الأبعاد وبالتالي يتم إجراء التحميل الزائد لها من أجل كل الأنواع.

إذا قمت بتمرير قيمة غير سلمية إلى تابع ذو وسيط سلمي (أي: تابع يأخذ عادة وسيطاً سلمياً مثل cos(x)) عندها سيعمل التابع على كل مركبة على حدة. مثلاً إذا كتبت:

```
float3 v = float3(0.0f, 0.0f, 0.0f);

v = cos(v);
```

عندئذ سيكون: v = (cos(x), cos(y), cos(z)).

توضح الأمثلة التالية كيف يتم استدعاء بعضاً من التوابع السابقة:

```
float x = sin(1.0f);
float y = sqrt(4.0f);

vector u = {1.0f, 2.0f, -3.0f, 0.0f};
vector v = {3.0f, -1.0f, 0.0f, 2.0f};
float s = dot(u, v);

float3 i = {1.0f, 0.0f, 0.0f};
float3 j = {0.0f, 1.0f, 0.0f};
float3 k = cross(i, j);

matrix<float, 2, 2> M = {1.0f, 2.0f, 3.0f, 4.0f};
matrix<float, 2, 2> T = transpose(M);
```

16.8: الخلاصة

- تُكتب برامج لغة HLSL في ملفات نصية ASCII ويتم ترجمتها في تطبيقاتنا باستخدام التابع D3DXCompileShaderFromFile.
- تسمح لنا الواجهة ID3DXConstantTable بتحديد قيم متحولات برنامج المظلل من داخل التطبيق. عملية الاتصال ضرورية لأن المتحولات في المظلل قد تتغير من إطار إلى إطار. مثلاً، إذا تغيرت مصفوفة العرض في التطبيق، عندها سنحتاج إلى تحديث متحول مصفوفة العرض في المظلل. حيث يمكننا فعل ذلك من خلال ID3DXConstantTable.
- يجب تعريف بنيتا دخل وخرج من أجل كل مظلل نكتبه وذلك لوصف تنسيق المعطيات التي تدخل إلى المظلل والتي تخرج منه.