

Studio di fattibilità di un modello feature-based per oggetti free form

C.E. Catalano, F. Giannini, M. Meirana

Rapporto Tecnico I.M.A.- N. 19/00

Indice

1. Presentazione del problema.....	2
2. Manipolazione della superficie	3
3. Feature strutturali	4
3.1 Modifica ed interazione tra feature strutturali	9
3.2 Esempio	12
4. Feature di dettaglio.....	15
4.1 Modifica ed interazione relative a feature di dettaglio	17
4.2 Esempio	20
5. Entità virtuali	21
6. Conclusioni	24
Bibliografia.....	25

1. Presentazione del problema

Nel campo della progettazione estetica, la **rappresentazione CAS/CAD** (Computer Aided Styling e Computer Aided Design) dei prodotti avviene sostanzialmente attraverso una modellazione a superfici, al contrario di quanto accade nell'ambito meccanico, dove si utilizzano gli strumenti della modellazione solida.

Un modello geometrico comunemente utilizzato è il modello B-rep il quale, oltre a contenere informazioni di tipo geometrico, memorizza anche la topologia dell'oggetto: contiene, quindi, anche i legami di adiacenza tra entità (ad esempio, facce e relativi spigoli). Un modello di questo tipo descrive compiutamente la geometria, ma non dà alcuna informazione di tipo semantico; in altre parole, vi sono tutte le informazioni necessarie, ma non sono organizzate per significato.

Un approccio già ampiamente sperimentato in campo meccanico è quello **feature-based**, in cui gli elementi geometrici vengono raggruppati in un'unica entità secondo un comune significato. In questo modo, l'utente riesce a gestire il modello con più facilità in quanto lavora con elementi geometrici di più alto livello.

Nel campo della progettazione estetica, l'inserimento della semantica all'interno del modello è sicuramente un passo in avanti per due motivi principali:

1. l'estetica di un prodotto non può essere direttamente associata alla geometria, se prima non si è formalizzata una descrizione intermedia;
2. gli utenti, in questo caso, sono stilisti che non sono né esperti né interessati ad un modello geometrico strettamente matematico.

Stante queste considerazioni, l'**idea** è quella di arricchire il modello geometrico estendendolo ad un modello feature-based strutturato che permetta di rappresentare anche la semantica. Più precisamente, si vogliono utilizzare due modelli:

- B-rep, che mantiene la geometria e la topologia;
- feature-based, come sovrastruttura del modello B-rep che mantiene informazioni relative alle feature estetiche.

Le varie feature, strutturali e di dettaglio [1], andranno relazionate, in modo tale da evitare l'esigenza di avere a disposizione una storia completa nel modello geometrico.

Per completezza, abbiamo preso in considerazione anche la possibilità di costruire un modello che mantiene la storia di costruzione di tutti gli elementi che lo compongono. Si pone, allora, la scelta tra un modello history-based, molto potente, ma anche piuttosto oneroso, e un *modello doppio*, che permette di trattare separatamente l'aspetto geometrico e quello semantico mediante una strutturazione della geometria stessa.

Ciò che seguirà è, in pratica, uno **studio di fattibilità**: per decidere la soluzione *migliore*, analizzeremo quali requisiti dovrebbe possedere un modello strutturato che faciliti la creazione e la modifica di oggetti free-form. Entrambi gli aspetti – creazione e modifica – possono essere avvantaggiati dalla definizione tramite pochi parametri significativi di entità ad alto livello. Inoltre, nel primo caso, si aggiunge l'opportunità del riutilizzo di modelli preesistenti di oggetti simili; nel secondo, si ottimizza la fase di variazione della forma grazie ad operatori agenti su un numero ridotto di parametri.

Concretamente il modello feature-based dovrà essere rappresentato al calcolatore tramite una **struttura dati** opportuna: l'indecisione è tra la lista ed il grafo. Una lista permette di memorizzare nei campi le diverse feature e garantisce un ritrovamento sequenziale delle informazioni. D'altro canto, non permette di istituire dei legami privilegiati tra gli elementi della lista. Questa esigenza è, invece, supportata dai grafi in quanto permettono di collegare con archi quei nodi che sono in relazione tra loro. In questo caso, i nodi descriveranno le feature, mentre gli archi le relazioni. Ovviamente, la struttura del grafo ha costi computazionali (in spazio ed in tempo) maggiori rispetto alla lista e quindi si deve verificare se il numero e l'importanza delle relazioni trovate giustificano tale scelta. Il rischio è quello di costruire grafi sparsi (a molte componenti connesse e componenti connesse piccole).

2. Manipolazione della superficie

L'efficienza di una rappresentazione è necessaria non tanto nella fase di creazione del modello quanto durante quella di manipolazione. Infatti, le modifiche alle superfici, il conseguente aggiornamento della rappresentazione, il riutilizzo di modelli già esistenti sono una parte fondamentale nel processo di design (non solo estetico, ma anche ingegneristico). E' per una migliore ottimizzazione di questo tipo di processi che le feature devono essere scelte opportunamente: una volta definiti i parametri delle feature e le loro relazioni con le altre entità, l'operazione di modifica dovrebbe risultare il più facile e completa possibile.

Quando si parla di **feature strutturali**, si deve sempre tenere presente che si sta lavorando sulla superficie a livello globale, proprio per definizione stessa di feature strutturale. Di conseguenza, quando si modifica una feature di questo tipo, si sa che si sta modificando la superficie globalmente. Le interazioni tra feature strutturali comportano modifiche globali, contrapposte alle variazioni che si effettuano creando **feature di dettaglio** che sono, invece, locali. Infatti, quando si inserisce in un modello una feature di dettaglio, la parte interessata dalla modifica risulta essere circoscritta e l'impatto visivo globale dell'oggetto non cambia. Vi sono, poi, modifiche locali che non sono legate a feature e che sono di tipo sostanzialmente diverso dalle precedenti.

Per esempio, lo spostamento di uno o più punti di controllo di una superficie è un'operazione di basso livello e non può essere considerata come feature.

L'analisi successiva verte quindi sulle proprietà delle feature estetiche evidenziate, e sulla relazione tra il modello geometrico e la sovrastruttura. E' ovvio infatti che i due modelli debbano comunicare per poter gestire lo scambio di informazioni a diversi livelli.

3. Feature strutturali

Le feature strutturali sono curve portanti della rappresentazione e sono deducibili dagli schizzi dello stilista: sono quelle linee che egli stesso ha evidenziato in modo particolare nel disegno manuale. Tra queste, alcune sono curve di struttura, come ad esempio le sezioni ed i profili (o contorni) dell'oggetto, mentre altre sono linee che danno l'impatto estetico al prodotto. In corrispondenza del ruolo di tali curve, abbiamo distinto le feature strutturali in due classi: i **contorni** e le **linee di carattere**. I primi, che possono essere suddivisi in sezioni e profili, sono ottenuti per intersezione delle superfici con un piano; in particolare i profili sono delle particolari sezioni che descrivono la forma dell'involucro che racchiude l'oggetto. Le linee di carattere, invece, non sono tipicamente curve reali, ma solo percepite (come, ad esempio, le linee di luce e di ombra). La caratteristica che accomuna tutti i tipi di feature strutturali è che, per la loro importanza, non possono scomparire, ma solo essere modificate.

Matematicamente parlando, i contorni sono definiti appunto come intersezione, mentre le linee di carattere sono individuate da condizioni sulla curvatura (di cui qui non ci occupiamo) e sulla regolarità delle superfici costruite a partire da esse [2,3]. Infatti, una repentina variazione di curvatura (continuità geometrica G^1), una mancata condizione di tangenza (continuità G^0) oppure un taglio (continuità G^{-1}) producono degli effetti visivi che caratterizzano in diversi modi la superficie. Si deve specificare, comunque, che la regolarità deve essere valutata a meno dei raccordi utilizzati per rendere più liscia la superficie. Per esempio, considerando uno spigolo smussato, è lo spigolo vivo che identifica la linea di carattere, mentre lo smussamento viene trattato separatamente come feature di raccordo.

Il compito di specificare quali sono le entità da considerare come feature strutturali e qual è il loro utilizzo (e cioè se sono sezioni, profili o linee di carattere) è lasciato all'utente. In particolare, non tutti i contorni devono essere considerati come feature, ma solo quelli che si ritengono più significativi; gli altri verranno trattati direttamente nel modello geometrico di base.

Riepilogando brevemente, le feature strutturali possono essere:

- contorni utilizzati per creare le superfici che costituiscono l'oggetto;
- curve che delimitano superfici adiacenti con continuità geometrica G^0 , in tal caso sono dette curve di bordo;
- curve utilizzate per creare superfici con particolari curvature;
- curve utilizzate per creare tagli.

Ogni feature strutturale viene memorizzata come nodo della struttura dati e, per essere definita, necessita di parametri opportuni. Essendo un'entità ad alto livello, pochi elementi significativi permettono di rappresentarla interamente. Gli elementi (**parametri**) da considerare nella rappresentazione di una feature strutturale sono:

- il tipo di feature strutturale: contorno (sezione o profilo) oppure linea di carattere;
- link alle curve che compongono la feature strutturale: non si deve dimenticare, infatti, che le curve della rappresentazione vengono descritte tramite spline, che sono proprio curve definite a tratti;
- link alle superfici comprese nell'area di influenza della feature strutturale;
- condizioni di continuità tra le curve che costituiscono la feature.

Per le feature strutturali che sono di bordo, che equivale a dire che fanno parte della frontiera di una o più superfici del modello, sembra superfluo considerare anche un puntatore alle superfici adiacenti alla feature strutturale. Infatti tramite il link alle curve si accede alla rappresentazione B-rep che contiene tutte le informazioni di adiacenza tra le varie entità del modello. Il link alle superfici per feature di bordo serve però, come vedremo in seguito, se l'utente stabilisce che l'area influenzata dalla presenza delle feature è più vasta rispetto a quella costituita dalle sole superfici adiacenti. Per le feature strutturali che non sono di bordo e che quindi sono entità virtuali è necessario memorizzare anche il link alle superfici ad esse adiacenti perché l'informazione non è ricavabile direttamente dal modello B-rep.

Quindi per rappresentare le relazioni tra le feature strutturali e le superfici del modello B-rep è sufficiente memorizzare:

- per le feature di bordo, le superfici influenzate dalla presenza della feature strutturale ma che non sono adiacenti alla feature stessa (a quelle adiacenti si può accedere dal modello B-rep);
- per le feature che non sono di bordo, tutte le superfici (adiacenti e non) influenzate dalla presenza della feature.

Inoltre ogni tipologia di feature strutturale avrà propri parametri aggiuntivi che la caratterizzano. In particolare, la differenza sostanziale nella definizione di una sezione o profilo rispetto ad una linea di carattere sta nella specifica del piano di intersezione in un caso e della regolarità delle superfici sottogiacenti nell'altro. Ovviamente si possono considerare altri attributi anche di tipo non geometrico che qui non trattiamo. Ad esempio, per confrontare modelli di automobili potrebbe essere necessaria l'aggiunta di nomi identificativi per mettere in evidenza il ruolo delle singole feature (ad esempio per specificare che una feature è parte del tetto piuttosto che del cofano).

Una feature strutturale si può rappresentare informalmente, utilizzando la simbologia della progettazione orientata agli oggetti, nel seguente modo (Fig. 1):

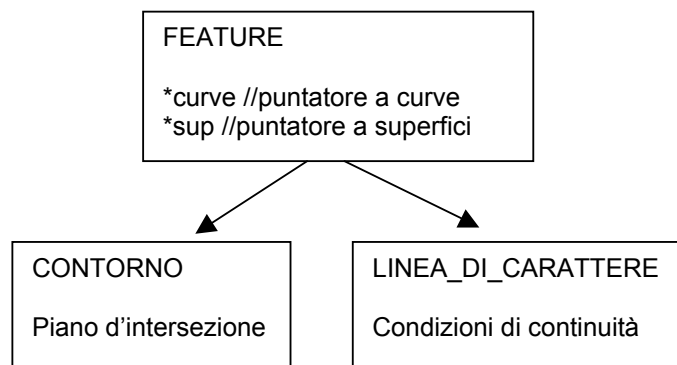


Fig. 1: Rappresentazione di feature strutturali

dove:

- ogni nodo rappresenta una classe e gli archi stanno ad indicare che le classi CONTORNO e LINEA_DI_CARATTERE sono derivate dalla classe base FEATURE. Ovviamente i parametri e i metodi della classe base vengono ereditati dalle classi derivate.
- Il puntatore a curve `*curve` è un puntatore alla lista di curve che compongono la feature. E' quindi, per le curve reali, un puntatore al modello B-rep; mentre per quelle virtuali è un puntatore ad una lista aggiuntiva di elementi contenenti le informazioni sulla geometria delle curve. Ad ogni curva si aggiunge anche un attributo che indica quali vincoli esistono relativamente al grado di continuità geometrica con cui si raccordano le curve costituenti la feature.
- Se la feature è di bordo, il puntatore a superfici `*sup` è un puntatore alla lista di superfici non adiacenti alla feature ma che sono influenzate dalla sua presenza. Se invece la feature è costituita da curve virtuali tale puntatore punta alle superfici che sono state create utilizzando la

feature stessa come parametro di costruzione (per maggiori dettagli vedere capitolo 5) e a quelle eventualmente aggiunte dall'utente.

In modo analogo si possono definire ulteriori classi derivate, e in particolare si deve spendere qualche parola in più sulla definizione di quelle linee di carattere che implicano una regolarità G^{-1} della superficie. Tali feature vengono denominate **gap** e possono essere sia strutturali sia di dettaglio. Un gap può essere ottenuto posizionando componenti diverse ad una certa distanza oppure tagliando *strisce* di superfici preesistenti. In Fig. 2 sono evidenziati alcuni gap.



Fig. 2: Esempi di gap

Durante la fase di CAS, i gap sono individuati come curve di separazione netta tra superfici, mentre è solo nella fase CAD che vengono fornite tutte le specifiche complete del prodotto e vengono pertanto aggiunte le *rifiniture*. Quindi, nel CAS si deve identificare solo la linea portante del taglio; a livello di definizione di tale feature, si deve specificare che quello che si sta definendo è un gap, qual è la curva che lo definisce (curva a cui si deve far puntare *curve) e le relative superfici (collegate attraverso il puntatore *sup). Quindi in questa fase il gap non è ancora tale ma è solo rappresentato tramite una linea di carattere (la curva portante). Il gap vero e proprio si costruisce solo nella fase CAD in cui si devono aggiungere le specifiche sul tipo di taglio: tra i parametri, le curve che delimitano la zona troncata e che andranno poi raccordate alla superficie in un secondo momento. In particolare, a livello della definizione della struttura, si possono distinguere essenzialmente due tipi di gap:

- il gap per cui le linee di taglio sono semplicemente una copia della curva introdotta nella fase CAS (Fig. 3.a);
- il gap per cui le curve di taglio effettivo si discostano dalla linea di carattere (Fig. 3.b).

Nel primo caso, i parametri da inserire sono la linea di carattere e il discostamento tra i tagli effettivi; nel secondo caso, si dovranno specificare le due curve di taglio e il rapporto (ad esempio, la distanza) tra tali linee e la linea di carattere.

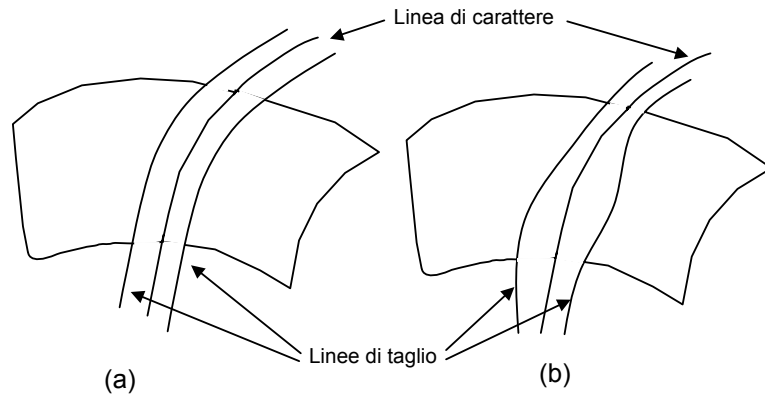


Fig. 3: Esempi di curve di taglio

Si può quindi considerare la classe rappresentante il gap (Fig. 4) come classe derivata dalla classe LINEA_DI_CARATTERE con la specifica dei relativi parametri aggiuntivi che sono utili per mantenere la feature strutturale. Per il gap è necessario memorizzare oltre alla linea di carattere anche le curve di supporto in quanto, dopo il trimming, la feature diventa una linea virtuale non più appartenente al modello geometrico, mentre le curve di supporto sono linee effettivamente giacenti sulla superficie e quindi mantengono il legame di adiacenza con le superfici coinvolte (vedere, come esempio, rispettivamente le linee di carattere e quelle di taglio di Fig. 3).

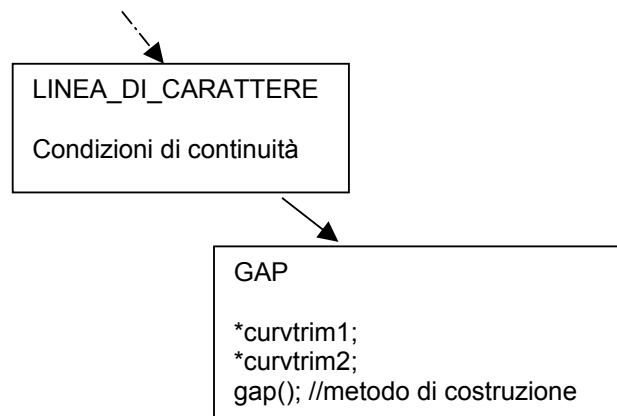


Fig. 4: Rappresentazione del gap

I parametri presenti nel nodo GAP sono i puntatori alle curve di supporto (*curvtrim1 e curvtrim2) e alla linea portante (*curve) mentre la funzione gap() è il metodo di costruzione del gap. Il metodo di costruzione crea o riceve come dato di input le curve di supporto e, a partire da esse, costruisce il gap vero e proprio. Solitamente la creazione del gap fa sparire la porzione di superficie su cui giace la curva che lo definisce, la quale dovrà essere inserita tra le curve virtuali e alla quale farà riferimento il puntatore *curve.

Un problema da affrontare è come trattare le entità virtuali corrispondenti ai parametri geometrici non appartenenti al modello B-rep dell'oggetto.

Dopo aver definito da quali elementi sono costituiti i nodi rappresentanti le feature strutturali, la cosa più importante da stabilire è come far interagire tali nodi con il modello B-rep e cioè come agiscono effettivamente i puntatori *curve e *sup.

Inoltre si deve specificare dove mettere l'informazione riguardante le condizioni di continuità geometrica e come ci si deve comportare se la linea di carattere è solo una curva percepita e non realmente appartenente al modello geometrico (entità virtuale).

3.1 Modifica ed interazione tra feature strutturali

L'operazione di modifica di una feature strutturale comporta due tipi di problemi da risolvere:

- il rapporto tra la curva modificata e le superfici sottostanti;
- l'interazione tra più feature.

Il primo punto - come modificare le superfici a partire dalla nuova curva costruita - viene già risolto a livello del modello geometrico, nel senso che è un problema superato dai sistemi attuali. Un modellatore a superfici è in grado di supportare l'aggiornamento delle superfici, una volta effettuata una manipolazione da parte dell'utente; il problema è sostanzialmente il tipo di modifiche che il sistema riesce a gestire. Quello che si può fare in presenza di una feature strutturale è quello di imporre una certa tipologia di propagazione della modifica e cioè specificare come si deve comportare il modellatore geometrico quando ne viene richiesto l'aggiornamento a seguito della modifica di una feature strutturale. Questo avviene normalmente nella modellazione per feature in cui ad ogni feature si associa un operatore che in caso di modifica permette il diretto aggiornamento della feature stessa e delle relative superfici. Non si deve trascurare che la definizione di operatori di modifica di oggetti free-form non è semplice in quanto esistono più modi per costruire e modificare superfici a partire da una feature strutturale. Ad esempio, si può utilizzare l'operatore di sweep oppure operatori, come lo skin, che permettono di interpolare più curve. Una

volta definito l'operatore di modifica, esso può essere aggiunto ai metodi della classe che rappresenta la feature a cui deve essere associato.

Rimane da trattare il secondo punto, che coinvolge, invece, la sovrastruttura da noi introdotta. Il fulcro da analizzare riguarda la determinazione del legame tra le varie feature strutturali del modello.

Prima di procedere nella spiegazione, conviene specificare meglio il discorso dell'**area di influenza** che è la regione al di fuori della quale la superficie è non modificabile. Se tale area comprende solo parziali parti dell'oggetto ciò comporta processi di approssimazione ed ottimizzazione poiché non è detto che tutti i vincoli (come la continuità sulla frontiera) possano essere contemporaneamente ed esattamente soddisfatti. Sembra comunque sensato dare per default una definizione di area di influenza per ogni singola feature strutturale:

data una feature strutturale si considera come sua area di influenza
quella comprendente tutte e sole le superfici ad essa adiacenti.

Con tale definizione di area di influenza, si possono stabilire già a priori vincoli tra superfici e feature strutturali senza imporli run time. Infatti scegliendo come area di influenza quella costituita dalla superfici adiacenti non si creano problemi di gestione della struttura in quanto l'informazione di quali sono le superfici coinvolte è ottenuta, per le feature costituite da entità virtuali, attraverso il puntatore *sup che è di collegamento tra la sovrastruttura e il modello B-rep e, per le feature che sono di bordo, direttamente dal modello B-rep.

Trattandosi di feature strutturali e quindi di entità che influenzano strutturalmente l'oggetto si può lasciare all'utente la possibilità di estendere l'area di influenza di default di una feature strutturale aggiungendo altre superfici direttamente adiacenti.

Tali superfici andranno temporaneamente aggiunte all'area di influenza attraverso il puntatore *sup e verranno tolte immediatamente dopo l'esecuzione della modifica.

Passando ora al punto centrale, se si modifica una feature strutturale e le altre feature non intersecano la sua area di influenza è inutile definire alcun **legame**. L'aggiornamento rimane locale: la feature è aggiornata automaticamente dall'utente che la sta modificando e così le curve del modello B-rep a cui è collegata e le superfici, come succede normalmente, sono aggiornate solo nel modello B-rep. Il problema si pone nel caso contrario e cioè quando l'area di influenza della feature modificata interseca quella di altre feature. E' inutile creare un legame tra tutte le feature strutturali, mentre sembra più appropriato relazionare solo quelle che hanno aree di influenza di default intersecanti. Infatti in questo caso è molto probabile che la modifica di una coinvolga anche l'altra.

Inoltre ci sono casi in cui due o più feature strutturali interagiscono sicuramente; questo accade quando esse si intersecano. In queste circostanze, quando si modifica una delle feature, l'utente può decidere di voler mantenere il vincolo di passaggio per il punto di intersezione. Operativamente questo può essere ottenuto aggiungendo un attributo, che specifichi il vincolo di passaggio, al link che relaziona le feature interagenti.

L'*idea* su come gestire la propagazione della modifica di una feature strutturale all'intero modello, è quella di dare priorità alla curva che si è scelto di modificare. Dalla nuova curva, si ricostruiscono le superfici e si ricalcolano le feature eventualmente presenti nell'area di influenza. L'utente può scegliere tra due alternative:

- 1) si effettua la nuova valutazione e poi si controlla se le altre feature continuano a mantenere le caratteristiche iniziali. Il modello deve comunque aggiornarsi andando a ricalcolare le nuove feature strutturali e verificando che soddisfino le esigenze dell'utente. Un problema è come stabilire che proprietà conservare e come valutarle ma tale scelta si può lasciare all'utente a cui possono essere eventualmente proposte le nuove feature strutturali. Per aggiornare i nodi, per ogni feature si modificano quelli relativi agli edge a cui punta e in tal modo cambia la sua geometria; si ha solo un aggiornamento di puntatori.
- 2) si effettua una modellazione vincolata, ovvero si considerano le feature strutturali interagenti con quella che si sta modificando come dei vincoli sulla forma, decidendo, prima di effettuare la propagazione della modifica al modello, quali proprietà devono continuare a soddisfare le altre feature presenti nell'area di influenza della feature modificata. Il vincolo più palese è quello relativo alla forma/geometria della feature strutturale; ad esempio nel caso in cui la feature è virtuale si possono imporre particolari condizioni sulla curvatura della superfici ad essa relative. Operativamente, per mantenere la geometria di una feature strutturale, si possono vincolare gli edge che la compongono ad essere non modificabili e tale vincolo può essere inserito nel nodo rappresentante la feature stessa.

Si è giunti a questa proposta, basata su un criterio di priorità, dopo aver analizzato la situazione in cui le feature strutturali interagivano direttamente a livello della prima modifica. Creando un legame effettivo tra queste entità senza dare criteri di priorità run-time, si predispone un rimando continuo tra le feature che continuano ad essere ricorsivamente modificate e questo perché le feature sono mantenute allo stesso livello d'importanza. Più sensato, invece, riferirsi sempre al modo di operare del progettista. Avendo deciso di modificare un'entità ben determinata, l'utente ha fissato quella curva come prioritaria e quindi sarà essa a guidare la modifica globale. Le altre feature strutturali non vanno certo trascurate, ma dovranno *adattarsi* alla richiesta dell'utente.

Naturalmente alcune caratteristiche dovranno rimanere tali, come ad esempio il grado di continuità tra gli elementi, e quindi una modellazione vincolata deve essere sempre possibile. Anche in questo caso, si otterrà un'approssimazione dei desiderata iniziali. Un'alternativa è quella di utilizzare algoritmi che ottimizzano la superficie con lievi modifiche per entrambe le curve piuttosto che tenere fissa una ed ottimizzare solo l'altra. La scelta dipende essenzialmente dal modellatore a disposizione.

Essendo la priorità stabilita run-time, non si ritiene necessario appesantire la struttura aggiungendo relazioni che mantengono le condizioni di priorità tra feature strutturali.

L'approccio proposto permette di gestire le relazioni tra le feature strutturali che si trovano nella stessa area di influenza, considerando con priorità massima la feature che si è scelto di modificare. Rimane il problema di **riaggiornare** le altre feature strutturali.

Se si modifica una feature di bordo per una o più patch, è ragionevole pensare che si possa trovare una parametrizzazione della nuova superficie in cui quella curva continui a rimanere di bordo. Ciò è utile perché le nuove curve e superfici continuano ad essere contenute nel modello geometrico e aggiornate in quello feature-based. Se appartengono al B-rep, le informazioni ad esse relative sono incluse nel modello geometrico e quindi la gestione sarà meno problematica. Nel caso contrario di feature non di bordo, si ha che le curve che le costituiscono non appartengono al modello B-rep e quindi si ricade nel caso di entità virtuali (vedere Cap. 5).

3.2 Esempio

Presentiamo un esempio che illustri come poter rappresentare un oggetto tramite una rappresentazione feature-based. Consideriamo Fig. 5 in cui è mostrata la parte frontale di una moto mettendo in evidenza alcune feature strutturali e la loro rappresentazione nel modello proposto. Metà del motociclo è stata appositamente oscurata perché irrilevante per la nostra discussione in quanto è l'esatto simmetrico della parte non oscurata. Pertanto d'ora in poi considereremo, di questo esempio, la rappresentazione della sola parte anteriore destra del motociclo.

In Fig. 5 sono state rappresentate parzialmente anche le relazioni tra alcune delle entità del modello che ovviamente è molto più complesso. In particolare:

- F_i rappresenta l' i -esima feature strutturale
- e_{ij} rappresenta il j -esimo edge dell' i -esima feature strutturale
- S_h rappresenta l' h -esima superficie

- le frecce tratteggiate mostrano il tragitto del puntatore *curve
- le frecce non tratteggiate mostrano (in modo molto semplificato) le relazioni stabilite tramite il modello B-rep

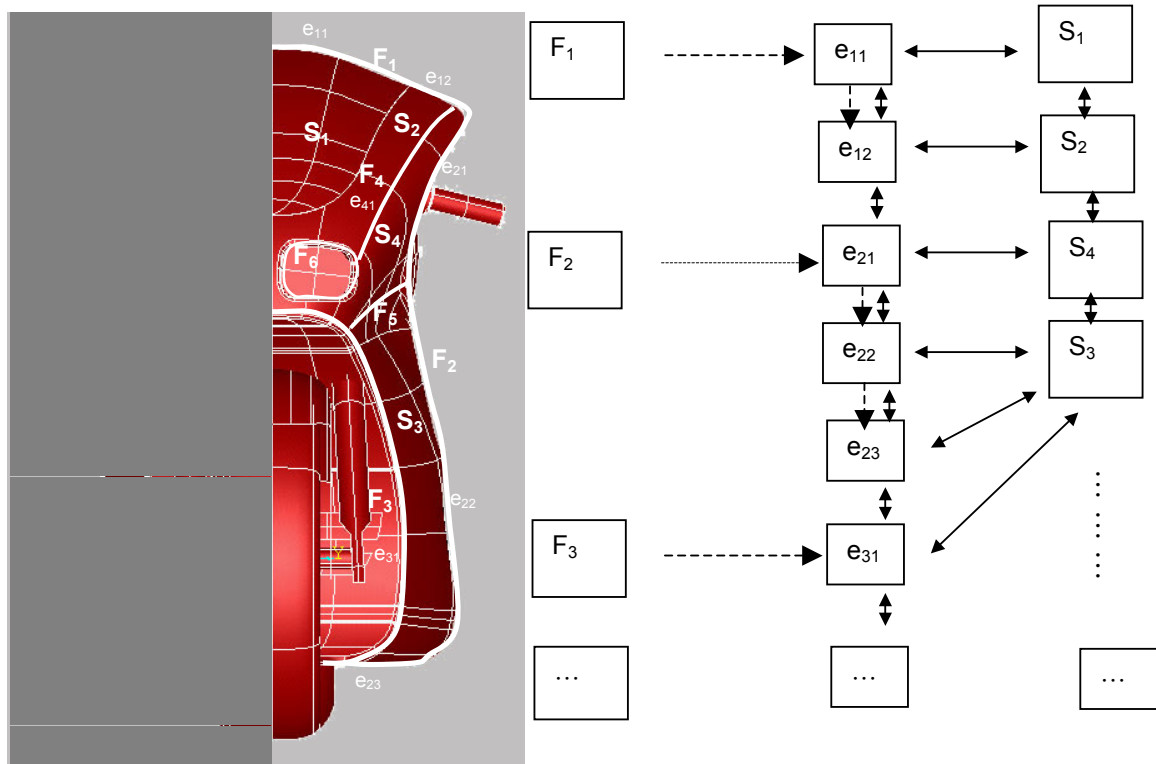


Fig. 5: Esempio di rappresentazione di feature strutturali

Non abbiamo indicato il percorso seguito dal puntatore *sup supponendo, per ora, di considerare l'area di influenza di default di ogni feature strutturale e di ricavare le relazioni con le superfici direttamente tramite il modello B-rep.

Supponiamo di modificare la feature strutturale F_1 che, essendo parte della frontiera delle superfici S_1 ed S_2 , è una feature di bordo. Le superfici S_1 ed S_2 si dovranno aggiornare per andare a soddisfare le nuove condizioni imposte dalla forma della nuova F_1 . La frontiera di S_2 è però delimitata dalla feature F_4 che si trova pertanto nell'area di influenza di F_1 .

Per come è stata effettuata la modifica, la feature F_1 ha priorità massima rispetto ad F_4 e quindi l'aggiornamento della superficie S_2 deve tener conto dei vincoli imposti dall'utente sulla forma di quest'ultima feature.

I nodi del modello B-rep sono riaggiornati e sostituiti dai nuovi nodi. Quando viene eseguita la sostituzione si devono aggiornare anche le relazioni stabilite dai puntatori di tipo *curve. In Fig. 6 si vede l'aggiornamento del modello a seguito della modifica dell'edge e_{11} della feature F_1 ; in particolare, le entità in grigio sono quelle che non appartengono più al modello B-rep. Inoltre si è supposto di avere un vincolo che impone la non modificabilità della feature F_4 e quindi la costruzione della superficie S'_2 risulta essere vincolata.

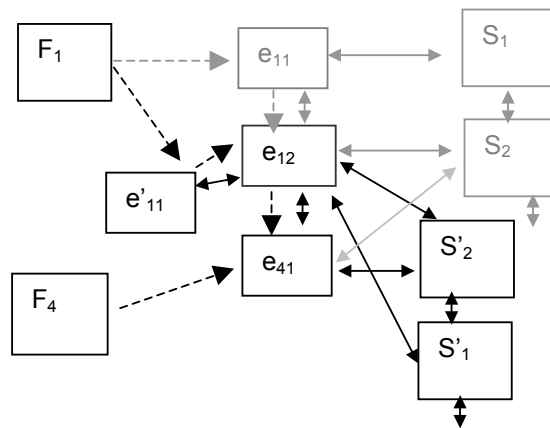


Fig. 6: Modifica di una feature strutturale

Ponendo in relazione i nodi rappresentanti le feature F_1 ed F_2 , che hanno la caratteristica di avere aree di influenza non distinte si può stabilire immediatamente come propagare i vincoli di forma in caso di modifica. In Fig. 6 non abbiamo aggiunto queste relazioni per non appesantire lo schema ma comunque si può sottolineare che ogni feature, tra quelle evidenziate, è in relazione di adiacenza (rispetto all'area di influenza) con tutte le altre.

Se l'utente decide di includere, all'area di influenza di default, altre superfici esse devono essere collegate alla feature attraverso il puntatore *sup che contraddistingue le superfici che sono solo temporaneamente coinvolte dalla modifica. Dopo aver effettuato la modifica, si deve aggiornare anche la lista di entità relazionate tramite il puntatore *sup eliminando quelle temporaneamente aggiunte.

Consideriamo ora il caso di una feature strutturale non di bordo mettendo in evidenza quali sono le problematiche che sorgono in presenza di entità virtuali. Nell'esempio si ha la feature F_5 che è contenuta nella superficie S_4 ma non fa parte della sua frontiera e quindi di essa non vi è traccia nel modello B-rep. L'utente ha creato F_5 per guidare l'andamento della superficie in prossimità della zona da essa occupata; F_5 è stata probabilmente utilizzata come elemento costruttivo per la creazione di S_4 .

Operativamente, l'utente definisce le entità che costituiscono F_5 e gli dà la connotazione di feature strutturale. Conseguentemente viene creato il nodo che rappresenta F_5 , con le corrispondenti relazioni con il modello B-rep. È opportuno osservare che, una volta creata S_4 , gli edge che costituiscono la feature, non fanno più parte del modello in quanto non delimitano alcuna superficie nonostante siano stati utilizzati come elementi di costruzione (eventualmente sono vincoli di passaggio).

Per non perdere le informazioni relative alla feature strutturale si devono memorizzare gli edge che la compongono e le relazioni con il modello B-rep indicando che la feature non appartiene più al modello ma è solo virtuale. L'utente deve comunque avere l'opportunità di modificare F_5 anche se è virtuale e poter ricostruire S_4 riconsiderando, eventualmente, il metodo costruttivo che è stato utilizzato. Vedremo nel capitolo riguardante le entità virtuali, come trattare questi casi.

4. Feature di dettaglio

Vengono considerate di dettaglio tutte le feature identificate nella tassonomia [2] con l'aggiunta dei **raccordi** (fillet, blending). Questi ultimi si considerano come feature vere e proprie; in particolare, i parametri necessari per istanziarli sono: edge del raccordo, continuità, raggio/raggi, sezioni.

In generale, nella rappresentazione di una feature di dettaglio si devono considerare i seguenti elementi:

- le superfici che compongono la feature (link alla geometria rappresentata nel modello B-rep),
- leading line,
- area di influenza (questa volta deve essere direttamente specificata dall'utente),
- condizioni di continuità,
- metodo di costruzione.

La leading line è quella curva (reale o virtuale) che guida l'andamento della feature di dettaglio, l'area di influenza è la zona effettivamente modificata dall'inserimento della feature, le condizioni di continuità sono quelle tra il bordo dell'area d'influenza ed il resto della superficie. Sfruttando tali informazioni si applica il metodo di costruzione per creare le superfici che costituiscono la feature.

Analogamente a quanto visto per le feature strutturali, una feature di dettaglio si può rappresentare graficamente nel seguente modo (Fig. 7):

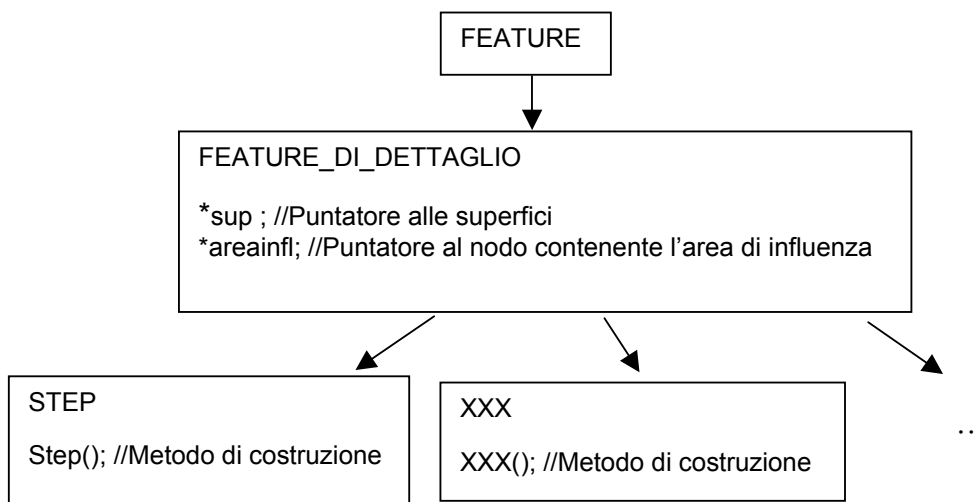


Fig. 7: Feature di dettaglio

Anche in questo caso, la classe rappresentante le feature di dettaglio deriva dalla classe base FEATURE e in particolare il puntatore `*curve` punta alla leading line, che poi non è altro che la linea di carattere della feature stessa. Inoltre:

- il puntatore `*sup` punta alle superfici che compongono la feature e che sono generate attraverso il metodo costruttivo (è quindi un puntatore ad elementi del modello B-rep).
- `*areainfl` è un puntatore alla regione di trimming ovvero la parte di superficie dell'oggetto che verrà sostituita con le superfici costituenti la feature. I nodi a cui punta `*areainfl` sono nodi virtuali in quanto dopo l'operazione di trimming rappresentano superfici non appartenenti al modello B-rep e pertanto andranno memorizzati. Per assegnare gradi di continuità sulle varie componenti della frontiera della feature si può aggiungere l'informazione ai nodi puntati da `*areainfl`.
- le classi rappresentanti le varie tipologie di feature di dettaglio, di cui qui abbiamo citato esplicitamente solo lo STEP, derivano dalla classe FEATURE_DI_DETAGLIO. Tali classi si distinguono essenzialmente per il metodo utilizzato per la loro costruzione. Ad ogni classe derivata dovranno essere aggiunti anche ulteriori parametri, dipendenti dalle caratteristiche di ogni feature, e necessari per applicare il metodo di costruzione. Si

possono comunque associare alle varie tipologie di feature di dettaglio dei parametri di default.

È evidente che nella rappresentazione di ogni feature di dettaglio ci sono sia elementi realmente appartenenti al modello B-rep (ad esempio le superfici che la compongono) che elementi virtuali (ad esempio, l'area di influenza o, in certi casi, la leading line).

Vediamo ora, analogamente a quanto fatto per le feature strutturali, come relazionare le feature di dettaglio nel modello strutturato.

4.1 Modifica ed interazione relative a feature di dettaglio

La modifica della superficie – ottenuta con la creazione della feature di dettaglio - in questo caso è a livello locale e quindi riguarderà una porzione piuttosto limitata della superficie.

La feature viene istanziata tramite i parametri di definizione e la modifica della stessa potrà essere effettuata semplicemente variando tali parametri. Ovviamente se si effettuano modifiche locali, come ad esempio lo spostamento di punti di controllo, si perde la connotazione di feature di dettaglio. In tal caso il nodo rappresentante la feature deve essere eliminato dalla rappresentazione feature-based oppure si può etichettare come entità non valida da riutilizzare per eventuali ricostruzioni, con l'avvertenza di avvisare l'utente che alcuni aggiornamenti (come lo spostamento di punti di controllo) non sono stati attuati. Per mantenere la tipologia di feature di dettaglio, la sua modifica potrà essere effettuata solo attraverso un'operazione di eliminazione e re-istanziamento della stessa con diversi parametri. Nella rappresentazione feature-based si dovranno aggiornare i parametri che sono variati mentre il modello B-rep viene aggiornato automaticamente.

Il problema che invece affrontiamo ora è l'interazione tra feature che può avvenire:

2. tra due o più feature di dettaglio,
3. tra feature strutturali e feature di dettaglio.

L'interazione tra due feature di dettaglio avviene solo nel caso in cui vi è un'intersezione tra le rispettive aree di influenza (tale area coincide ora, diversamente da quanto avviene per le feature strutturali, con l'area effettivamente occupata dalla feature).

La *scelta* che si è pensata più conveniente è quella di mantenere un ordine di creazione attraverso un grafo orientato che relazioni le feature con aree di influenza intersecanti. Se si modifica una feature di dettaglio, tutte quelle interagenti che sono state costruite successivamente dovranno essere ricalcolate. Quindi la feature che è modificata ha priorità massima rispetto a quelle

alle altre che dovranno adeguarsi alla nuova configurazione. In questo caso la priorità non è più run-time ma determinata dall'ordine di costruzione che è utile per gestire la propagazione delle modifiche e quindi ha senso che venga memorizzato.

La costruzione di una feature di dettaglio avviene tramite la sostituzione dell'area di influenza con le nuove superfici e quindi nel modello geometrico avviene una eliminazione e ricostruzione di superfici. Il problema è come ricalcolare le nuove feature al fine di aggiornare il modello a seguito di una modifica. Ad esempio, se si diminuisce l'ampiezza dell'area di influenza della feature direttamente modificata si deve recuperare l'informazione su com'era la superficie prima della creazione della feature stessa. Il problema è risolto utilizzando le informazioni memorizzate tramite il puntatore *areainfl. Con tale puntatore si mantengono, infatti, tutte le superfici coinvolte che possono pertanto essere riutilizzate.

Le feature di dettaglio interagenti con quella modificata devono essere ricalcolate tenendo presente che molto probabilmente l'area di influenza è cambiata e va pertanto ricalcolata. L'aggiornamento di tale area, che ricordiamo essere virtuale, si ottiene riapplicando il metodo utilizzato per determinarla.

È importante conoscere l'ordine di creazione anche perché se si considerano due feature di dettaglio interagenti, la prima che è stata creata perde le sue caratteristiche di forma nella zona compresa nell'area di influenza della feature con cui interagisce. In Fig. 8 si vede un esempio in cui le due feature di dettaglio interagiscono e in particolare F_1 è stata costruita prima di F_2 . Chiaramente F_2 modifica la forma di F_1 nella parte di quest'ultima compresa nell'area di influenza di F_2 .

Se si modifica F_1 è necessario andare a ricostruire F_2 riapplicando il metodo costruttivo e tenendo in considerazione che l'area di influenza è cambiata. Il collegamento tra le due feature verrà mantenuto a meno che non si effettui una modifica che comporti la disgiunzione delle due aree di influenza.

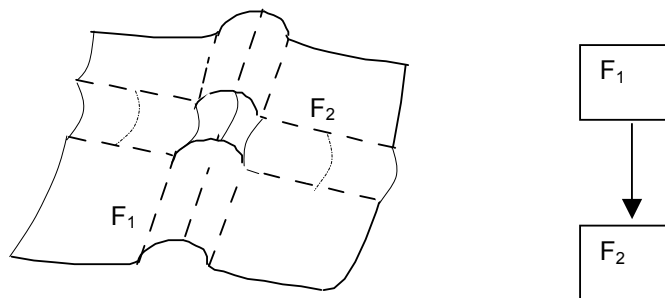


Fig. 8: Esempio di feature di dettaglio che interagiscono

Per quanto riguarda l'interazione tra feature strutturali e feature di dettaglio vediamo quali casi si possono verificare.

Il primo caso riguarda la costruzione di una feature di dettaglio intersecante una feature strutturale. Quest'ultima è già posizionata al momento di creare il dettaglio in quanto la sua natura è di essere strutturale e pertanto di essere creata nella fase iniziale di progettazione. Quindi, quando si aggiunge un dettaglio, la superficie sottostante alla feature di dettaglio è sicuramente già stata definita. La feature di dettaglio deve essere compatibile con tale configurazione.

Rimane da stabilire quali sono i vincoli da imporre alla feature strutturale che può venir modificata dall'inserimento del dettaglio. In tal caso (sempre che si voglia permettere la modifica strutturale) è necessario stabilire come riaggiornare il modello e come ricalcolare la nuova (aggiornata) feature strutturale. Per non perdere l'intento iniziale dell'utente, si può mantenere, tra le entità virtuali, anche la feature originariamente definita relazionandola con quella modificata.

Il secondo caso è relativo alla modifica di una feature di dettaglio in seguito alla modifica di una feature strutturale. La modifica può essere propagata nei due modi seguenti ed è l'utente che sceglierà il più appropriato.

La feature di dettaglio può seguire passivamente la modifica e in tal caso perderà le caratteristiche di feature con cui era stata definita inizialmente.

Quindi il nodo che la rappresenta dovrà essere eliminato dal modello feature-based lasciando inalterati i nodi relativi alle entità che la costituiscono in quanto appartenenti esclusivamente al modello B-rep. Ovviamente tutte le informazioni di creazione della feature (se non si ha un modello history-based) verranno perse e ulteriori modifiche della zona non potranno più essere supportate dall'approccio feature-based.

Se l'utente desidera invece mantenere la struttura di feature di dettaglio non si deve permettere la modifica diretta (tramite operatori di deformazione) delle superfici che la compongono e sarà necessaria una ridefinizione della feature che potrà avvenire secondo due modalità:

1. i nuovi parametri geometrici (area di influenza e leading line) saranno ottenuti come trasformati di quelli iniziali secondo la funzione di deformazione. Non sempre sarà possibile seguire questo approccio in quanto le nuove configurazioni possono essere non applicabili; ad esempio, la posizione della leading line può essere non compatibile con quella dell'area di influenza.
2. considerando il fatto che i parametri geometrici (curve su superfici) utilizzati nella definizione sono solitamente costruiti nel piano di vista e poi proiettati sulla superficie, si potrebbe semplicemente proiettare gli stessi parametri sulle nuove superfici e dopo di che ricalcolare la feature tramite il metodo di costruzione.

Nel primo caso, il metodo di costruzione viene applicato considerando i nuovi parametri geometrici mentre nel secondo caso viene applicato riutilizzando gli stessi parametri con cui la feature era stata inizialmente definita. Ovviamente, a seconda della modalità che si decide di seguire per ricreare la feature di dettaglio, si otterranno risultati diversi e quindi sarà l'utente a scegliere, tra i due approcci, quello che ritiene più opportuno.

4.2 Esempio

Consideriamo ora alcune feature di dettaglio in modo da esemplificare quanto detto.

In Fig. 9 viene mostrata la fiancata laterale di un'automobile in cui si possono distinguere tre diverse feature il cui indice segue l'ordine di creazione. In particolare:

- F_1 è una feature strutturale che delimita la portiera dell'auto;
- F_2 è una scanalatura che supponiamo essere ottenuta attraverso l'operatore di creazione di una specifica feature di dettaglio;
- F_3 è una feature di dettaglio che caratterizza la maniglia della portiera ed è ottenuta utilizzando il metodo di costruzione relativo al gap di dettaglio.

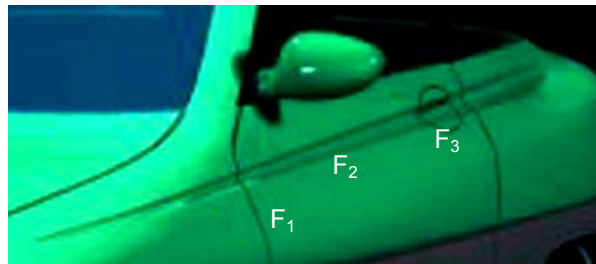


Fig. 9: Esempi di interazioni tra feature

Chiaramente la feature F_2 interagisce con le altre due. Se F_2 viene modificata, per esempio aumentando la sua profondità questo va ovviamente ad incidere sulla forma delle altre due feature.

Operativamente la nuova forma di F_2 verrà determinata riapplicando il metodo costruttivo tenendo conto del nuovo parametro. Per ottenere la nuova F_3 si deve invece calcolare la sua nuova area d'influenza e successivamente applicare il metodo costruttivo del gap di dettaglio. L'utente può decidere di far seguire passivamente alla feature F_3 , la modifica di F_2 , e in tal caso F_3 perde la sua connotazione di feature e quindi il nodo che la rappresenta va eliminato dal modello feature-based.

La geometria di F_1 verrà aggiornata tramite l'aggiornamento del modello B-rep ma la sua geometria originale (quella definita inizialmente dall'utente) deve essere mantenuta come entità virtuale in quanto tale feature è di tipo strutturale.

Graficamente la situazione si può rappresentare nel seguente modo (Fig. 10):

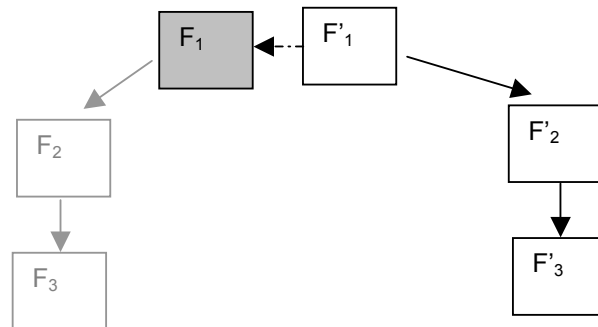


Fig. 10: Aggiornamento di feature a seguito di modifica

Le tre feature vengono aggiornate e così le loro relazioni: le feature indicate con l'apice sono quelle nuove.

La feature F_1 è stata indicata in modo diverso in quanto non appartiene più al modello ma deve essere mantenuta, come entità virtuale, in relazione con F'_1 ; infatti le informazioni relative ad F_1 devono essere memorizzate per conservare l'intento iniziale dello stilista.

Ovviamente si devono aggiornare anche tutti i collegamenti al modello B-rep che qui non abbiamo evidenziato.

5. Entità virtuali

È arrivato il momento di occuparci delle entità virtuali e cioè:

- feature strutturali che non sono di bordo (ad esempio, le linee di carattere e alcune sezioni),
- entità non appartenenti al modello ma che sono state utilizzate come parametri di costruzione per la creazione di feature di dettaglio (come, ad esempio, la leading line e l'area di influenza) o per la creazione di parti dell'oggetto (ad esempio, sezioni non identificate come feature strutturali).

Se la feature strutturale considerata è virtuale, e cioè non appartiene al modello B-rep, deve essere trattata con particolare attenzione affinché le informazioni che la riguardano non vengano perse. Per distinguere una feature virtuale da una reale (dove con reale si intende esistente nel modello B-rep) è sufficiente aggiungere nel nodo FEATURE un flag. Se tramite il flag si ha l'informazione che la feature è o è diventata virtuale allora i nodi a cui fa riferimento il puntatore *curve devono essere memorizzati con il loro contenuto.

Una feature strutturale virtuale può essere rappresentata nel seguente modo (Fig. 11):

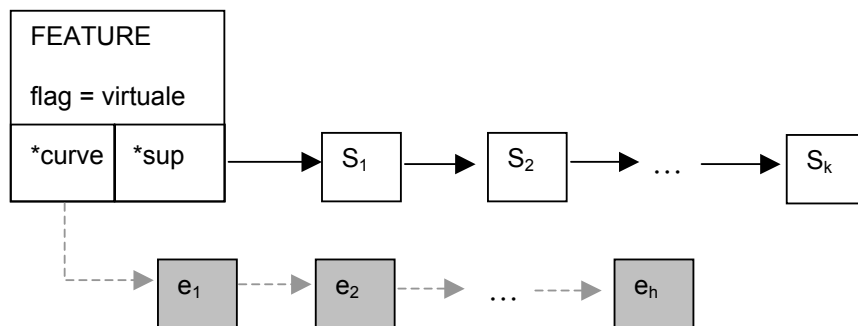


Fig. 11: Rappresentazione di una feature strutturale virtuale

I nodi e_i rappresentano gli edge (virtuali) che costituiscono la feature strutturale e memorizzano la geometria come avviene nella rappresentazione B-rep. I nodi S_j rappresentano invece le superfici originate a partire dalla feature e pertanto fanno parte del modello B-rep in cui sono memorizzate tutte le informazioni.

Un esempio di feature strutturale virtuale si ha se si considera come feature una sezione e se essa viene utilizzata dal modellatore come elemento di costruzione e cioè come curva da interpolare utilizzando un operatore per la creazione di superfici a partire da curve. In Fig. 12 sono messe in evidenza alcune feature strutturali che non sono di bordo.

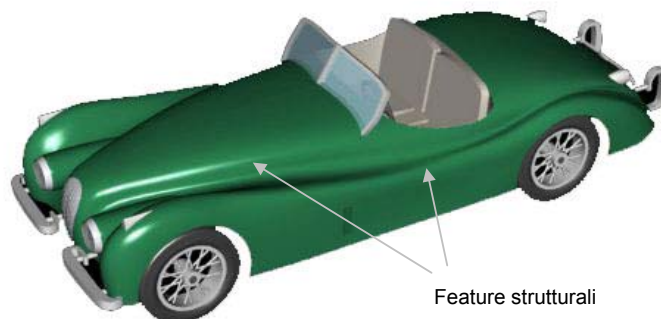


Fig. 12: Esempi di feature virtuali

Trattiamo ora il problema del riaggiornamento del modello qualora venga modificata una feature che non è di bordo e che quindi è di tipo virtuale. Nel modello B-rep non c'è traccia delle feature virtuali ed è proprio per eventuali manipolazioni che è stata aggiunta la memorizzazione degli elementi che le costituiscono.

Per aggiornare il modello si possono seguire due approcci. Se assumiamo che siano noti gli algoritmi di valutazione di una curva su superficie, si possono ricalcolare gli elementi che costituiscono la feature a partire dalla nuova superficie. Altrimenti, si può risalire alle nuove curve tramite il metodo costruttivo iniziale che quindi deve essere memorizzato. Oltre a tale metodo si devono mantenere anche tutte le entità che sono state utilizzate nella fase costruttiva della feature in modo da permetterne la ricostruzione riapplicando il metodo stesso.

Quest'ultimo problema si pone anche nel caso di entità virtuali che non siano feature e cioè per quelle linee di costruzione che permettono di creare la feature vera e propria (ad esempio, le curve che delimitano effettivamente il gap). Poiché esse non fanno parte del modello geometrico, devono essere memorizzate in qualche modo per poterle poi recuperare in fase di manipolazione.

Esistono anche entità che pur non appartenendo al modello geometrico sono state utilizzate come supporto per la creazione di entità che fanno parte del modello. Queste entità virtuali, se utilizzate per definire una feature strutturale devono essere comunque memorizzate, eventualmente riutilizzate in caso di modifica e riaggornate. In generale, per mantenere le linee di costruzione virtuali è necessario avere un modellatore history-based.

E' importante che nel nodo della feature venga memorizzato il **metodo costruttivo** delle superfici relative anche al fine di mantenere l'intento del progettista. Ciò è utile in modo particolare nel caso di manipolazione: una volta modificata la curva strutturale, si deve ricostruire la superficie sottogiacente rispettando il criterio con cui questa era stata costruita all'inizio, sempre che ciò sia possibile. La memorizzazione del metodo costruttivo del modello avviene con i modellatori history-based e quindi in questi casi dovrebbe essere semplice isolare la parte di metodo relativa ad ogni feature. Nel caso di un modellatore che non mantiene la storia di costruzione si deve implementare una storia locale che memorizzi i metodi costruttivi delle singole feature.

Vediamo ora alcuni esempi a questo proposito.

Nel caso delle sezioni, definite come intersezione della superficie con un piano fissato, ciò che deve essere mantenuto è il fatto che quella feature strutturale è una sezione. Il problema si pone quando viene modificata una feature strutturale in un'area di influenza in cui giace anche una sezione. Infatti, tale sezione viene modificata in conseguenza della modifica della superficie ma, in generale, la nuova linea non sarà più una sezione, nel senso che non sarà la soluzione dell'intersezione tra la nuova superficie ed il piano considerato all'inizio. L'aggiornamento di quella

sezione non implica quindi la sostituzione con quella modificata, ma con la curva ottenuta intersecando la nuova superficie con il piano. In questo modo, viene mantenuta la proprietà di essere sezione. Quindi per le sezioni non è necessario conoscere alcun metodo costruttivo ma solo qual è il piano di sezione e come aggiornare il nodo che lo rappresenta a seguito di modifica.

Se si considera una curva che è la direttrice o la sezione di uno sweep, si deve mantenere il fatto che la superficie è stata ottenuta utilizzando l'operatore di sweep poiché questo era l'intento dell'operatore. Le curve che definiscono lo sweep devono essere legate tra loro perché eventuali modifiche della superficie devono permetterne la ricostruzione riapplicando l'operatore di sweep.

Consideriamo ora una superficie costruita per interpolazione di feature strutturali. In tal caso, se si memorizza l'operatore di interpolazione usato, si può ricostruire la superficie riapplicando lo stesso operatore considerando che esso deve ora interpolare la curva modificata lasciando le altre intatte.

Per quanto riguarda le feature di dettaglio, quando vengono definite esiste il puntatore *areainfl che punta ad elementi che sono virtuali (l'area trimmata dal modello) e talvolta anche il puntatore *curve può puntare ad una leading line che non appartiene al modello. La rappresentazione di tali entità virtuali è già stata trattata nel capitolo relativo alle feature di dettaglio.

In particolare ricordiamo che: *areainfl sarà aggiornato ogniqualvolta si decide di modificare l'area influenzata dalla relativa feature di dettaglio e metterà in relazione sempre superfici di tipo virtuale, mentre se la leading line è un'entità virtuale essa verrà trattata in modo analogo a quanto avviene per le feature strutturali.

6. Conclusioni

La possibilità di avere un modello che mantiene la storia totale risolverebbe i problemi relativi alle interazioni tra le entità comprese nel modello geometrico.

L'idea dell'approccio seguito è invece quello di considerare un modello feature-based (sovrastuttura del modello B-rep) che include una storia parziale legata principalmente ai metodi di costruzione/modifica delle singole feature e al loro ordine di creazione.

Con il nostro approccio non si mantiene la storia completa in quanto non vengono memorizzate e organizzate tutte le operazioni di costruzione ma, in certi casi, è comunque richiesta la memorizzazione dei metodi di costruzione relativi alle feature strutturali.

I legami imposti alle feature strutturali vanno mantenuti per eventuali ricerche di pattern simili e per mantenere le relazioni tra le feature strutturali che hanno aree di influenza intersecanti. Per

quanto concerne le feature di dettaglio, esse hanno il vantaggio di agire come macro-operatori e di poter essere istanziate con facilità applicando il metodo di costruzione.

Bibliografia

- [1] J.J. Shah, M. Mantyla, *"Parametric and feature-based CAD/CAM"*, Wiley-Interscience Publication, John Wiley Sons, Inc., 1995
- [2] C. Catalano, M. Meirana, L. Papaleo. *"Structured modeling for aesthetic design"*, Spring Conference on Computer Graphics 2000, Bratislava, 3-6 Maggio 2000, pag. 267-274
- [3] M. Fontana, F. Giannini, M. Meirana; *"A Free Form Feature Taxonomy"*, Computer Graphics Forum, Vol. 18, n. 3, 1999, pag. 107-118
- [4] M. Fontana, F. Giannini, M. Meirana; *"Free Form Features for aesthetic design"*, accettato e in attesa di pubblicazione sulla rivista "International Journal of Shape Modeling
- [5] P.A. van Elsas, J.S. M. Vergeest, *"Displacement feature modelling for conceptual design"*, Computer-Aided Design Vol 30, N. 1, 1998, pp. 19-27
- [6] F.L. Krause, J. Luedemann, E. Rieger, *"Feature-Based Modelling of Conceptual Requirements for Styling"*, Proceedings IFIP WG 5.3, Life Cycle Modelling for Innovative Products and Processes, Berlin Noe./Dec. 1995, F. L. Krause and H.Jansen (Edts), London, Chapman Hall, 1996