几何变换示例

Ms_yam

2026-11-21

几何变换的结果可以直接当凭据输出，也可以当作其他函数（包括内联函数）的参数使用，还可以与其它宏包配合使用。

1　作为结果或参数

为方便演示，本节假设所有示例均按如下变换刷新变换函数：

```
\ExplSyntaxOn
\transform_init:n {3}
\transform_scale:nnn {2} {3} {4}
\transform_shift:nnn {1} {11} {21}
\transform_refresh:
\ExplSyntaxOff
```

1.1　直接输出

直接使用如下命令，则将各坐标分量（含后续）输出，并以逗号分隔。例如：

```
\transform_apply:nnnn {1} {2} {3} {cm}
```

将输出：3cm,17cm,33cm

1.2　普通函数的参数

几何变换的输出可以做为其他函数的参数使用，其形式有多种：

1.2.1 按逗号分隔列表处理

例如，下述代码

```
\cs_new:Nn \my_test:n
  {
    \clist_set:Nn \l_tmpa_clist {#1}
    当前参数为：
    <\clist_item:Nn \l_tmpa_clist {1}>、~
    <\clist_item:Nn \l_tmpa_clist {2}> 与
    <\clist_item:Nn \l_tmpa_clist {3}>。
  }
\cs_generate_variant:Nn \my_test:n {e}
\my_test:e { \transform_apply:nnnn {1} {2} {3} {cm} }
```

将输出：当前参数为：<3cm>、<17cm> 与 <33cm>。

注意，这个参数需要全展开（e 型或 x 型）再传递给函数；否则，效果将是：当前参数为：<3cm,17cm,33cm>、<> 与 <>。

1.2.2 指定定界符的参数

如果觉得逗号分隔列表操作太麻烦，可以通过类似以下代码的方式，直接读取各个分量。

```
\cs_new:Npn \my_test:nnn (#1,#2,#3)
  {
    $\sqrt{ (#1 cm)^2 + (#2 cm)^2 + (#3 cm)^2 } =
     \fp_eval:n {sqrt((#1)^2 + (#2)^2 + (#3)^2)} cm$;
  }
\exp_last_unbraced:Ne \my_test:nnn
  { (\transform_apply:nnnn {1} {2} {3} {}) }
```

其结果为：$\sqrt{(3cm)^2 + (17cm)^2 + (33cm)^2} = 37.24244889907214cm$;

1.3 内联函数

如果觉得上小节的全展开比较麻烦，也可以采用内联模式：

```
\transform_apply_inline:nnnn {1} {2} {3}
  {
    $\sqrt{ (#1 cm)^2 + (#2 cm)^2 + (#3 cm)^2 } =
     \fp_eval:n {sqrt((#1)^2 + (#2)^2 + (#3)^2)} cm$;
  }
```

将同样输出：$\sqrt{(3cm)^2 + (17cm)^2 + (33cm)^2} = 37.24244889907214cm$；

2 与其它宏包配合

为方便演示，本节假设所有示例均按如下变换刷新变换函数：

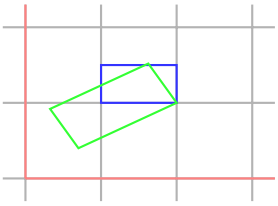
```
\ExplSyntaxOn
\transform_init:n {2}
\transform_rotated:nnn {2} {1} {30}
\transform_scale:nnnn  {2} {1} {1.5} {1.2}
\transform_refresh:
\ExplSyntaxOff
```

2.1 与 l3draw 宏包配合

几何变换输出结果可以直接用于 l3draw 图形中，以下是一个示例：

```
\draw_begin:
  % 绘制栅格
  \color_select:n { black!30 }
  \draw_set_linewidth:n { 0.8pt }
  \draw_path_grid:nnnn { 1cm } { 1cm }
      { -3mm , -3mm }  { 33mm , 23mm }
  \draw_path_use_clear:n { stroke }
  % 绘制坐标轴
  \color_select:n { red!50 }
  \draw_path_moveto:n {   0cm,   0cm }
  \draw_path_lineto:n { 3.3cm,   0cm }
  \draw_path_moveto:n {   0cm,   0cm }
  \draw_path_lineto:n {   0cm, 2.3cm }
```

```
    \draw_path_use_clear:n { stroke }
    % 绘制原始路径
    \color_select:n { blue!80 }
    \draw_path_moveto:n { 1cm,   1cm }
    \draw_path_lineto:n { 2cm,   1cm }
    \draw_path_lineto:n { 2cm, 1.5cm }
    \draw_path_lineto:n { 1cm, 1.5cm }
    \draw_path_close:
    \draw_path_use_clear:n { stroke }
    % 通过变换后的路径
    \color_select:n { green!80 }
    \draw_path_moveto:n { \transform_apply:nnn {1} { 1 } {cm} }
    \draw_path_lineto:n { \transform_apply:nnn {2} { 1 } {cm} }
    \draw_path_lineto:n { \transform_apply:nnn {2} {1.5} {cm} }
    \draw_path_lineto:n { \transform_apply:nnn {1} {1.5} {cm} }
    \draw_path_close:
    \draw_path_use_clear:n { stroke }
\draw_end:
```



2.2 与 Tikz 宏包配合

几何变换输出结果也可以直接用于 Tikz 图形中，以下是一个示例（LaTeX2e 命令形式）：

```
\begin{tikzpicture}[x=1cm,y=1cm]
  % 绘制栅格
  \draw[black!30, line width=0.8pt, step=1cm]
      (-3mm,-3mm) grid (33mm,23mm);
  % 绘制坐标轴
```

```
    \draw[red!50]
        (0,0) -- (3.3,0)
        (0,0) -- (0,2.3);
    % 绘制原始路径
    \draw[blue!80]
        (1,1) --
        (2,1) --
        (2,1.5) --
        (1,1.5) -- cycle;
    % 绘制变换后的路径
    \draw[green!80]
        ( \TransformApply [cm] {1, 1} ) --
        ( \TransformApply [cm] {2, 1} ) --
        ( \TransformApply [cm] {2, 1.5} ) --
        ( \TransformApply [cm] {1, 1.5} ) -- cycle;
\end{tikzpicture}
```

