
インターフェイスの街角- スナッピングの活用

増井 俊之

図形エディタと制約システム

通常、図形エディタには、図形の位置や大きさを揃えて図表や絵をきれいに描けるようにする機能が用意されています。格子状のグリッド上にだけ図形を配置したり、別の図形に揃えて図形を描いたり移動したり、あるいは選択した複数の図形の端や中心を揃えるコマンドを用意したり、図形の移動を水平/垂直のみに制限するといった方法がよく使われています。

これらの機能は、編集操作にさまざまな制限を設けて図形をきれいに配置するためのものです。“図形を等間隔で並べる”“図形と線は重ならない”などといったさらに複雑な条件もユーザーが指定できるようにした制約型図形編集システムでは、望ましい配置を保ったまま編集作業がおこなえます。

任意の制約条件を指定できるようにすると、それらの条件をつねに満たせるとはかぎらず、解として得られる配置が複数ある場合もあります。ユーザーの操作に沿って動く場合には、マウスが移動するごとに条件を満たす解を高速に計算する必要もあります。このように、制約型図形編集システムには柔軟で高速な制約解決システムが必要です。制約解決システムは図形編集以外にも応用が利くため、かなり前から各種のアルゴリズムがさかんに研究・開発され、複雑な制約条件が数多く定義されていても高速に解を求められるシステムが実用化されています。

しかし、一般ユーザーが図形を編集する場合、複雑な制約が必要になることはあまりありません。しかも条件が複雑になれば定義も面倒になるので、制約型図形編集システムは一般にはそれほど普及していないようです。ごく普通

の図形の編集に必要な比較的単純な制約条件を簡単に定義できる図形編集システムがあれば、活用する場面も多くなると思います。

スナッピング

簡単な制約のなかに、描画/編集中の図形をグリッドやほかの図形に自動的に揃える手法があります。このようなシステムでは、描く図形の制御点がグリッドやほかの図形の重力に引きつけられるように動いたり、移動中の図形がグリッドやほかの図形に引き寄せられてパチッと吸い付く(スナップする)ような動きをします。この操作を“スナッピング (snapping)”と呼びます。この機能は、一般的な図形エディタでは図形の位置や大きさを揃えるために使われます。これをさらに活用し、きれいな図を簡単に描くさまざまな手法が考案されています。

たとえば、Xerox PARC の Eric Bier らは、描画された図形を基準として新たな図形や線が描ける Snap-Dragging[1] というシステムを考案しました。このシステムでは、多角形の頂点や線の交点など、描画作業において重要と思われる点にマウスカーソルがスナップするようになっています。これによって、矩形の頂点から線を描いたり、線の交点から別の線を描き始めたりといったことが簡単にできます。一方、東京大学の本田 氏はスナッピングを拡張し、移動や拡大、回転などのモードを変更することなく図形が編集できる Integrated Manipulation という操作手法を考案しました [2]。このシステムでは、移動した図形を別の図形にスナップさせる際に、図形の回転軸(ピボット)もスナップさせることができます。したがって、周囲の図形の配置によっては、図形を移動しながら

本田さんのフル
ネームはご存じで
すか？

回転して別の図形にスナップさせるといったことが 1 回の操作で可能になります。

HyperSnapping

ユーザーのスナップ操作の意図をシステムが正しく解釈し、さらに複雑な制約条件が得られれば、それをもとにした高度な編集作業が可能になります。たとえば図形 B を移動して図形 A にスナップさせて揃えるという操作がおこなわれた場合には、A がたまたま都合のよい位置にあったからだと解釈することもできれば、A と B の関係が深い (A と B の位置に制約条件が存在する) からだとも考えられます。ユーザーが後者を意図したのであれば、A の位置変更と連動して B の位置も変わるほうが都合がよいでしょう。この例では、B を移動して A にスナップするという 1 つの作業で B と A の制約関係が定義できたことになります。また、似たようなスナップ操作が繰り返し実行された場合、それらの操作に共通する処理を抽出して自動的にマクロとして定義することも可能です。

このように、ユーザーが意図をはっきり認識しながらスナップ操作をおこなえば、編集作業とともにみずからの意図をシステムに指令し、その後の編集作業に活用することができま

今回は、上記のアイデアに沿って作成した「HyperSnapping」システムを紹介します。このシステムの特徴は、ユーザーの意図をシステムに伝えるためにスナップ操作を活用する点にあります。以下、例を挙げながら HyperSnapping の機能について説明します。

通常のドラッグ操作

図形上でマウスをクリックして動かし、図形を直接ドラッグすることができます。図 1 は、矩形をドラッグして任意の位置に移動しているところです。通常は、マウスの動きに従って矩形が移動します。

グリッドの制御

一定の距離を越えてドラッグをおこなうと背景にグリッドが出現し、図形はグリッドにスナップしながら移動します。グリッドへのスナップ点は小さな矩形で表現されています。以下では、スナップ点を「アンカー」、アンカーをもつ図形を「アンカー図形」と呼びます。アンカー

図 1 矩形をドラッグ

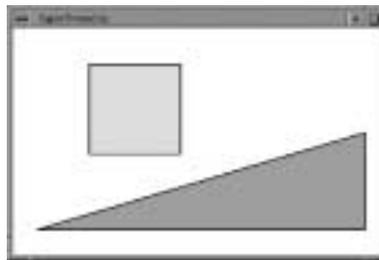


図 2 小さなグリッドにスナップ

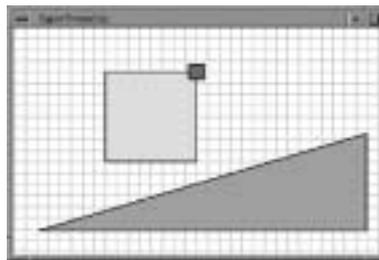
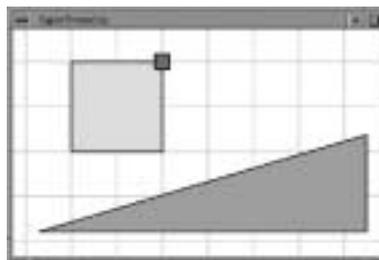


図 3 大きなグリッドにスナップ



は、あとで解説するさまざまな操作で基準点として働きます。スナップの候補点が複数あるときは、ドラッグの向きによりアンカーが選択されます。

さらにドラッグを続けると、グリッドのサイズは段階的に大きくなっていきます。つまり、ドラッグの距離に応じてスナップの単位が大きくなったり、小さくなったりするわけです (図 2~3)。

この機能により、図形を遠くに移動する場合はおおらかな位置に揃え、近くに移動するときは細かく制御することが可能になります。

グリッドの基準点の制御

グリッドは、それ以前の操作においてアンカーをもっていた図形を基準に生成されます。アンカー図形が三角形である場合は、図 4 のように三角形の大きさにもとづいてグリッドが生成されます。これによって、三角形の縦横にあ

図 4 別の図形を基準としたグリッド使用

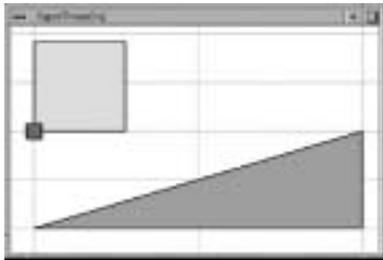
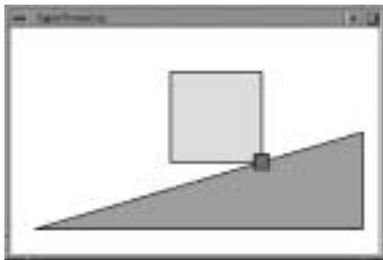
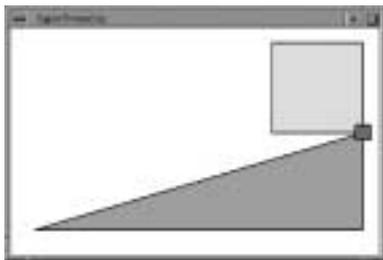


図 5 頂点や辺へのスナップング



わせた位置に別の図形を簡単に配置することができます。

アンカーによる回転/拡大中心の指定

ドラッグ中の図形は、グリッドだけでなくほかの図形の頂点や辺にもスナップします(図 5)

グリッドへのスナップングと同様、図形のドラッグ距離が小さい場合はスナップングは起こりません。スナップングが起きるのは、ある程度以上、図形を移動したときだけです。図形 A を図形 B からほんのすこし離れた位置に移動させたいときは、まず A を移動して B にスナップさせ、その後に A をわずかに移動させます。

図形の頂点付近をクリックして移動すれば、アンカーを軸として図形の回転/拡大をおこなうことができます(図 6) マウスをクリックせずに図形上で移動すると、クリックした場合の操作が影として表示されます。

図形を回転/拡大したときも、移動と同様にほかの図形やグリッドへのスナップングがおこなわれるため、図形を

図 6 アンカーを中心とした拡大/回転

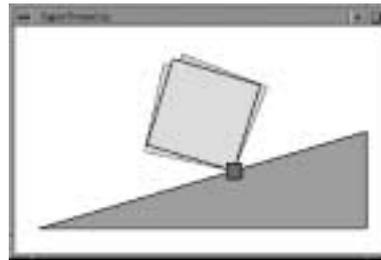
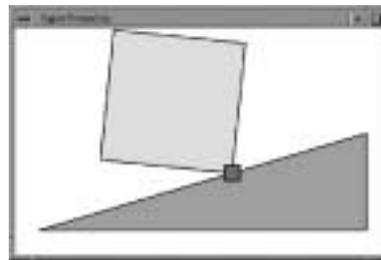


図 7 回転して斜辺に揃える

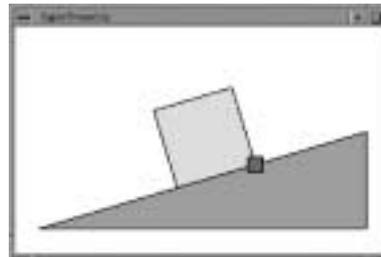
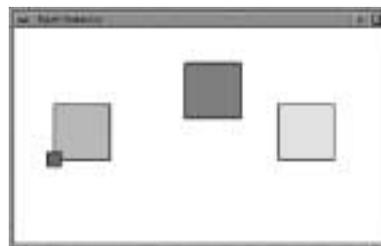


図 8 3 個の矩形の初期状態



ほかの図形の辺に揃えることができます(図 7)

グループ化と制約の自動生成

アンカー図形に別の図形をスナップさせると、両者のあいだに制約関係が生成され、複数の図形を 1 つのグループとして操作できます。

図 8 の状態で、中央の矩形を移動して左の矩形にスナップさせると、スナップ点(サブアンカー)が白い小さな矩形が表示されます(図 9)

図 9 中央の矩形を左の矩形にスナップ

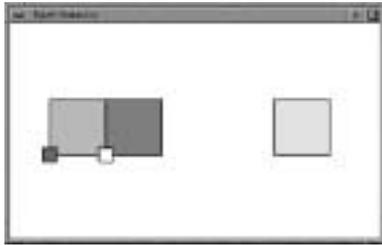


図 10 右の矩形を中央の矩形にスナップ

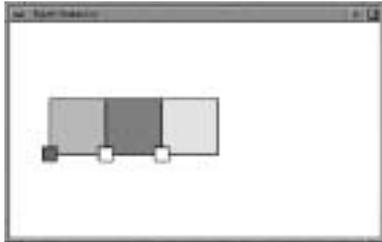


図 11 右の矩形をドラッグ

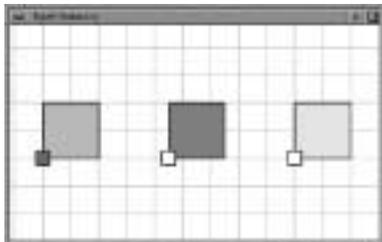
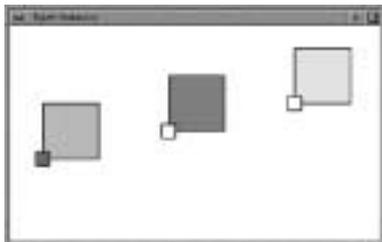


図 10 は、右側の矩形も移動してスナップさせたところ
です。これら 3 個の矩形のあいだには制約関係が自動的に
定義されます。たとえば、右または中央の矩形を移動する
と、ほかの 2 つの矩形もそれに従って移動します(図 11)

アンカー図形を移動した場合は、図 12 のようにサブア
ンカー図形も同時に移動します。

サブアンカー図形を回転させると、アンカーを中心とし
て全体が回転します(図 13)、アンカー図形を拡大/回転
すると、すべての図形が同じように拡大/回転されます。

図形以外の場所をクリックするとアンカー/サブアンカ

図 12 アンカー図形をドラッグ

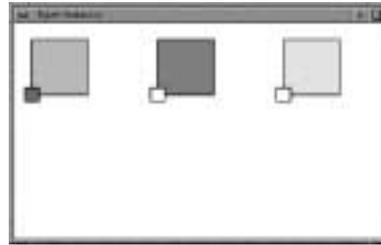


図 13 サブアンカー図形の回転

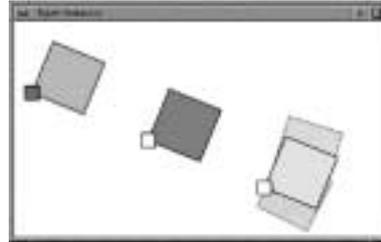


図 14 サブアンカー図形のコピー

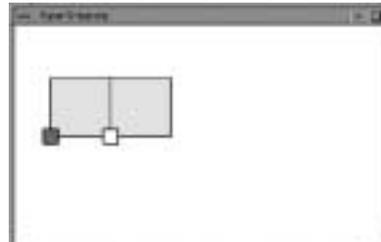
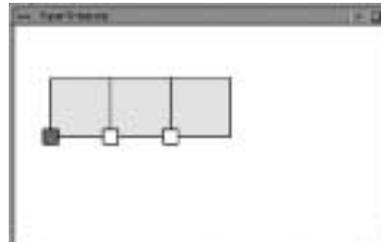


図 15 サブアンカー図形のコピー(繰り返し)



—はすべて消去され、制約関係も解消されます。

繰り返し操作

図 14 は、1 個の矩形をコピーしてからペースト操作に
より新たな矩形を生成し、もとの矩形にスナップさせた
ところです。同じ操作をもう 1 回実行すると図 15 の状態
になります。

このように、同じ操作を 2 回以上繰り返した場合は、操
作の履歴から繰り返し部分を抽出して次の操作を予測/実

図 16 繰り返し操作指令による再実行

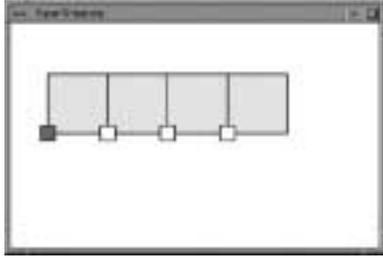


図 17 サブアンカー図形を移動

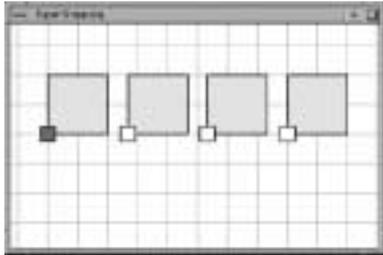
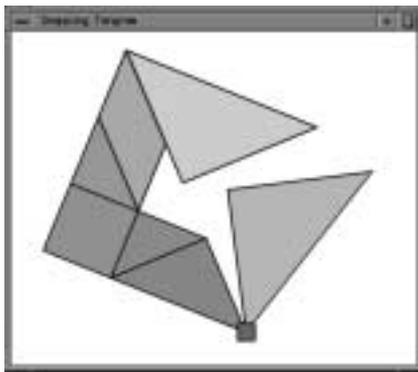


図 18 タングラム



行することができます [3]。この状態で“繰り返し実行”を指示すると、操作履歴中の繰り返し操作が自動的に抽出され、図 16 のように再実行されます。

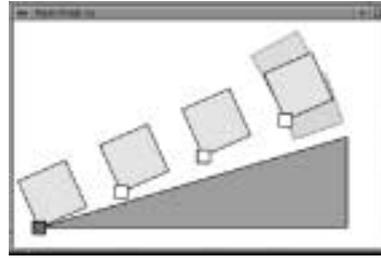
予測によって生成された図形のあいだには制約関係が成立しているため、サブアンカーの矩形を移動するとほかの矩形も移動します(図 17)

その他の描画例

スナッピングを利用すると、図 18 の“タングラム”のような傾いた図も簡単に描けます。

図 19 は、三角形の斜辺に 4 個の矩形を等間隔で並べようとしているところです。制約やスナッピングなどの機能のない図形エディタでは、こういった図形の描画は簡単

図 19 三角形の斜辺に矩形を並べる



ではありません。これに対し、HyperSnapping ではメニューなどをまったく使わずにこのような図形が描けます。

おわりに

今回は、スナッピング操作と制約解決システムを組み合わせ、きれいな図を簡単に描く方法を紹介しました。

この連載でも、以前にキーボードのない PDA などテキストを簡単に入力できる「POBox」システムや、それを図形や絵の作成に応用したシステムを紹介しました。今回の方法を発展させていけば、図形やテキストをきれいに編集/整形する操作が容易におこなえます。今回のシステムは、まだまだ完成にはほど遠いのですが、ペンやマウスさえあれば誰でもテキストや図表が簡単に描けるエディタに育てていきたいと考えています。

(ますい・としゆき ソニー CSL)

[参考文献]

- [1] Eric Allan Bier, “Snap-dragging”, *Computer Graphics*, Vol.20, No.4, pp.233-240, August 1986
- [2] Masaaki Honda, Takeo Igarashi, Hidehiko Tanaka and Shuichi Sakai, “Integrated Manipulation: Context-aware Manipulation of 2D Diagrams”, *Proceedings of the ACM Symposium on User Interface Software and Technology (UIST'99)*, ACM Press, November 1999
- [3] Toshiyuki Masui and Ken Nakayama, “Repeat and predict – two keys to efficient text editing”, *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI'94)*, pp.118-123, Addison-Wesley, April 1994

可能であれば移動環境などを加筆していただけないか