but is missing a t-value and should be  $D^T(2AP+B)t$ .

page 711: The right-hand side of the first displayed equation has the term  $\sigma_1 e_1 U_1$  but the sigma term should not be there,  $e_1 U_1$ .

page 712: The variable  $r$  is not declared in the first block of pseudocode; it should be a floating-point type.

page 716: The first displayed equation of Section 15.7.7 has  $P$  but should be  $C$ .

page 717: The classification of the projection intervals is correct but the pseudocode is incorrect. It should be the following, where the return conditions are equivalent to  $\text{Dot}(\text{plane.N}, X) < \text{plane.d}$  for all cone points  $X$ .

```
bool Culled(Cone cone, Plane plane)
{
    float dotNC = Dot(plane.N, cone.C);
    float dotND = Dot(plane.N, cone.D);
    float dotNPerpD = sqrt(1 - dotND * dotND);
    if (dotNPerpD >= cone.cosTheta)
    {
        float dotNK = dotNC + cone.h * dotND;
        float s = cone.h * cone.tanTheta * dotNPerpD;
        return dotNK + s < plane.d;
    }
    else if (dotND > 0)
    {
        float s = cone.h * cone.tanTheta * dotNPerpD;
        float r = cone.h * abs(dotND) + s;
        return dotNC + r < plane.d;
    }
    else // dotND < 0
    {
        return dotNC < plane.d;
    }
}
```

page 725: The sentence below the second displayed equation has  $(Y_i - d_i D)^2$  but should be  $|Y_i - d_i D|^2$ . The third displayed equation has a term  $R N$  on the right-hand side that should be deleted. The last displayed equation has an upper limit of summation for the y-terms of  $n$ , but it should be  $m$ .

pages 728-729: The third displayed equation on page 728 is missing an upper limit of summation  $m$ . The same missing limit occurs on page 729 in the third displayed equation from the bottom of the page.

page 736: The last line of the block of pseudocode has  $\backslash}$  but should be  $}$ .

page 744: The first displayed equation has a term  $2^{f-1}$  but should be  $2^{f-c}$ .

page 747: In section 16.7.1, the range of the function is  $R$  (denoting the real numbers) but should be  $R^n$  (denoting the n-tuples whose components are real numbers).

page 750: The first equation is  $C_1 = F(t_i, X_n)$  but should be  $C_1 = F(t_i, X_i)$ . The equation of step 3 is ambiguous because  $i$  is used to denote the index of a component of a vector but previously  $i$  referred to an iteration of the numerical method. Instead,  $\Delta = (1/\epsilon) \max_r |X_{\text{half},r} - X_{\text{full},r}|/h$   $F_r(t_i, X_i) + \epsilon_0$ , where  $X_{\text{half},r}$  is the  $r$ -th component of the vector  $X_{\text{half}}$  at iteration  $i$ ,  $X_{\text{full},r}$  is the  $r$ -th component of the vector  $X_{\text{full}}$  at iteration  $i$ , and  $F_r(t_i, X_i)$  is the  $r$ -th component of the vector  $F(t_i, X_i)$  at iteration  $i$ . The maximum is computed over all component indices  $r$ . In the sentence after the equation, similarly change  $F_i(t_i, x_i)$  to  $F_r(t_i, X_i)$ . In step 5, change  $x_n$  to  $X_i$ .

page 753: An equation near the bottom of the page has  $V_0$  but should be  $P_0$ .

page 757: The displayed equation for the partial derivative of  $E$  has  $[p(z;a - \tan^{-1}(z))]$  but should be  $[p(z;a) - \tan^{-1}(z)]$ . In the equation after that, the second integral has an upper limit of  $2$  but it should be  $1$ . The coefficients and errors in Table 16.3 do not look right for  $n = 4$  or  $5$ . Implemented code to solve the linear systems and reran the experiments.

n	a0	a1	a2	a3	a4	a5	maxerror
2	0.995983	-0.292281	0.083022				1.32470e-03
3	0.999317	-0.322288	0.149036	-0.040866			1.99441e-04
4	0.999883	-0.330600	0.181453	-0.087177	0.021869		3.08528e-05
5	0.999980	-0.332694	0.194021	-0.117696	0.054085	-0.012301	4.85847e-06

The maximum error occurs at  $z = 1$ . As pointed out in my GPGPU book, minimax approximations give better approximations than least-squares in the sense of balancing the error across the domain.

page 760: The first paragraph of Section 17.1.1 has  $W = U \times W$  but should be  $W = U \times V$ .

page 765: The right-hand side of the first line in Equation (17.13) has hat (^) symbols but the hats should be removed. The same change should be made to the right-hand side of the first line in Equation (17.14), and the minus sign (-) in this equation should be a plus sign (+). The

right-hand side of the second line of Equation (17.15) has three matrix terms. The lower-right entry of the last matrix has (0) but should be (1). The last sentence should say **where  $I_3$  is the  $3 \times 3$  identity matrix and  $R_3$  is the matrix in Equation (17.2).**

page 773: Unfortunately, the conversion from rotation matrix to quaternion is based on the vector-on-the-left convention for matrix-vector multiplication. In the book, the vector-on-the-right convention is used. In the first paragraph of Section 17.2.7, replace  $(r_{12} - r_{21}, r_{20} - r_{02}, r_{01} - r_{10})$  by  $(r_{21} - r_{12}, r_{02} - r_{20}, r_{10} - r_{01})$ . Replace  $x = (r_{12} - r_{21})/(4w)$  by  $x = (r_{21} - r_{12})/(4w)$ . Replace  $y = (r_{20} - r_{02})/(4w)$  by  $x = (r_{02} - r_{20})/(4w)$ . Replace  $z = (r_{01} - r_{10})/(4w)$  by  $x = (r_{10} - r_{01})/(4w)$ . In the second paragraph of Section 17.2.7: Replace  $w = (r_{12} - r_{21})/(4x)$  by  $w = (r_{21} - r_{12})/(4x)$ . Replace  $w = (r_{20} - r_{02})/(4y)$  by  $w = (r_{02} - r_{20})/(4y)$ . Replace  $w = (r_{01} - r_{10})/(4z)$  by  $x = (r_{10} - r_{01})/(4z)$ .

page 803: The first paragraph after the block of pseudocode has **PrintlnUset** but should be **PrintlnUse**.

pages 804-805: On page 804, the pseudocode for the second assignment operator has 4 occurrences of **rkPointer** in the body of the function but should be **rkReference**. A similar replacement must be made on page 805 in the bodies of the equality and inequality comparisons. A final replacement must be made in the text after the pseudocode, replace **rkPointer.m\_pkObject** by **rkReference.m\_pkObject**.

page 855: The code for the LookUp function has **-m\_fRotSpeed** but should have **+m\_fRotSpeed**. Similarly, the LookDown function has **+m\_fRotSpeed** but should be **-m\_fRotSpeed**. The actual source code has the correct values.

page 917: The first paragraph has an equation  $B = N \cdot T$  but should be  $B = N \times T$ .

page 926: The equation  $V = (P-V)/|P-V|$  should be  $V = (P-E)/|P-E|$ . The displayed equation  $R = E - 2(N \cdot E)N$  should be  $R = V - 2(N \cdot V)N$ .

pages 930-931: The pseudocode has a switch statement for which only one of the cases has a **break** statement. All cases should have **break** statements.

page 933: The middle line of Equation (20.3) has a term  $\sin(\varphi)$  but should have  $\sin(\theta)$ .

page 948: There is an equation  $E+tP$  that should be  $E+t(P-E)$ .

---

**04 June 2011, page 349.** Before the last displayed equation involving X, the sentence should start with "Observing that  $w_0 = 1 = \sum_{i=1}^{T-1} w_i$ ".

---

**16 March 2010, pages 35-36.** This is just a clarification. Figure 2.12 shows an arbitrary point P on the plane. Equation (2.42) is valid no matter how you choose P. On page 36, the closest point on the plane to the eyepoint is referred to using the equation  $P = E + d_{\min}D$ . This P is a specific point on the plane and is not the same P that appears in Figure 2.12. However, the planar point P closest to E is the point I substituted in Equation (2.42) to obtain the first displayed equation on page 36.

---

**14 March 2010, page 455.** A simple implementation of the dynamic collision detection system using spheres is **SphereColliders.zip**. The pseudocode of Section 8.3 needed some minor adjustments.

---

**07 March 2010, page 25.** Before the displayed equation for Y, there is a sentence "The component of X in the D direction is dX". This should say "... is dD".

---

**23 February 2009, page 447.** The discriminant should be

$$\Delta^2 - \rho = (1/4) * (p_1 - p_0 + q_1 - q_0)^2$$


---

**06 December 2008, page 739.** The first displayed equation is

$$\max\{|a_0|, 1+|a_1|, \dots, 1+|a_{n-1}|\} = 1 + \max\{|a_0|, \dots, |a_{n-1}|\}$$

but the last equality should be an inequality

$$\max\{|a_0|, 1+|a_1|, \dots, 1+|a_{n-1}|\} \leq 1 + \max\{|a_0|, \dots, |a_{n-1}|\}$$


---

**05 July 2008, page 325.** The GetKnot function should have "if (i >= n+1) return 1;".

---

**31 December 2007, page 36.** The first line should have "d = Dot(D,X-E), u = Dot(U,X-E), and r = Dot(R,X-E)".

---

**23 December 2007, page 23.** The first paragraph of the subsection "Reflection" has "reflection W of V" but should say "reflection U of V".

---

**21 July 2007, page 24.** The top paragraph correctly describes  $I - 2N \cdot \text{Transpose}(N)$  as the reflection matrix. Equation (2.23) should also use this formula:

$$\begin{aligned} R &= I - 2N \cdot \text{Transpose}(N) = \\ &\begin{Bmatrix} 1-2n_0n_0 & -2n_0n_1 & -2n_0n_2 \\ -2n_0n_1 & 1-2n_1n_1 & -2n_1n_2 \\ -2n_0n_2 & -2n_1n_2 & 1-2n_2n_2 \end{Bmatrix} \end{aligned}$$


---

**16 July 2007, pages 21-22.** The S matrices in the displayed equations after Equation (2.19), (2.20), and (2.21) each have an extraneous 1



entry. The diagonal entries for those matrices must all be zero.

**09 June 2007, page 205.** I made the statement "The parser is not complete. Currently, it does not support shader programs whose inputs are arrays." As it turns out, the parser does support arrays! For example, consider the Cg program stored in a file named `VertexColorArray.cg`:

```
void v_VertexColorArray
(
in float4      kModelPosition : POSITION,
in float3      kModelColor : COLOR,
out float4     kClipPosition : POSITION,
out float3     kVertexColor : COLOR,
uniform float3 kColorArray[2],
uniform float4x4 WVPMatrix)
{
kClipPosition = mul(kModelPosition,WVPMatrix);
kVertexColor = (kModelColor + kColorArray[0] + kColorArray[1])/3.0f;
}
```

The OpenGL output from compiling this with Cg1.5 is

```
!!ARBvp1.0
# cgc version 1.5.0014, build date Sep 18 2006 20:36:59
# command line args: -profile arbvp1
# source file: VertexColorArray.cg
#vendor NVIDIA Corporation
#version 1.5.0.14
#profile arbvp1
#program v_VertexColorArray
#semantic v_VertexColorArray.kColorArray
#semantic v_VertexColorArray.WVPMatrix
#var float4 kModelPosition : $vin.POSITION : POSITION : 0 : 1
#var float3 kModelColor : $vin.COLOR : COLOR0 : 1 : 1
#var float4 kClipPosition : $vout.POSITION : HPOS : 2 : 1
#var float3 kVertexColor : $vout.COLOR : COLOR : 3 : 1
#var float3 kColorArray[0] : : c[5] : 4 : 1
#var float3 kColorArray[1] : : c[6] : 4 : 1
#var float4x4 WVPMatrix : : c[1], 4 : 5 : 1
#const c[0] = 0.3333333
PARAM c[7] = { { 0.33333334 }, program.local[1..6] };
TEMP R0;
TEMP R1;
MUL R0, vertex.position.y, c[2];
MAD R0, vertex.position.x, c[1], R0;
ADD R1.xyz, vertex.color, c[5];
MAD R0, vertex.position.z, c[3], R0;
ADD R1.xyz, R1, c[6];
MAD result.position, vertex.position.w, c[4], R0;
MUL result.color.xyz, R1, c[0].x;
END
# 7 instructions, 2 R-reg
```

The parser does create user-defined variables named "kColorArray[0]" and "kColorArray[1]", so you can indeed hook up to them in an application. For example,

```
// Your class' storage for the shader constant kColorArray[0].
float m_afColor0[4];

// Your class' storage for the shader constant kColorArray[1].
float m_afColor1[4];
VertexShader* pkVShader = WM4_NEW VertexShader("VertexColorArray");
PixelShader* pkPShader = WM4_NEW PixelShader("PassThrough3");
ShaderEffect* pkEffect = WM4_NEW ShaderEffect(1);
pkEffect->SetVShader(0, pkVShader);
pkEffect->SetPShader(0, pkPShader);
m_pkRenderer->LoadResources(pkEffect);
Program* pkVProgram = pkEffect->GetVProgram(0);
pkVProgram->GetUC("kColorArray[0]")->SetDataSource(m_afColor0);
pkVProgram->GetUC("kColorArray[1]")->SetDataSource(m_afColor1);
```

Any changes to `m_afColor0` and `m_afColor1` will propagate to the vertex program.

**11 May 2007, page 124.** Equation (2.113) has  $(w_1*y_1 - w_0*y_0)$  in the denominator. This should be  $(w_1*i_1 - w_0*i_0)$ .

**22 March 2007, page 30.** The first displayed equation is "Y = MX + B". The symbol Y was already used on page 29 for the matrix representation of the linear part of the affine transformation, namely,  $Y = MX$ . The displayed equation on page 30 should use a different symbol, say "Z = MX + B", which is the matrix-vector representation of the affine transformation.

**07 March 2007, page 69.** The first displayed equation has the determinant of a 3x3 matrix. The first column is  $\{-r_0, -u_0, -d_0\}$ . The minus signs do not belong there. Instead, the first column should be  $\{+r_0, +u_0, +d_0\}$ .

**07 March 2007, page 70.** The first two displayed equations have minus signs in front of the determinants; these do not belong there. The third displayed equation has a matrix whose first column is  $\{-r_0, -u_0, -d_0\}$ . The minus signs do not belong there. Instead, the first column should be  $\{+r_0, +u_0, +d_0\}$ .

The displayed equation with  $\det(H\_proj \cdot M)$  has a minus sign starting the right-hand side of the equation; this does not belong there. The  $3 \times 3$  matrix on the right-hand side needs to be prefixed with "det". In the second line after that displayed equation, " $\det(H\_proj) \det(M) < 0$ " should be " $\det(H\_proj) \det(M) > 0$ ".

**07 March 2007, page 104.** The last case of the right-hand side of Equation (2.93) has " $N-1$ " when " $i > N$ ". This should be " $N$ " when " $i > N$ ".

## Book Corrections Organized by Page Number

**Section 2.2.1, pages 21-22.** The  $S$  matrices in the displayed equations after Equation (2.19), (2.20), and (2.21) each have an extraneous 1 entry. The diagonal entries for those matrices must all be zero.

**Section 2.2.1, page 23.** The first paragraph of the subsection "Reflection" has "reflection  $W$  of  $V$ " but should say "reflection  $U$  of  $V$ ".

**Section 2.2.1, page 24.** The top paragraph correctly describes  $I - 2 \cdot N \cdot \text{Transpose}(N)$  as the reflection matrix. Equation (2.23) should also use this formula:

```
R = I - 2 * N * Transpose(N) =
{
{ 1-2*n0*n0 ,   -2*n0*n1 ,   -2*n0*n2 },
{ -2*n0*n1 ,  1-2*n1*n1 ,   -2*n1*n2 },
{ -2*n0*n2 ,   -2*n1*n2 ,  1-2*n2*n2 }
}
```

**Section 2.2.1, page 25.** Before the displayed equation for  $Y$ , there is a sentence "The component of  $X$  in the  $D$  direction is  $dX$ ". This should say "... is  $dD$ ".

**Section 2.2.2, page 30.** The first displayed equation is " $Y = MX + B$ ". The symbol  $Y$  was already used on page 29 for the matrix representation of the linear part of the affine transformation, namely,  $Y = MX$ . The displayed equation on page 30 should use a different symbol, say " $Z = MX + B$ ", which is the matrix-vector representation of the affine transformation.

**Sections 2.2.3 and 2.2.4, pages 35-36.** This is just a clarification. Figure 2.12 shows an arbitrary point  $P$  on the plane. Equation (2.42) is valid no matter how you choose  $P$ . On page 36, the closest point on the plane to the eyepoint is referred to using the equation  $P = E + d_{\min} D$ . This  $P$  is a specific point on the plane and is not the same  $P$  that appears in Figure 2.12. However, the planar point  $P$  closest to  $E$  is the point  $I$  substituted in Equation (2.42) to obtain the first displayed equation on page 36.

**Section 2.2.4, page 36.** The first line should have " $d = \text{Dot}(D, X-E)$ ,  $u = \text{Dot}(U, X-E)$ , and  $r = \text{Dot}(R, X-E)$ ".

**Section 2.3.5, page 55.** The next-to-last paragraph has "Suffice it to say that my OpenGL and Direct3D renderers have been implemented to use the exact same view and projection matrices,...". The book shipped with Wild Magic 4.0 and used the  $[0,1]$  depth projection matrix (Direct3D). But this is not what the OpenGL rasterizer requires, so the source code is incorrect. The bug was fixed for Wild Magic 4.7.

**Section 2.4.2, page 69.** The first displayed equation has the determinant of a  $3 \times 3$  matrix. The first column is  $\{-r_0, -u_0, -d_0\}$ . The minus signs do not belong there. Instead, the first column should be  $\{+r_0, +u_0, +d_0\}$ .

**Section 2.4.2, page 70.** The first two displayed equations have minus signs in front of the determinants; these do not belong there. The third displayed equation has a matrix whose first column is  $\{-r_0, -u_0, -d_0\}$ . The minus signs do not belong there. Instead, the first column should be  $\{+r_0, +u_0, +d_0\}$ .

The displayed equation with  $\det(H\_proj \cdot M)$  has a minus sign starting the right-hand side of the equation; this does not belong there. The  $3 \times 3$  matrix on the right-hand side needs to be prefixed with "det". In the second line after that displayed equation, " $\det(H\_proj) \det(M) < 0$ " should be " $\det(H\_proj) \det(M) > 0$ ".

**Section 2.5.4, page 90.** The second line of pseudocode has **ymin = last maximum ...** but should be **ymax = last maximum ...**

**Section 2.6.3, page 104.** The last case of the right-hand side of Equation (2.93) has " $N-1$ " when " $i > N$ ". This should be " $N$ " when " $i > N$ ".

**Section 2.6.7, page 124.** Equation (2.113) has  $(w_1 \cdot y_1 - w_0 \cdot y_0)$  in the denominator. This should be  $(w_1 \cdot i_1 - w_0 \cdot i_0)$ .

**Section 3.4.4, page 205.** I made the statement "The parser is not complete. Currently, it does not support shader programs whose inputs are arrays." As it turns out, the parser does support arrays! For example, consider the Cg program stored in a file named VertexColorArray.cg:

```
void v_VertexColorArray
(
in float4      kModelPosition : POSITION,
in float3      kModelColor    : COLOR,
out float4     kClipPosition  : POSITION,
out float3     kVertexColor   : COLOR,
```

```
uniform float3    kColorArray[2],
uniform float4x4  WVPMatrix)
{
    kClipPosition = mul(kModelPosition,WVPMatrix);
    kVertexColor = (kModelColor + kColorArray[0] + kColorArray[1])/3.0f;
}
```

The OpenGL output from compiling this with Cg1.5 is

```
!!ARBvp1.0
# cgc version 1.5.0014, build date Sep 18 2006 20:36:59
# command line args: -profile arbvp1
# source file: VertexColorArray.cg
#vendor NVIDIA Corporation
#version 1.5.0.14
#profile arbvp1
#program v_VertexColorArray
#semantic v_VertexColorArray.kColorArray
#semantic v_VertexColorArray.WVPMatrix
#var float4 kModelPosition : $vin.POSITION : POSITION : 0 : 1
#var float3 kModelColor : $vin.COLOR : COLOR0 : 1 : 1
#var float4 kClipPosition : $vout.POSITION : HPOS : 2 : 1
#var float3 kVertexColor : $vout.COLOR : COL0 : 3 : 1
#var float3 kColorArray[0] : : c[5] : 4 : 1
#var float3 kColorArray[1] : : c[6] : 4 : 1
#var float4x4 WVPMatrix : : c[1], 4 : 5 : 1
#const c[0] = 0.3333333
PARAM c[7] = { { 0.33333334 }, program.local[1..6] };
TEMP R0;
TEMP R1;
MUL R0, vertex.position.y, c[2];
MAD R0, vertex.position.x, c[1], R0;
ADD R1.xyz, vertex.color, c[5];
MAD R0, vertex.position.z, c[3], R0;
ADD R1.xyz, R1, c[6];
MAD result.position, vertex.position.w, c[4], R0;
MUL result.color.xyz, R1, c[0].x;
END
# 7 instructions, 2 R-reg
```

The parser does create user-defined variables named "kColorArray[0]" and "kColorArray[1]", so you can indeed hook up to them in an application. For example,

```
// Your class' storage for the shader constant kColorArray[0].
float m_afColor0[4];

// Your class' storage for the shader constant kColorArray[1].
float m_afColor1[4];
VertexShader* pkVShader = WM4_NEW VertexShader("VertexColorArray");
PixelShader* pkPShader = WM4_NEW PixelShader("PassThrough3");
ShaderEffect* pkEffect = WM4_NEW ShaderEffect(1);
pkEffect->SetVShader(0, pkVShader);
pkEffect->SetPShader(0, pkPShader);
m_pkRenderer->LoadResources(pkEffect);
Program* pkVProgram = pkEffect->GetVProgram(0);
pkVProgram->GetUC("kColorArray[0]")->SetDataSource(m_afColor0);
pkVProgram->GetUC("kColorArray[1]")->SetDataSource(m_afColor1);
```

Any changes to m\_afColor0 and m\_afColor1 will propagate to the vertex program.

---

**Section 5.2.2, page 325.** The GetKnot function should have "if (i >= n+1) return 1;".

---

**Section 5.5, page 349.** Before the last displayed equation involving X, the sentence should start with "Observing that  $w_0 = 1 = \sum_{i=1}^T w_i$ ".

---

**Section 8.1.1, pages 399-400.** Page 399 has four bold E but should be H. Page 400 has one similar replacement.

---

**Section 8.1.1, page 400.** Line 5 from the top of the page has ... right child of the root ... but should be ... left child of the root ...

---

**Section 8.1.3, page 413.** In the discussion,  $\sigma = (V \cdot D)/|D|^2$  should be  $\sigma = (V \cdot D)/|D|$ .

---

**Section 8.1.3, page 423.** The vector  $W_{(0)}$  should be  $V_{(0)}$ .

---

**Section 8.1.3, page 428.** The vector  $VW_{(0)}$  should be  $V_{(0)}$ .

---

**Section 8.1.4, page 440.** The middle line of the 3-line displayed equation at the bottom of the page has  $-a_1 A_2 \times B_0$  but should be  $-a_1 A_2 \cdot B_0$ .

---

**Section 8.2.2, page 447.** Figure 8.20 has 6 occurrences of  $-a_2/a_1$  that should be  $-a_1/a_2$ .

**Section 8.2.2, page 447.** The discriminant should be

$$\text{delta}^2 - \text{rho} = (1/4) * (p_1 - p_0 + q_1 - q_0)^2$$

**Section 8.2.2, page 448.** The second displayed equation from the top has  $c_{\text{first}} = p_0 \dots$  but should be  $c_{\text{first}} = p_1 \dots$

**Section 8.2.3, page 452.** The pseudocode loop for Newton's method to find the contact time is missing a line of code at the end of the loop, after the if-block. The derivative needs to be updated, so the line should be `gDer0 = GDer(obj0,vel0,obj1,vel1,t0);`

**Section 8.3, page 455.** A simple implementation of the dynamic collision detection system using spheres is [SphereColliders.zip](#). The pseudocode of Section 8.3 needed some minor adjustments.

**Section 8.4.1, page 473.** page 473: The last displayed equation has a division by  $n^2 + x_v^2 + y_v^2$  but should be the square root,  $(n^2 + x_v^2 + y_v^2)^{1/2}$ .

**Section 8.5.7, page 497.** The first paragraph has the phrase ... so in practice the  $18n+11$  is too large. For typical artist-generated models,  $n=64$  is a reasonable choice. This needs rewriting to ...so in practice the maximum number of bytes 299 is larger than expected. For typical artist-generated models, one expects approximately 64 bytes.

**Section 8.5.7, page 499.** Figure 8.39 is missing a blue dot that marks the intersection point  $A_1$ . The edge  $\langle V_0, A_1 \rangle$  is drawn in black but should have been drawn in red.

**Section 9.1, page 508.** In the first displayed equation of Section 9.1, the left-hand side has the acceleration  $\ddot{X}$  which is correct, but the last argument of the function on the right-hand side also has the acceleration. This argument needs to be the velocity  $\dot{X}_i$ .

**Section 10.1, page 529.** In the third line from the bottom, there is  $(P_1 - P)$  but should be  $(P_1 - P_0)$ .

**Section 10.2, page 534.** The last line of Exercise 10.1 has an equation with  $y^2$  which should be  $y_2$ .

**Section 10.4.3, page 537.** The line before the second displayed equation has  $V \cdot ((X-P))$ . The first left parenthesis should be removed,  $V \cdot (X-P)$ . The third line of the second displayed equation has  $\dots V V^T (X-P)$ . A right parenthesis is missing,  $\dots V V^T (X-P)$ .

**Section 10.4.4, page 538.** The interval after the first displayed equation is  $(0, \pi/2$  and is missing a right parenthesis,  $(0, \pi/2)$ . The left-hand side of the second displayed equation has  $D \cdot (X-P)$  but the vertical bar should not be there,  $D \cdot (X-P)$ . The right-hand side of the third displayed equation has the same vertical bar that should be removed. The left-hand side of the third displayed equation has  $\dots D D (X-P)$  which is missing a transpose superscript,  $\dots D D^T (X-P)$ .

**Section 11.4, page 550.** The first line is **Replacing Equation (11.11) in Equation (11.8)...** which should be **Replacing Equations (11.10) and (11.11) in Equation (11.8)...**

**Section 11.5.2, page 556.** The pseudocode has `basis[j][k] = c0 * basis[j-1][k] * flnvD0 + c1 * basis[j-1][k+1];` This should be `basis[j][k] = c0 * basis[j-1][k] + c1 * basis[j-1][k+1];`

**Section 11.7, page 562.** There are two occurrences of points using lower-case  $p$ ,  $p_i$  and  $p_{i-1}$ . These should use upper-case  $p$ ,  $P_i$  and  $P_{i-1}$ .

**Section 11.8.4, page 569.** The displayed equation after Equation (11.27) has the term  $\delta X''(t)$  (contains the second derivative) but should be  $\delta X'''(t)$  (contains the third derivative).

**Section 12.6, page 584.** The first paragraph of Section 12.6 has  $[0,1]$  as the domain for  $(u,v)$ , but it should be the Cartesian product  $[0,1]^2$ .

**Section 12.7, page 587.** The first paragraph of Section 12.7 claims that the subdivision is discussed for cylinder, sphere, and ellipsoids, but in fact the section talks only about rectangle and triangle patches. However, if you have a tessellation of a cylinder, sphere, or ellipsoid, the subdivision algorithm for triangles applies to those meshes. I modified the text to **This section describes subdivision algorithms for rectangle and triangle patches.**

**Section 12.7.1, page 588.** A displayed equation starts with  $q_i$ . To be consistent with previous notation and that of the last displayed equation, this should use upper-case  $q$ ,  $Q_i$ .

**Section 13.1.2, pages 614-615.** The first displayed equation on page 614 has equalities of lengths of vectors but the last term is squared radius,  $r^2$ . This should be  $r$ . The same change should be made for the second displayed equation on page 615.

**Section 13.2.2, page 619.** The first displayed equation has terms  $V_j$  but should be  $P_j$ .

**Section 13.2.2, page 620.** The last displayed equation has term  $X_i$  but should be  $P_i$  to be consistent with the notation used previously in the section.

**Section 13.5.1, page 634.** The second line of the first paragraph of Section 13.5.1 defines the cylinder radius using upper-case  $r$ ,  $R$ , but it should be lower-case  $r$ ,  $r$  because later in the paragraph a reference is made to the radius using lower case.

**Section 13.6.2, page 637.** Line 2 has the factorization  $M = R D R^T$ . In the subsection *Fitting Points with a Gaussian Distribution*, the factorization for  $M$  involves two terms written as  $R^T D R$ . These should be modified to  $M = R D R^T$ .

**Section 14.1.1, page 640.** The last line of the multiline Equation (14.1) has a term  $(I - D D)$  but should be  $(I - D D^T)$

**Section 14.2.1, page 642.** The last displayed equation starts with a column vector  $(s, t)$ . The last equality in that line is missing the multiplier  $1/\delta$  that was included in the left-hand side of that equality.

**Section 14.3, page 650.** The pseudocode has a line  $t = (b1 \geq 0 ? 0 : (-b1 \geq a11 ? 1 : -b1/a11))$ ; but should be  $t = (b1 \geq 0 ? 0 : (-b1 \geq a11 ? 1 : -b1/a11))$ ; [The last b-term had an extra 1-subscript.]

**Section 14.4.1, pages 651-652.** The last displayed equation of page 651 has  $b_1 E_1 + b_2 E_2$  but should be  $b_1 E_0 + b_2 E_1$ . The same replacement should be made in the first displayed equation of page 652.

**Section 14.5, page 656.** There is an equation  $P_0 = P + \dots$  that should be  $P_0 = C + \dots$

**Section 14.6.1, page 659.** The pseudocode comments have *Or the line and triangle are rectangle*. This should say *Or the line and rectangle are parallel*. The pseudocode does not declare loop indices  $i0$  and  $i1$  as type *int*. The pseudocode has a line  $tClosest = tmpT$ ; but should be  $tClosest = tmpLT$ ;

**Section 14.6.1, page 659.** The pseudocode

```
float sb1 = -tmpST/seg.e;
sClosest[0] = rectangle.e[0] * ((1-i1)*(2*i0-1)+i1*sb1);
sClosest[1] = rectangle.e[1] * ((1-i0)*(2*i1-1)+i0*sb1);
```

should be

```
float sb1 = tmpST/seg.e;
sClosest[0] = rectangle.e[0] * ((1-i1)*(2*i0-1)+i1*sb1);
sClosest[1] = rectangle.e[1] * (i1*(2*i0-1)+(1-i1)*sb1);
```

**Section 14.6.2, page 660.** The third line of the first paragraph for Section 14.6.3 has *rectangle* but should be *rectangle*.

**Section 14.8, page 663.** The pseudocode does not declare loop index  $i$  as type *int*.

**Section 14.10, pages 667-668.** The pseudocode does not declare loop indices  $i$  and  $j$  as type *int*.

**Section 14.10, page 669.** The pseudocode does not declare loop indices  $i0$ , and  $i1$ , or  $j$  as type *int*.

**Section 14.13.2, page 673.** Counting from the bottom of the page, lines -6, -7, and -9 use the word *ellipse* but should use *ellipsoid*. MY NOTE: Sections 14.13.1 and 14.13.2 were motivated by an article in Graphics Gems IV that claims the largest root of the polynomial corresponds to the closest point. As I discovered later (when trying to formulate a robust algorithm), this is not always true! See

 *Distance from a Point to an Ellipse, an Ellipsoid, or a Hyperellipsoid*

**Section 14.13.4, page 676.** The last paragraph before Section 14.13.5 has the phrase *equidistance from C* but should be *equidistance from P*.

**Section 15.1, page 683.** The pseudocode has a line *Find(line,object,quantity,tvalue)*; which has an extra right parenthesis, it should be *Find(line,object,quantity,tvalue)*;

**Section 15.3.2, page 694.** Figure 15.6 has a 2-by-3 array of graphs which the pseudocode refers to by letters; for example, by Figure 15.6 (a). The graphs are not labeled. The top row of graphs from left to right are (a), (b), and (c). The bottom row of graphs from left to right are (d), (e), and (f).

**Section 15.3.2, page 695.** The paragraph before the last block of pseudocode has  $t = (e_0 - x_p)/t_d$  but should be  $t = (e_0 - x_p)/x_d$ .

**Section 15.4.1, page 700.** The pseudocode has  $discr \geq \epsilon$  but should have  $discr \geq 0$ , and the comment *discr is effectively zero* should be *discr is zero*.

**Section 15.5.1, page 705.** The statement is made that the line is parallel to the capsule axis when  $|z_d| = 1$ . To clarify, this statement implies  $x_d = y_d = 0$ , because the beginning of the section mentions that  $\{U, V, D\}$  is an orthonormal set, which implies  $D = (x_d, y_d, z_d)$  is a unit-length vector, which further implies that if a component has magnitude 1, the other components must be zero.

**Section 15.5.1, page 706.** In the subsection *Line Not Parallel to Capsule Axis*, there is an equation  $x^2 + y^2 = r^2$  but should be  $x^2 + y^2 = r^2$ . In the subsection *Line Not Parallel to Capsule Axis*, the first displayed equation has right-hand side  $a_2t^2 + 2a_1t + a_0$  but should be  $a^2t^2 + 2a_1t + a_0$ .

**Section 15.6, page 709.** The right-hand side of the first displayed equation has the term  $D^T(2AP+B)$  but is missing a  $t$ -value and should be  $D^T(2AP+B)t$ .

**Section 15.7.1, page 711.** The right-hand side of the first displayed equation has the term  $\sigma_1 e_1 U_1$  but the sigma term should not be there,  $e_1 U_1$ .

**Section 15.7.1, page 712.** The variable  $r$  is not declared in the first block of pseudocode; it should be a floating-point type.

**Section 15.7.7, page 716.** The first displayed equation of Section 15.7.7 has  $P$  but should be  $C$ .

**Section 15.7.7, page 717.** The classification of the projection intervals is correct but the pseudocode is incorrect. It should be the following, where the return conditions are equivalent to  $\text{Dot}(\text{plane.N}, X) < \text{plane.d}$  for all cone points  $X$ .

```
bool Culled(Cone cone, Plane plane)
{
    float dotNC = Dot(plane.N, cone.C);
    float dotND = Dot(plane.N, cone.D);
    float dotNPerpD = sqrt(1 - dotND * dotND);
    if (dotNPerpD >= cone.cosTheta)
    {
        float dotNK = dotNC + cone.h * dotND;
        float s = cone.h * cone.tanTheta * dotNPerpD;
        return dotNK + s < plane.d;
    }
    else if (dotND > 0)
    {
        float s = cone.h * cone.tanTheta * dotNPerpD;
        float r = cone.h * abs(dotND) + s;
        return dotNC + r < plane.d;
    }
    else // dotND < 0
    {
        return dotNC < plane.d;
    }
}
```

**Section 16.3.2, page 725.** The sentence below the second displayed equation has  $(Y_i - d_i D)^2$  but should be  $|Y_i - d_i D|^2$ . The third displayed equation has a term  $R N$  on the right-hand side that should be deleted. The last displayed equation has an upper limit of summation for the  $y$ -terms of  $n$ , but it should be  $m$ .

**Section 16.3.5, pages 728-729.** The third displayed equation on page 728 is missing an upper limit of summation  $m$ . The same missing limit occurs on page 729 in the third displayed equation from the bottom of the page.

**Section 16.4.2, page 736.** The last line of the block of pseudocode has  $\backslash$  but should be  $}$ .

**Section 16.5.1, page 739.** The first displayed equation is

$$\max\{|a_0|, 1+|a_1|, \dots, 1+|a_{\{n-1\}}|\} = 1 + \max\{|a_0|, \dots, |a_{\{n-1\}}|\}$$

but the last equality should be an inequality

$$\max\{|a_0|, 1+|a_1|, \dots, 1+|a_{\{n-1\}}|\} \leq 1 + \max\{|a_0|, \dots, |a_{\{n-1\}}|\}$$

**Section 16.6.1, page 744.** The first displayed equation has a term  $2^{r-1}$  but should be  $2^{r-c}$ .

**Section 16.7.1, page 747.** In section 16.7.1, the range of the function is  $R$  (denoting the real numbers) but should be  $R^n$  (denoting the  $n$ -tuples whose components are real numbers).

**Section 16.7.1, page 750.** The first equation is  $C_1 = F(t_i, X_n)$  but should be  $C_1 = F(t_i, X_i)$ . The equation of step 3 is ambiguous because  $i$  is used to denote the index of a component of a vector but previously  $i$  referred to an iteration of the numerical method. Instead,  $\Delta = (1/\epsilon) \max_r |X_{\text{half},r} - X_{\text{full},r}| / |h F_r(t_i, X_i) + \epsilon_0|$ , where  $X_{\text{half},r}$  is the  $r$ -th component of the vector  $X_{\text{half}}$  at iteration  $i$ ,  $X_{\text{full},r}$  is the  $r$ -th component of the vector  $X_{\text{full}}$  at iteration  $i$ , and  $F_r(t_i, X_i)$  is the  $r$ -th component of the vector  $F(t_i, X_i)$  at iteration  $i$ . The maximum is computed over all component indices  $r$ . In the sentence after the equation, similarly change  $F_i(t_i, x_i)$  to  $F_r(t_i, X_i)$ . In step 5, change  $x_n$  to  $X_i$ .

**Section 16.7.2, page 753.** An equation near the bottom of the page has  $V_0$  but should be  $P_0$ .



**Section 16.8.3, page 757.** The displayed equation for the partial derivative of E has  $[p(z;a - \tan^{-1}(z))]$  but should be  $[p(z;a) - \tan^{-1}(z)]$ . In the equation after that, the second integral has an upper limit of 2 but it should be 1. The coefficients and errors in Table 16.3 do not look right for  $n = 4$  or 5. Implemented code to solve the linear systems and reran the experiments.

n	a0	a1	a2	a3	a4	a5	maxerror
2	0.995983	-0.292281	0.083022				1.32470e-03
3	0.999317	-0.322288	0.149036	-0.040866			1.99441e-04
4	0.999883	-0.330600	0.181453	-0.087177	0.021869		3.08528e-05
5	0.999980	-0.332694	0.194021	-0.117696	0.054085	-0.012301	4.85847e-06

The maximum error occurs at  $z = 1$ . As pointed out in my GPGPU book, minimax approximations give better approximations than least-squares in the sense of balancing the error across the domain.

**Section 17.1.1, page 760.** The first paragraph of Section 17.1.1 has  $W = U \times W$  but should be  $W = U \times V$ .

**Section 17.2, page 765.** The right-hand side of the first line in Equation (17.13) has hat (^) symbols but the hats should be removed. The same change should be made to the right-hand side of the first line in Equation (17.14), and the minus sign (-) in this equation should be a plus sign (+). The right-hand side of the second line of Equation (17.15) has three matrix terms. The lower-right entry of the last matrix has (0) but should be (1). The last sentence should say **where  $I_3$  is the  $3 \times 3$  identity matrix and  $R_3$  is the matrix in Equation (17.2).**

**Section 17.2.7, page 773.** Unfortunately, the conversion from rotation matrix to quaternion is based on the vector-on-the-left convention for matrix-vector multiplication. In the book, the vector-on-the-right convention is used. In the first paragraph of Section 17.2.7, replace  $(r_{12} - r_{21}, r_{20} - r_{02}, r_{01} - r_{10})$  by  $(r_{21} - r_{12}, r_{02} - r_{20}, r_{10} - r_{01})$ . Replace  $x = (r_{12} - r_{21})/(4w)$  by  $x = (r_{21} - r_{12})/(4w)$ . Replace  $y = (r_{20} - r_{02})/(4w)$  by  $x = (r_{02} - r_{20})/(4w)$ . Replace  $z = (r_{01} - r_{10})/(4w)$  by  $x = (r_{10} - r_{01})/(4w)$ . In the second paragraph of Section 17.2.7: Replace  $w = (r_{12} - r_{21})/(4x)$  by  $w = (r_{21} - r_{12})/(4x)$ . Replace  $w = (r_{20} - r_{02})/(4y)$  by  $w = (r_{02} - r_{20})/(4y)$ . Replace  $w = (r_{01} - r_{10})/(4z)$  by  $x = (r_{10} - r_{01})/(4z)$ .

**Section 18.5, page 803.** The first paragraph after the block of pseudocode has **PrintInUset** but should be **PrintInUse**.

**Section 18.5, pages 804-805.** On page 804, the pseudocode for the second assignment operator has 4 occurrences of **rkPointer** in the body of the function but should be **rkReference**. A similar replacement must be made on page 805 in the bodies of the equality and inequality comparisons. A final replacement must be made in the text after the pseudocode, replace **rkPointer.m\_pkObject** by **rkReference.m\_pkObject**.

**Section 18.5, page 805.** The two paragraphs after the pseudocode ("Although ... due to pointer aliasing.") are taken from the 1st edition of the book. In that edition, the decrement of the counter occurred before the increment, in which case these paragraphs apply. In the 2nd edition, the smart pointer code had been rewritten to use the increment first and the decrement second, so the paragraphs are irrelevant.

**Section 18.5, page 808.** Two of the paragraphs on the page mention calls to the "copy constructor". These should be calls to the "constructor". In the 1st edition of the book, the objects were smart pointers already, so in the code for that book, in fact the copy constructors were called.

**Section 18.9.5, page 855.** The code for the LookUp function has **-m\_fRotSpeed** but should have **+m\_fRotSpeed**. Similarly, the LookDown function has **+m\_fRotSpeed** but should be **-m\_fRotSpeed**. The actual source code has the correct values.

**Section 20.5.2, page 917.** The first paragraph has an equation  $B = N \cdot T$  but should be  $B = N \times T$ .

**Section 20.7, page 926.** The equation  $V = (P-V)/|P-V|$  should be  $V = (P-E)/|P-E|$ . The displayed equation  $R = E - 2(N \cdot E)N$  should be  $R = V - 2(N \cdot V)N$ .

**Section 20.8, pages 930-931.** The pseudocode has a switch statement for which only one of the cases has a **break** statement. All cases should have **break** statements.

**Section 20.9, pages 933.** The middle line of Equation (20.3) has a term  $\sin(\varphi)$  but should have  $\sin(\theta)$ .

**Section 20.14, pages 948.** There is an equation  $E+tP$  that should be  $E+t(P-E)$ .

