

**O'ZBEKISTON RESPUBLIKASI OLIY VA O'RTA MAXSUS
TA'LIM VAZIRLIGI**

TO'LAYEV BEKMUROT

**LOYIHALASH JARAYONLARINI
AVTOMATLASHTIRISH ASOSLARI
CHIZMALARNI AVTOMATLASHTIRILGAN
ISHLAB CHIQISH TIZIMLARI**

500000 – «Muhandislik, ishlov berish va qurilish tarmoqlari» ta'lim
sohasi yo`nalishlari talabalari uchun o`quv qo`llanma

TOSHKENT 2009

A n n o t a t s i y a

O`quv qo`llanmada chizmalarni avtomatlashtirilgan ishlab chiqish prinsiplari bayon qilingan, tizimlar konsepsiyasi muayyan paketlarda ishlash bilan bog`liq bo`lgan juz`iy detallarga e`tibor berilmasdan tahlil etilgan. Mavjud dasturiy ta`minot bilan samarali ishlash va loyihalash jarayonini avtomatlashtiruvchi dasturlarni tuzish uchun foydalanuvchi nafaqat o`zi ishlaydigan muhit haqida, balki bu muhit asosida yotuvchi prinsiplar haqida ham tasavvurga ega bo`lishi kerak. Fundamental bilimlar talabaga muayyan muhitdagi istalgan muayyan tizimni tez o`zlashtirishga va o`zlashtirgan bilimlaridan maksimal darajada effektiv foydalanishga yordam beradi.

А н н о т а ц и я

В учебном пособии изложены принципы автоматизированной разработки чертежей, описаны концепции систем, не вдаваясь в конкретные детали, связанные с работой в конкретных пакетах. Для эффективной работы с существующим программным обеспечением и создания программ, автоматизирующих процесс проектирования, пользователь должен иметь представление не только о среде, в которой он работает, но и о принципах, лежащих в её основе. Фундаментальное знание помогает студенту быстро изучить любую конкретную систему с конкретной средой и использовать её максимально эффективно.

T h e s u m m a r y

In the manual principles of the automated working out of drawings are stated, concepts of systems are described, not pressing in the concrete details connected with work in concrete packages. For effective work with the existing software and creations of the programs automating process of designing, the user should know not only about Wednesday in which it works, but also about the principles laying in its basis. The fundamental knowledge helps the student to study quickly any concrete system with the concrete environment and to use it as much as possible effectively.

Taqrizchilar: t.f.d., dots. Bazarov B.I. (TAYI);

t.f.d., prof. Mamadjanov A.M. (ToshDTU)

MUQADDIMA

Kompyuterlar hisoblash quvvatining hayratli darajada ortishi hamda loyihalash va ishlab chiqarish dasturaviy ta`minotining keng tarqalishi shunga olib keldiki, muhandislar o`zlarining kundalik vazifalarini yechishda avtomatlashtirilgan loyihalash tizimi (ALT)dan foydalanish imkoniyatiga ega bo`ldilar. Xalqaro raqobat, tajribali mutaxassislar sonining tobora ortib borishi va sifatga qo`yiladigan talablarning ortishi korxonalar rahbarlarini loyihalash va ishlab chiqarishni avtomatlashtirishga majbur qilmoqda. Buning oqibatida oliy maktab o`quvchilari loyihalashga taalluqli bo`lgan o`quv fanlarining dasturlarini o`zgartirish ehtiyojini sezishmoqda. Bu o`zgartirishdan maqsad – talabalarni ALTdan foydalanishga o`rgatish va ularda bu tizimlar asosida yotgan asosiy prinsiplar haqida tasavvur hosil qilishdir.

Bu o`quv qo`llanmaning maqsadi – talabalarda muayyan paketlarda ishlash bo`yicha amaliy malaka va ko`nikmalari hosil qilish bilan birga, bu prinsiplarni va konsepsiyalarni bayon qilishdir. Ba`zilarga talabalarni mavjud tizimlardan yoki hatto eng ko`p tarqalgan bitta tizimdan foydalanishga o`rgatish kifoya deb tuyulishi mumkin, chunki muhandislik ishi sohasidagi talaba ALTni ishlab chiquvchi emas, balki undan foydalanuvchi bo`ladi. Lekin gap shundaki, mavjud dasturaviy ta`minot bilan samarali ishlash va loyihalash jarayonini avtomatlashtiruvchi dasturlarni yaratish uchun, foydalanuvchi nafaqat o`zi ishlayotgan muhit haqida, balki bu muhitning asosida yotgan prinsiplar haqida tasavvurga ega bo`lishi kerak. Fundamental bilim talabaga muayyan muhitli istalgan muayyan tizimni tez o`rganib olishga va undan maksimal darajada samarali foydalanishga yordam beradi. Nazariy tayyorgarliksiz talaba tizimiy dokumentatsiya atamalarida qiyinchiliklarga uchraydi, xatoliklar haqidagi xabarlarini tahlil qilishda esa u yana ko`proq qiyinchiliklarga uchraydi.

O`quv qo`llanma oliy texnikaviy o`quv muassasalarida tahsil olayotgan talabalar uchun yozilgan. U bilan ishlash uchun dasturlash asoslarini, matematik

analizni, matritsali va vektorli algebrani bilish kifoya qiladi; talabada ALT bo`yicha bilimlar bo`lishi talab etilmaydi.

Har bir bobning oxirida masalalar keltirilgan, ularning vazifasi – talabalar tomonidan materialni o`zlashtirish sifatini tekshirishdir.

1 – BOB. ALTGA KIRISH

Zamonaviy korxonalar agar *eng yaxshi sifatli, tannarxi arzonroq bo`lgan*, yangi mahsulotni qisqa vaqt oralig`ida tayyorlab chiqarishmasa, ular butun dunyo raqobatiga bardosh bera olmaydilar. Shu sababli ular loyihalash va ishlab chiqarish masalalarini avtomatlashtirish va bir-biriga bog`lashda grafik interfeysning ulkan imkoniyatlaridan foydalanishga intilishmoqda. Bunda yangilikni va mahsulotni ishlab chiqarish vaqti qisqaradi, tannarxi arzonlashadi. Bu maqsadda *avtomatlashtirilgan loyihalash (computer-aided design – CAD)*, *avtomatlashtirilgan ishlab chiqarish (computer-aided manufacturing – CAM)* va (chizmalarni) *avtomatlashtirilgan ishlab chiqish yoki konstruksiyalash (computer-aided engineering – CAE)* texnologiyalaridan foydalaniladi.

Avtomatlashtirilgan loyihalash (CAD) – bu loyihalarni yaratish, o`zgartirish, tahlil qilish va optimallashtirishni osonlashtirish uchun kompyuter tizimlaridan foydalanishga asoslangan texnologiyadir [14]. Demak, kompyuter grafikasi bilan ishlovchi istalgan dastur, muhandislik hisoblarida foydalaniladigan istalgan ilova kabi, avtomatlashtirilgan loyihalash tizimlariga taalluqli bo`ladi. Boshqacha aytgagnda, *CAD*ning ko`p vositalari shakllar bilan ishlash uchun mo`ljallangan geometrik dasturlardan, to tahlil qilish va optimallashtirish uchun ixtisoslashgan ilovalargacha yoyilgan [17]. *CAD*ning eng asosiy vazifasi – bu konstruksiya (mexanizm detallari, arxitektura elementlari, elektron sxemalar, binolar planlari va sh.k.)ning geometriyasini aniqlashdir, chunki geometriya mahsulot hayotiy siklining hamma bo`lajak bosqichlarini belgilaydi. Odatda, bu maqsad uchun, chizmalarni ishlab chiqish tizimlari va geometrik modellashtirishdan foydalaniladi. Shu sababli bu tizimlar odatda avtomatlashtirilgan loyihalash tizimi deb hisoblanadi. Bundan tashqari, bu tizimlarda aniqlangan geometriya *CAE* va *CAM* tizimlarida keyinchalik bajariladigan operatsiyalar uchun asos sifatida foydalanishi mumkin. Bu – *CAD*ning eng ahamiyatli afzalliklaridan biridir, vaqtni tejash hamda hisoblashda har gal konstruksiyani nuldin boshlab aniqlash zarurati bilan bog`liq bo`lgan xatoliklar sonini qisqartirish imkonini beradi. Demak, shuni ta`kidlash

mumkinki, ishchi chizmalarni avtomatlashtirilgan ishlab chiqish tizimlari va geometrik modellash tizimlari – avtomatlashtirilgan loyihalashning eng ahamiyatli komponentlaridir.

Avtomatlashtirilgan ishlab chiqarish (CAM) – bu korxonaning ishlab chiqarish resurslari bilan bevosita yoki bilvosita interfeys orqali ishlab chiqarish operatsiyalarini rejalash, boshqarish asoslangan texnologiyadir. Sonli-raqamli dasturaviy boshqarish (*numerical control – NT*) – ishlab chiqarishni avtomatlashtirishga boʻlgan eng yyetuk yondoshuvlardan biridir. Sonli-raqamli boshqaruv – bu stanokni boshqarish uchun dasturlashtirilgan komandalardan foydalanishdir; stanok jilvirlash, kesish, frezerlash, shtamplash, parmalash va boshqa usullar bilan xomakini tayyor detalga aylantiradi. Hozirgi paytda kompyuterlar *CAD* maʼlumotlar bazasidagi buyumlarning geometrik parametrlari va operator taqdim etayotgan qoʻshimcha maʼlumotlar asosida sonli-raqamli dasturaviy boshqariluvchi stanoklar uchun katta dasturlarni generatsiya qilishga qodir.

Avtomatlashtirilgan ishlab chiqarish tizimlarining yana bir ahamiyatli funksiyasi – robotlarni dasturlashdir, ular sonli-raqamli dasturaviy boshqariluvchi stanoklarga asboblari va ishlov beriladigan detallarni oʻrnatib, avtomatlashtirilgan moslanuvchan uchastkalarda ishlashi mumkin. Robotlar oʻzlarining xususiy vazifalarini ham bajarishi, masalan, payvandlash, jihozlari va detallarni yigʻishi va ularni sex boʻylab tashishi mumkin.

Jarayonlarni rejalashtirish ham asta-sekin avtomatlashtirilmoqda. Jarayonlar rejasi qurilmani tayyorlash boʻyicha operatsiyalar ketma-ketligini boshidan oxirigacha hamma zarur boʻlgan jihozlarda aniqlash mumkin. Jarayonlarni rejalashtirishni toʻliq avtomatlashtirish amalda muvqin boʻlmasa ham, muayyan detalga ishlov berish rejasi, agar shunga oʻxshash detallarga ishlov berish rejasi bor boʻlsa, avtomatlashtirilishi mumkin. Buning uchun guruhlash texnologiyasi ishlab chiqilgan, u oʻxshash detallarni oilalarga birlashtirish imkonini beradi. Agar detallar umumiy ishlov berish xususiyatlari (oʻrindiqlar, pazlar, faskalar, teshiklar va h.k.)ga ega boʻlsa, ular oʻxshash hisoblanadi. Detaillar oʻxshashligini bilib olish

uchun *CAD* ma'lumotlar bazasi shunday xususiyatlar haqida ma'lumotlarga ega bo'lishi kerak. Bu vazifa obyektga yo'nalgan modellash yoki elementlarni tanish yordamida amalga oshiriladi.

Avtomatlashtirilgan konstruksiyalash (*computer-aided engineering – CAE*) – bu *CAD* geometriyasini tahlil qilish, modellash va mahsulot konstruksiyasini takomillashtirish uchun uning xususiyatlarini o'rganish uchun kompyuter tizimidan foydalanishdir. *CAE* vositalari tahlilning har xil variantlarini bajarishi mumkin. Masalan, kinematik hisoblash dasturlarini harakat trayektoriyalarini va mexanizmlardagi zvenolar tezliklarini aniqlashga qodir. Dinamik tahlil dasturlari avtomobillar kabi murakkab tarkibli qurilmalarda yuklanish va siljish (деформация)larni aniqlash uchun qo'llanilishi mumkin. Mantiqni tahlil qiluvchi va sinxronizatsiyalovchi dasturlar murakkab elektron zanjirlar ishini imitatsiya qiladi.

Konstruksiyalarni optimizatsiyalashga mo'ljallangan ko'p dasturaviy vositalar mavjud. Optimizatsiyalashga mo'ljallangan vositalar *CAE* sinfiga kiritilishi mumkin bo'lsa ham, odatda ularni alohida ko'rib chiqishadi. Optimizatsiyalash va tahlilni birlashtirish yo'li bilan konstruksiya shaklini avtomatlashtirilgan tarzda aniqlash bo'yicha tadqiqotlar olib borilmoqda [13]. Bu yondoshuvlarda konstruksiyaning boshlang'ich shakli sodda ko'rinishda taklif qilinadi, masalan, zichligi har xil bo'lgan katta bo'lmagan elementlardan tarkib topgan ikki o'lchamli to'g'ri to'rt burchakli obyekt kabi; bunday shakl kuchlanishlarga qo'yilgan cheklashlarni hisobga olgan holda muayyan maqsadga erishish imkonini beradi. Maqsad – ko'pincha og'irlikni minimallashtirish bo'ladi. Zichlikning optimal qiymatlari aniqlangandan so'ng, obyektning optimal shakli hisoblanadi. Bunga zichlik qiymati kichik bo'lgan elementlarni tashlab yuborish bilan erishiladi.

Tahlil va optimallashtirish metodlarining ajoyib xususiyati shundaki, ular konstruktorga tayyor mahsulot sifat (xossa)larini oldindan ko'rabilish va real prototiplarni yaratish va testdan o'tkazguncha mumkin bo'lgan xatoliklarni namoyon qilish vash u bilan ma'lum xarajatlarga yo'l qo'ymaslik imkonini beradi.

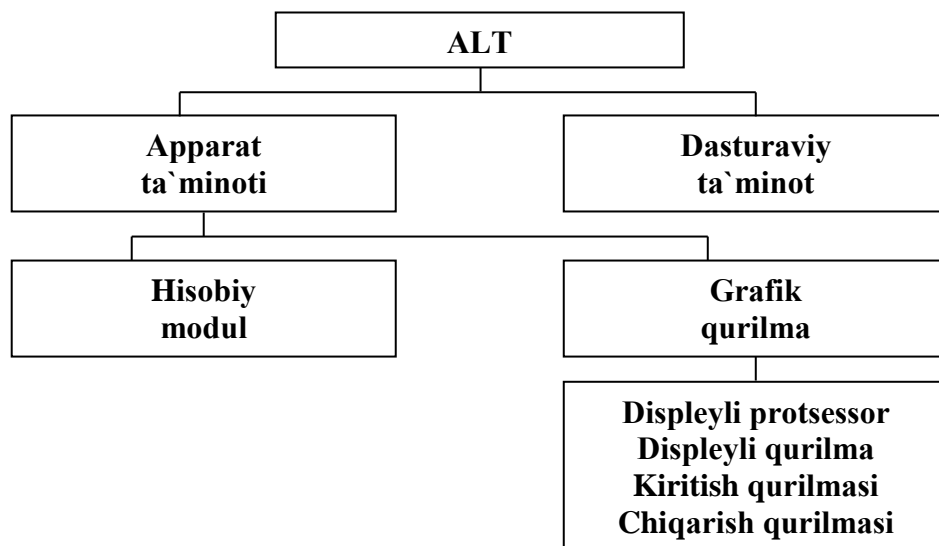
Demak, *CAD*, *CAM* va *CAE* texnologiyalarining vazifasi – mahsulot hayotiy siklining muayyan bosqichlarini avtomatlashtirish va samaradorligini oshirishdir.

Savollar va topshiriqlar

1. Loyihaviy va analitik modellar orasidagi farqni bayon qiling.
2. Nima uchun analitik model loyihaviydan farqlanadi?
3. Ishlab chiqish jarayoni doirasida qanday analitik operatsiyalar bajariladi?
4. Ishlab chiqish jarayonida *CAD* vositalaridan qanday foydalaniladi?
5. Ishlab chiqish jarayonida *CAD* vositalarining qanday variantidan foydalanish eng ahamiyatli hisoblanadi?
6. Ishlab chiqarish jarayonida *CAM* vositalaridan qanday foydalaniladi?
7. Ishlab chiqarish jarayonida *CAE* vositalaridan foydalanishning asosiy afzalligi nimada?

2 – BOB. ALT KOMPONENTLARI

Loyihalashga kompyuterga yo`nalgan yondoshishni amalga oshirish uchun maxsus apparat va dasturaviy ta`minot zarur. Shaklni interaktiv boshqarish bunda tayanch aspekt hisoblanadi, shu sababli shakllarni interaktiv o`zgartirish uchun apparat va dasturaviy ta`minot *CAD/CAM/CAE* tizimlarni tashkil qiluvchi asosiy komponentlar qatoriga kiradi. Grafik qurilmalar va kiritish-chiqarishning periferiya qurilmalari oddiy hisoblash moduli bilan birga *CAD/CAM/CAE* tizimlarining apparat ta`minotini tashkil qiladi. (2.1-rasm). Shakllarning ikki yoki uch o`lchamda o`zgartiruvchi yoki ularni foydalanuvchi boshqaruvida tahlil qiluvchi va shu vaqtning o`zida ma`lumotlar bazasini yangilovchi paketlar tayanch (bazaviy) dasturaviy komponentlar hisoblanadi:



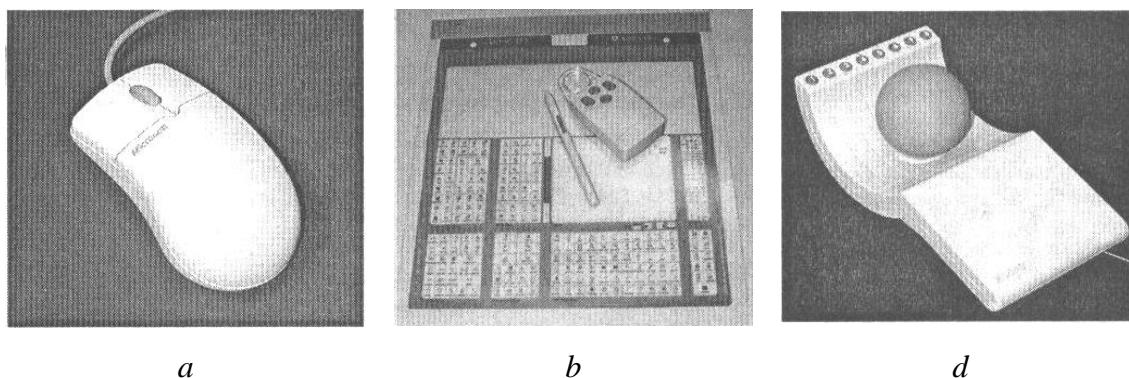
2.1-rasm. *CAD/CAM/CAE* tuzilmalarining komponentlari

2.1. Apparat ta`minoti

Grafik qurilma displeyli protsessor, aks ettiruvchi qurilma yoki displeyli qurilma (монитор) va bir yoki bir nechta kiritish qurilmalaridan tarkib topadi. Displey (экран) bo`lib, unga grafik tasvir chiqariladi, lekin muayyan tasvirni ekranga displeyli protsessor chiqaradi. Boshqacha aytganda, displeyli protsessor grafik komandalar kodlangan signallarni qabul qiladi, unda elektron nurlari

to'plamlari generatsiya qilinadi va bu nurlar monitoring kerakli joyiga yo'naltirilad, natijada ekranda istalayotgan tasvir hosil bo'ladi.

Grafik qurilma tarkibiga odatda bir yoki bir nechta kiritish qurilmalari kiradi. Unga klaviaturadan tashqari sichqoncha, speysbol va sonli-raqamli planshet (pero va roliki bilan birga) kiradi (2.2-rasm). Bu kiritish qurilmalarining vazifasi – shakllarni interaktiv rejimda o'zgartirishga ko'maklashishdir, bunda foydalanuvchiga grafik ma'lumotlarni kompyuterga bevosita kiritish imkoniyati beriladi. Har bir grafik qurilma odatda chiqarish qurilmalariga, masalan plotterga yoki rangli lazerli printeriga ulanadi. Bu qurilmalardan bir nechta grafik qurilmalar birgalikda foydalanishi mumkin.



2.2-rasm. Kiritish qurilmalari:

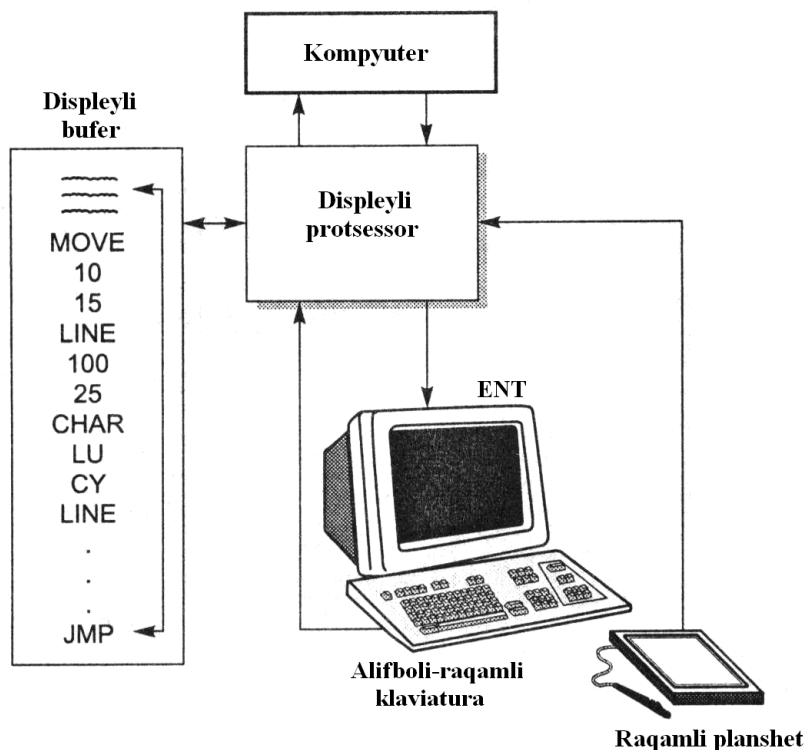
a – sichqoncha; *b* – sonli-raqamli planshet pero va rolik bilan birga; *d* – speysbol

Vektorli grafik qurilmalar XX asrning 60-yillarida paydo bo'ldi. Ular displeyli protsessor, xotiraning displeyli buferi va elektron-nurli trubkadan tarkib topadi (2.3-rasm).

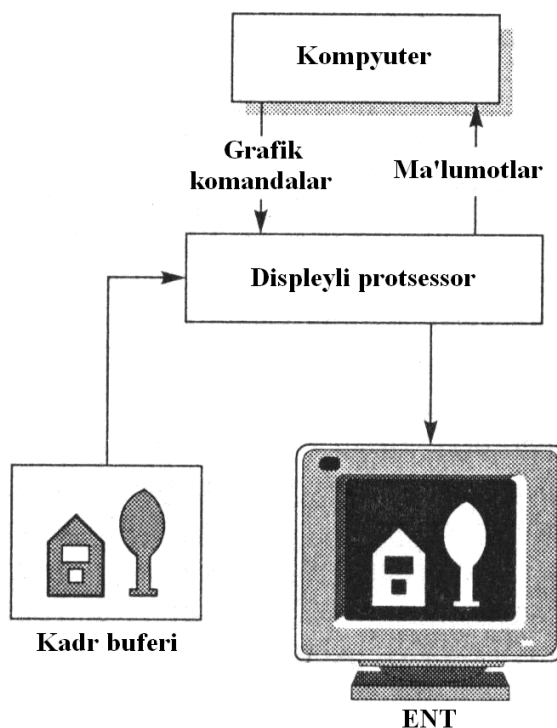
Rastrli grafik qurilmalar televizion texnologiyalarning keng tarqalishi natijasida XX asrning 70-yillarida paydo bo'ldi. «Sifat-narx» nisbati yuqori bo'lganligi tufayli, ular grafik qurilmalarning asosiy prinsiplarini qisqacha quyidagicha bayon qilishi mumkin.

Displeyli protsessor grafik komandalarni ilovadan oladi, ularni nuqtaviy tasvir-rastrga o'zgartiradi, bundan keyin rastrni xotiraning *kadr buferi (frame buffer)* deb nomlanuvchi bo'limida saqlaydi (2.4-rasm). Agar televizorga juda

yaqin kelinsa, rastrli tasvir haqida tasavvur hosil qilish mumkin. Nuqtalarning o'lchamlari o'rnatilgan ruxsat (разрешение) bilan belgilanadi. Rastrli grafik qurilmalar o'zining xotirasida tasvirni rastr ko'rinishida saqlaydi, vektorli qurilmalar esa xotirada faqat displeyli fayllarni saqlaydi. Shu sababli bu ikki turdagi qurilmalarda ularning xotiralariga qo'yiladigan talablar va ekrandagi tasvirni yangilash metodlari bir-biridan farq qiladi.



2.3-rasm. Vektorli grafik qurilma komponentlari



2.4-rasm. Rotorli grafik qurilma komponentlari

2.2. Apparat vositalarining konfiguratsiyasi

Oldingi paragrafda bayon qilingan grafik qurilmalar alohida-alohida kam ishlatiladi. Ko`pincha ular ko`p foydalanuvchilarga xizmat ko`rsatishga mo`ljallangan u yoki bu turdagi klasterga jamlanadi. Bunday klaster konfiguratsiyalarining uch asosiy varianti mavjud.

Birinchi konfiguratsiya meynfreyli (*mainframe*) va ko`pchilik grafik qurilmalardan tarkib topadi (2.5-rasm). Oddiy hisoblash markazlarida alfavitli-raqamli terminallar qanday ulansa, grafik qurilmalari ham meynfreynga aynan shunday ulanadi. Unga printer va plotter kabi chiqaruvchi qurilmalar ham ulanadi. Avtomobil ishlab chiqaruvchilari shu paytgacha bunday yondashuvdan foydalanishadi; ularda markazlashtirilgan ravishda ishlov beriladigan juda katta ma`lumotlar bazalari mavjud. Lekin bunday yondashuvda apparat va dasturaviy ta`minot katta boshlang`ich kapital jamlanmalarni talab qiladi, ekspluatatsion tizimga xizmat ko`rsatish ham arzon bo`lmaydi. Meynfreynga xizmat ko`rsatish tizimiy xotira va qattiq diskni kengaytirishni doim o`z ichiga oladi, bu katta bo`lmagan kompyuterlarga nisbatan meynfrey uchun ancha qimmatga tushadi. Bundan tashqari operatsion tizimni yangilash ham oson masalalardan emas.

CAD/CAM/CAE dasturlari tez-tez almashtirilib turishini talab qiladi, chunki yangi quvvatliroq versiyalar va alternativalar chiqarib turiladi, bundan tashqari dasturaviy ta`minot birinchi tanlanishida xatolikka yo`l qo`yilishi mumkin. Meynfreymlar uchun *CAD/CAM/CAE* dasturlari kichik kompyuterlar uchun dasturlarga nisbatan ancha qimmat turadi. Markaziy hisoblashlarning yana bir jiddiy kamchiligi – tizim javob berishi (отклик) vaqtning nostabilligidir. Meynfreyimli konfiguratsiyada turli grafik qurilmalarga taalluqli bo`lgan foydalanuvchilarning ilovalari meynfreymning hisoblash resurslari uchun bir-biri bilan raqobatlashadi. Shu sababli istalgan muayyan grafik qurilmalari uchun chaqiriqqa javob vaqti boshqa qurilmalardan qanday masalalar ishga tushirilib yuborilishiga bog`liq. Ba`zan grafika bilan interaktiv ishlash uchun chaqiriqqa javob vaqti haddan tashqari katta bo`lishi mumkin, ayniqsa boshqa foydalanuvchilar murakkab hisoblash masalalarini yechishayotgan bo`lsalar.

| Meynfreym | | | |
|--|--|--|----------------|
| Grafik qurilma | Grafik qurilma | Grafik qurilma | Plotter |
| Interaktiv kiritish qurilmalari | Interaktiv kiritish qurilmalari | Interaktiv kiritish qurilmalari | |

2.5-rasm. Meynfreyimli va uning grafik qurilmalari

Ikkinchi konfiguratsiya tarmoqqa birlashtirilgan loyihalovchining avtomatlashtirilgan ishchi joylari (ishchi stansiyalar – *workstations*)dan tashkil topadi. Chiqarish qurilmalari – plotterlar va printerlar ham ushbu tarmoqqa ulanadi (2.6-rasm). Ishchi stansiya – bu o`zining hisoblash resursiga ega bo`lgan grafik qurilmalardir. Hozirgi paytda bunday yondashuvdan juda keng foydalanilmoqda, chunki ishchi stansiyalarni tayyorlash borasida progress juda tez odimlamoqda. Ishchi stansiyalarning narxi saqlangan holda ularning unumdorligi har yili ikki marotaba ortmoqda. Bunday yondashuvning boshqa afzalliklari ham bor. Foydalanuvchi tarmoqdagi istalgan stansiya bilan ishlashi mumkin, u stansiyaning

o`zi yechayotgan masalaga muvofiq tanlaydi, bunda tizimiy resurslar boshqa foydalanuvchilar masalalariga bog`liq bo`lmaydi. Bunday yondashuvning yana bir afzalligi – boshlang`ich yirik mablag`lar talab qilinmaydi. Ishchi stansiyalar va dasturaviy paketlar soni, *CAD/CAM/CAE* resurslariga bo`lgan ehtiyoj ortgani sari, asta-sekin ko`payib borishi mumkin. Bu juda qulay, chunki jihozlar narxi yildan-yilga arzonlashib bormoqda.

| | | | |
|----------------------|--------------------------------------|--------------------------------------|----------------|
| Faylli server | Muhandislik ishchi stansiyasi | Muhandislik ishchi stansiyasi | Plotter |
| | Interaktiv kiritish qurilmasi | Interaktiv kiritish qurilmasi | |

2.6-rasm. Tarmoqqa biriktirilgan ishchi stansiyalar

Uchinchi konfiguratsiya ikkinchisiga o`xshash, farqi shundaki, ularga ishchi stansiyalar o`rniga *Windows...* va *NT* operatsion tizimli personal kompyuterlardan foydalaniladi. Personal kompyuterlar bazasidagi konfiguratsiyalar katta bo`lmagan kompaniyalarda, ayniqsa chiqariladigan mahsulotlar murakkb bo`lmagan soni kam bo`lgan detallardan tashkil topgan bo`lsa hamda *AutoCAD* tizimidan asosan chizmalarni qurish uchun foydalanadigan kompaniyalarda keng qo`llaniladi. Personal kompyuterlar va ishchi stansiyalar orasidagi farq kamayib borgani sari, konfiguratsiyaning ikkinchi va uchinchi turlari orasidagi tafovut ham kamayib boradi.

2.3. Dasturaviy komponentlar

Mahsulotni ishlab chiqarish vaqtini qisqartirish va narxini arzonlashtirish hamda sifatini yaxshilash uchun mahsulotning hayotiy siklida foydalanuvchi istalgan dastur *CAD/CAM/CAE* sinfiga kiritilishi mumkin. Ularning asosida *CAD* dasturlari yotadi, ular konstruktorga shakllarni yaratish va ularni monitorda interaktiv rejimda o`zgartirish hamda natijalarni ma`lumotlar bazasida saqlash imkonini beradi. Lekin chizmalarni ishlab chiqish jarayonini osonlashtiruvchi istalgan dastur *CAD* dasturi deb atalishi mumkin. Masalan, muayyan detal yoki

mexanizmni loyihalashni avtomatlashtirish uchun mo`ljallangan ixtisoslashgan ilova ham *CAD* ilovasi hisoblanadi.

Mahsulotni ishlab chiqarish jarayonida foydalaniladigan istalgan dastur *CAM* vositasi hisoblanadi. Demak, ishlab chiqarish operatsiyalarini korxonaning ishlab chiqarish resurslari bilan bevosita yoki bilvosita kompyuter interfeysi orqali rejalash, boshqarish va nazorat qilish uchun dasturlar *CAM* ga taalluqli bo`ladi. Bu detalni ishlab chiqarish jarayoni rejasini shakllantiruvchi dastur yoki sonli-raqamli dasturaviy boshqariluvchi stanok uchun dasturni yozadigan, rezets harakatini modellaydigan, detal sirtlariga ishlov berish mumkin.

CAE dasturi konstruksiya geometriyasini tahlil qilish uchun foydalaniladi, u ishlab chiquvchiga loyihani yaxshilash va optimallashtirish uchun mahsulotni modellashtirish va uni o`rganish imkonini beradi. Detaldagi kuchlanishlarni, deformatsiyalarni va issiqlik uzatilishini chet unsurlar metodi bilan hisoblash uchun dastur bunga misol bo`ladi.

CAD tizimida yaratilgan shakl freza harakati trayektoriyasini hisoblovchi yoki kuchlanishlarni uch o`lchamli tahlil qiluvchi ilovalar uchun faqat shu holda foydalanishligi mumkinki, agar ular uch o`lchamli qilib tayyorlangan bo`lsalar. Shu sababli konstruktorlar ko`p hollarda geometrik modellashtirish tizimida detalning uch o`lchamli modeli bilan ishlashni boshlaydilar.

2.4. Windows bazasida ALT

CAD/CAM/CAE dasturaviy ta`minotning bozori yetuklikka erishgandan so`ng, undagi vaziyat keskin o`zgaradi. Muhandislar va ishlab chiqaruvchilar ikki o`lchamli chizmalarni qurish vositalariga nisbatan ularga yanada imkoniyati kattaroq narsa zarurligini sezishdi. Ular loyihalashdan boshlab to ishlab chiqarishgacha bo`lgan uch o`lchamli chizmalarni quruvchi moslanuvchan tizimlar paydo bo`lishini kutishdi. Yaqin paytlargacha oliy sinfli *CAD* ning an`anaviy bozorida sanoat ilovalari hokimlik qiladi. Baxtimizga personal kompyuterlar vaqt o`tishi bilan haddan tashqari tezkor va quvvatli bo`lib qoladi, dasturaviy ta`minotni ishlab chiqaruvchilar esa *Microsoft Windows 95* va *NT* ajoyib grafik muhitining

afzalliklaridan foydalanuvchi yaxshi dasturaviy mahsulotlar ishlab chiqara boshlashdi. Bu kategoriyaga taalluqli birinchi dasturaviy mahsulotlar 1995 yilda chiqarildi, ko`pchilik mahsulotlarning birinchi versiyalari esa 1996 yilda chiqarildi.

Hamma yangi mahsulotlar quyidagi umumiy xususiyatlarga ega edilar.

*Birinchi*dan, ular *Windows 95* va *Windows NT* funksiyalaridan maksimal darajada foydalanilgan holda ishlab chiqildi, shu sababli ularning interfeyslari *Microsoft* boshqa dasturlarning interfeyslariga o`xshab chiqdi. Amaldagi bu dasturlar *Microsoft*ning idora mehnatini avtomatlashtirish uchun chiqargan boshqa mahsulotlariga o`xshar edi, shu sababli foydalanuvchilar tanish muhitda o`zlarini komfortli his qilishdi va ularni tez o`zlashtirib olishdi. Bundan tashqari, yangi dasturlar *Microsoft*ning ofis paketlari uchun xarakterli bo`lgan *obyektlarni tatbiq qilish va bir-biriga bog`lash (object Linking and Embedding – OLE)* funksiyalarini quvvatlashar (поддерживали) edi. Shunday qilib, geometrik modellashtirish paketida yaratilgan, uch o`lchamli detal yoki qurilmaning istalgan tasviri *Microsoft*ning boshqa dasturlari tomonidan foydalanishi mumkin.

*Ikkinchi*dan, yangi tizimlarda komponentli yondashuvdan foydalanildi, bu yondashuvda dasturaviy ta`minotning eng ahamiyatli komponentlari mavjud dasturlardan tanlanadi, so`ngra tizimni ishlab chiquvchi sinovdan o`tgan texnologiyalarni birlashtiradi xolos, bunda u o`zining e`tiborini bevosita loyihalashga taalluqli bo`lgan detallarga qaratadi. Bunday yondashuv ishlab chiqishga ketadigan vaqtni tejash imkoniyatini beradi, bunda foydalanuvchiga muayyan mahsulotni loyihalash jarayonida unga aynan zarur bo`ladigan imkoniyatlar beriladi.

*Uchinchi*dan, yangi tizimlar obyektga yo`naltirilgan texnologiyalarga asoslangan, uning uch aspektini alohida ko`rib chiqamiz. Dasturlash nuqtayi-nazaridan obyektga yo`nalgan texnologiya modulli dasturlar shunday tarzda yozilishni bildiradiki, unda modullardan, biron narsaga bog`liq bo`lmagan holda, qayta-qayta foydalanish imkoniyati ta`minlansin. Har bir funksiya bu tilda shunday tarzda yoziladiki, u mustaqil butunlik sifatida amal qila olsin. Obyektga yo`nalgan texnologiya tizim bilan foydalanuvchi orasidagi interfeysni ham aniqlaydi.

Foydalanuvchi interfeysning obyektga yo`nalganligi shu narsani bildiradiki, interfeysning har bir elementi vaziyatning o`zgarishiga va foydalanuvchi harakatiga mustaqil ravishda reaksiya qiladi. Bu foydalanuvchining tizim bilan ishlashni sezilarli darajada osonlashtiradi. Obyektga yo`nalgan texnologiyadan ma`lumotlarni samarali saqlash uchun ham foydalaniladi. CAD ning oddiy tizimlarida detal haqida ma`lumotlar odatda bir nechta fayllarda saqlanadi: bitta fayldan geometrik shakl uchun foydalaniladi, ikkinchidan – chet unsurlar tarmoqchasi uchun, uchinchidan – sonli-raqamli dasturli boshqariladigan stanok frezasi harakati trayektoriyasi uchun foydalaniladi. Obyektga yo`nalgan tizimlarda bitta detalga taalluqli hamma ma`lumotlar bor faylda saqlanadi. Bir xil ma`lumotlar turli fayllarda saqlanganida qo`shimcha takrorlanish sodir bo`ladi, obyektga yo`nalgan tizimlarda esa buning oldi olinadi, natijada xotiraning sezilarli qismi tanlanadi.

To`rtinchidan, tizimlar yoki parametrik, yoki variatsion modellashtirishni quvvatlaydi. Ikala yondashuv ham foydalanuvchiga cheklanishlarni berib, shakl alohida elementlarning xarakteristikalarini emas, balki shaklning o`zini aniqlash imkonini beradi. Bu yondashuvlar orasidagi farq – birinchi holda cheklanishlar birdaniga, ikkinchi holda esa – ketma-ket hisobga olinadi. To`g`ri to`rtburchakni, bir-biridan ma`lum masofada turgan, parallel kesmalarning ikki jamlanmasi sifatida aniqlash, shakli elementlari bilan bevosita ishlashga misol bo`ladi. Lekin o`sha to`g`ri to`rtburchak qo`shni kesmalar bir-biriga perpendikulyar bo`lishi va kesmalar orasidagi masofa berilishi bilan ham aniqlanishi mumkin. Parametrik yoki variatsion modellashtirish imkoniyatini quvvatlovchi ko`p tizimlar perpendikulyarlik va parallellik kabi cheklashlarni bevosita foydalanuvchining boshlang`ich eskizidan qabul qilishadi, bunda kiritilayotgan ma`lumotlar hajmi kamayadi. Bu holda foydalanuvchidan faqat o`lchamlar kiritilishi talab qilinadi, bundan keyin u bu o`lchamlarni o`zgartirib, shaklni o`zgartirish mumkin. Tizimning bunday funksiyasi *o`lchamlar bo`yicha modellashtirish (dimension-driven modeling)* deb ataladi. Bu shuni bildiradiki, geometriya detal o`lchamlari bilan aniqlanadi va bu o`lchamlarni qayta taqsimlab, geometriyani osonlik bilan

o`zgartirish mumkin. Bunday tizimlarda foydalanuvchi o`lchamlar orasidagi nisbatlarni ham ko`rsatish mumkin, ular o`lchamlarining o`zi o`zgarganda saqlanib qoladi. Bu – mahsulotda ishlab chiquvchi g`oyasini saqlab qolishga imkon beruvchi mexanizmlardan biridir. Lekin murakkab detal uchun geometriyani aqiqlovchi hamma cheklashlarni berish ancha murakkab ish. Bunday vaziyatlarda parametrik yoki variatsion modellashtirishni quvvatlovchi tizimlarga loyihalash uchun qo`shimcha ma`lumotlar talab qilinadi.

Va nihoyat, tizimlarga Internet orqali hamkorlikda loyihalashni quvvatlash kirib qo`shildi. Bu quvvatlash bir-biridan uzoqdagi foydalanuvchilarga bir detalning ustida ishlash imkonini beradi, bunda detalning modeli ularning ko`z oldidagi ekranlarda bo`ladi. Ishlab chiquvchilar o`zlarining detallarini boshqa ishlab chiquvchilar detallari bilan solishtirib, loyihani tekshirib chiqishlari mumkin.

Savol va topshiriqlar

1. Grafik qurilma komponentlarini sanab chiqing va ularning har birining vazifasini tushuntirib bering.
2. Tarmoqqa birikkan ishchi stansiyalardan tashkil topgan apparat konfiguratsiyasining afzalliklarini sanab chiqing.
3. Sizga ma`lum bo`lgan ikki o`lchamli chizmalarni chizish uchun dasturlarni sanab chiqing.
4. Sizga ma`lum bo`lgan geometrik modellashtirish tizimlarini sanab chiqing.
5. Sizga ma`lum bo`lgan CAM tizimlarini sanab chiqing. Har bir dasturaviy mahsulot funksiyalarini qisqacha bayon qiling.
6. Sizga ma`lum bo`lgan CAE tizimlarini sanab chiqing. Har bir dasturaviy mahsulot funksiyalarini qisqacha bayon qiling.

3 – BOB. GRAFIK DASTURLASHNING ASOSIY KONSEPSIYALARI

Shakllarni interaktiv o`zgartirish *CAD/CAM/CAE* tizimlari bilan ishlashning salmoqli qismini tashkil qiladi. Demak, monitor ekranida grafik tasvirni hosil qiluvchi dasturaviy ta`minot bunday tizimlarning eng ahamiyatli tarkibiy qismi bo`ladi. Shuning uchun bizga grafik tasvirlashning atamalari va asosiy tushunchalarini o`rganishga to`g`ri keladi.

3.1. Grafik kutubxonalar

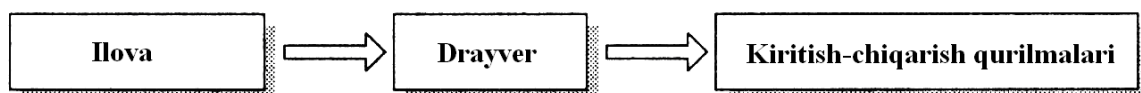
Kompyuterda dasturlash (*computer programming*) ilgari belgilangan grammatika qoidalariga mos ravishda kompyuter komandalari tilida «insho» yozishni bildirar edi. Kiritilayotgan raqamlarga javoban bajarilayotgan «insho» terminal yoki ma`lumotlar faylida kutilayotgan raqamlar va simvollar edi. Hozirgi paytda «insho»larning kirish va chiqarishda ko`pincha grafik informatsiya joylashadi. Bunday dasturlash *grafik (graphics programming)* deb, uning qo`llanish sohasi esa – *kompyuter grafikasi (computer graphics)* deb ataladi.

Grafik dasturlash oddiy dasturlash uchun zarur bo`lgan asosiy dasturaviy ta`minot (operatsion tizim, redaktor va kompilyator)dan tashqari maxsus grafik dasturlar ikki sinfga: qurilmalar drayveri va grafik kutubxonalarga bo`linadi.

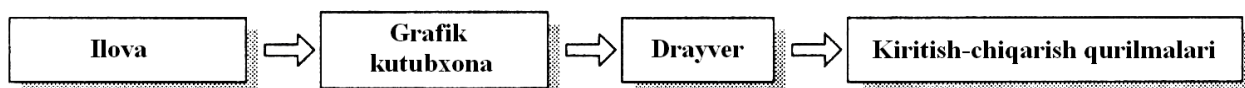
Drayver qurilmalari apparatga bog`liq kodlar jamlanmasi sifatida qaralishi mumkin; ular grafik qurilmaning protsessorini bevosita shunday boshqarishadiki, elektron nur to`plami zarur bo`lgan joyga drayverlar albatta apparatga bog`liq bo`ladi, ya`ni ular muayyan grafik protsessorga qattiq bog`langan bo`ladi. Assembler haqida hali shu gapni aytish mumkin, uning muayyan turi protsessorlarning aynan bir modelida bajarilishi mumkin. Grafik dastur drayver qurilmasidan bevosita foydalanganda ham shunday bo`ladi (3.1-rasm). Bunday grafik dasturni, boshqa grafik qurilmaga o`tilganda, drayverning yangi komandalardan foydalanib, qaytadan yozib chiqishga to`g`ri keladi. Bundan tashqari, drayver qurilmasining komandalari ancha primitiv, shuning uchun bunday

drayver qandaydir murakkab masalani yechadigan bo`lsa, u juda uzun bo`ladi, bunday dastur yomon o`qiladi.

Dasturchilar dasturlarni yuqori darajadagi tillarda yozishni afzal ko`rishadi. Grafik dasturlash bundan istisno emas, ayniqsa past darajadagi drayver komandalari bilan bog`liq qurilmalar bilan birga kutubxonalar yetkazib berila boshlandi, ular *grafik kutubxonalar (graphics libraries)* nomini oldi. Grafik kutubxona, masalan matematikka o`xshab, muayyan masalalarni yechish uchun mo`ljallangan nimdasturlar majmuidan iborat iborat bo`ladi. Muayyan nimdastur ekranda to`g`ri chiqib, doira yoki boshqa obyektни tasvirlash mumkin. Grafik kutubxona drayver qurilmasi komandalariga asoslanadi (3.2-rasm). Har bir nimdastur drayver quvvatlaydigan komandalar majmuidan foydalanib yaratiladi. Masalan, doirani tasvirlovchi nimdastur, ekranda nuqta yoki kesmalarni chizuvchi drayverning alohida komandalardan tuzilishi mumkin.



3.1-rasm. Qurilma drayveridan bevosita foydalanish



3.2-rasm. Grafik kutubxonadan foydalanish

Grafik kutubxona nimdasturlaridan, matematik nimdasturlardan foydalangan kabi, foydalanish mumkin. Zarur bo`lgan nimdastur asosiy dasturdan chaqirib olinadi, bu dasturchiga, ularning qiymatlarini hisoblash talab qilinganida, sinus va kosinus funksiyalarini chaqirib olishga o`xshaydi. Grafik kutubxona nimdasturlaridan foydalanishdagi muammolardan biri shundaki, ularning nom iva chaqirish usullari (kirish va chiqish argumentlari) har bir kutubxonada o`ziniki. Agar bir grafik kutubxona hamma mavjud qurilmalar bilan ishlay olsa, (agar hamma mavjud qurilmalar drayverlari bu kutubxonani quvvatlasa, bu nazariy nuqtayi-nazardan mumkin bo`lar edi), qiyinchiliklar vujudga kelmas edi. Lekin ba`zi sabablarga ko`ra dasturaviy ta`minotni ishlab chiquvchilar hamma drayverlar

bilan ishlay oladigan grafik kutubxonani yaratishni istashmayapti yoki eplay olishmayapti, shu sababli har bir kutubxonaning uni quvvatlaydigan drayverlar doirasi mavjud. Demak, bunday kutubxona grafik qurilmalarning faqat cheklangan majmui bilan ishlay oladi, ko`p qurilmalar bilan ishlashga mo`ljallangan grafik dasturlarni esa bir necha kutubxonalardan foydalanilgan holda qaytadan yozib chiqishga to`g`ri keladi.

3.2. Koordinatalar tizimlari

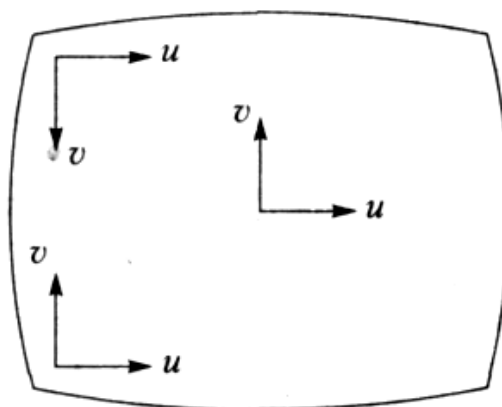
Obyekt tasvirini grafik qurilmaning ekraniga chiqarish uchun ikkita asosiy masalani:

- Obyekt hamma nuqtalarning fazadagi holatini ko`rsatish;
- Ular namunalarining holatini monitorda aniqlash kerak.

Nuqtalarning fazodagi va monitordagi holatlarini berish uchun koordinatalar tizimlaridan foydalaniladi. Koordinatalarning har xil tizimlari o`zaro qanday bog`langanligini tushunish kera. Bu ayniqsa uch o`lchamli obyektни tekis ekranda loyihalash uchun ahamiyatlidir. Real obyekt inson ko`zining to`riga qanday qonuniyat bilan proyeksiyalansa, ekrandagi proyeksiya ham shunday qonuniyat bo`yicha quriladi.

Koordinatalar tizimlari orasida birinchi navbatda biz *qurilma koordinatalar tizimi (device coordinate system)*ni qurib chiqamiz, u nuqtaning ekrandagi holatini belgilaydi. Bu tizim gorizontaal o`q x va vertikal o`q y lardan tarkib topgan (3.3-rasm). E`tibor bering, hisob boshlanishi ixtiyoriy tanlanishi mumkin. Ekran istalgan nuqtasining holatini berish uchun x va y o`qlari kifoya qiladi, shuning uchun bu ikki o`qqa perpendikulyar bo`lgan uchinchi o`q aniqlanmaydi. Istalgan nuqtaning holati ikkita butun sonlar x va y bilan beriladi, ularning qiymatlari koordinata boshi va x va y o`qlari bo`ylab nuqta orasidagi piksellar soniga teng bo`ladi. Lekin bir nuqtaning o`zi koordinata boshining holati, o`qlari yo`nalishi va masshtabga qarab har xil x va y qiymatlari bilan berilishi mumkin. Bu parametrlar har xil grafik qurilmalar uchun yetarli darajada erkin o`rnatiladi (3.3-rasmga

qarang). Shu sababli grafik qurilma almashtirilganda grafik dasturdagi apparat koordinatalarining o`zgartirilishi talab qilib qolinishi mumkin.



3.3-rasm. Qurilmaning koordinata tizimlari

Qurilmaning virtual koordinatalar tizimi (virtual device coordinate system) yuqorida bayon qilingan muammoni bartaraf qilish imkonini beradi. Qurilmaning virtual koordinatalar tizimi hamma ishchi stansiyalar uchun o`qlar hisobining boshini, yo`nalashini va masshtabini fiksatsiya qiladi. «Virtual» so`zi shuni bildiradiki, ushbu hisoblash tizimi faqat dasturchining tasavvurida mavjud bo`ladi. Odatda hisob boshi ekranning chapdagi quyi burchagida joylashadi, x o`qi o`ngga, y o`qi esa – yuqoriga yo`naladi. Ikkala koordinata ham noldan birgacha bo`lgan diapazonda o`zgarishi mumkin. Holati virtual koordinata tizimida berilgan nuqta istalgan ekranda faqat bitta joyga tushadi. Bu dasturchiga, qurilmalarning muayyan koordinata tizimlari haqida boshni qotirmasdan, shakllarni bir xil aniqlash imkonini beradi. Grafik dastur virtual koordinatalarni qurilma drayverining nimdasturiga uzatadi, u virtual koordinatalarni muayyan koordinatalarga aylantiradi.

Qurilmaning virtual va oddiy koordinatalar tizimlari ekranda nuqtaning holatini berish imkoniyatini beradi. Endi uch o`lchamli fazo bilan ishlash uchun koordinatalar tizimlarini ko`rib chiqamiz.

Asosiy uch o`lchamli koordinatalar tizimi atigi uchta:

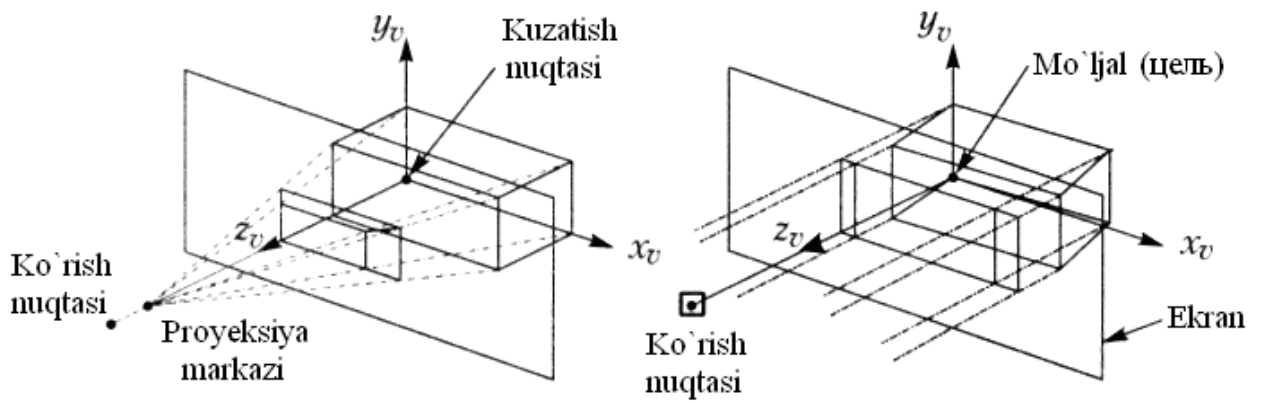
- *tashqi koordinatalar tizimi (world coordinate system);*

- *model koordinatalar tizimi (model coordinate system)*;
- *kuzatuvchi koordinatalar tizimi (viewing coordinate system)*.

Tashqi yoki dunyo koordinatalar tizimi (world coordinate system) – bu bizni qiziqtirayotgan dunyoni bayon qilish uchun foydalaniladigan tayanch tizimdir. U bu dunyo obyektlariga nisbatan tashqi bo`ladi. Masalan, bizni qiziqtirayotgan dunyo sinf bo`lsa, bunday tizimdan partalar, stullar va doskaning joylanishlarini bayon qilish uchun foydalanish mumkin.

Keyingi qadam – dunyo (sinf)dagi har bir obyektning shaklini bayon qilish bo`ladi. Obyekt shakli obyekt bilan bog`langan koordinatalar tizimiga – *modellar koordinatalar tizimi (model coordinate system)* nisbatan obyektning barcha yoki bir nechta xarakterli nuqtalarining koordinatalari bilan aniqlanadi. Bunday tarzda aniqlangan obyekt nuqtalarining koordinatalari obyekt fazoda siljiganida yoki aylanganida ham o`zgarmaydi. Ular faqat obyekt shakliga bog`liq bo`ladi. Model koordinatalar tizimi qaysi obyektga bog`langan bo`lsa, o`sha obyekt bilan birga siljiydi. Shuning uchun har bir obyektning shakli o`sha obyektning shaxsiy model koordinatalar tizimida aniqlanadi. Istalgan obyektning joylashishi va oriyentatsiyasi ushbu obyekt model koordinatalar tizimining tashqi koordinatalar tizimiga nisbatan joylashishi va oriyentatsiya bilan beriladi. Koordinatalar tizimlarining bir-biriga nisbatan joylashishi va oriyentatsiyasi o`zgartishlar matritsasi bilan aniqlanadi. Tashqi koordinatalar tizimi va hamma obyektlar uchun model tizimlarining mavjudligi dunyoni, ya`ni ushbu dunyodagi hamma obyektlarning joylashishi va shaklini aniqlaydi. Boshqacha aytganda, o`zgartishlar matritsasini qo`llash tashqi tizimda istalgan obyekt istalgan nuqtasining koordinatalarini olish imkonini beradi.

Keyingi qadam – uch o`lchamli obyektlar yoki ularning nuqtalarini agar ular inson ko`zi to`riga qanday proyeksiyalansa, monitorga shunday proyeksiyalashdir. Kompyuter grafikasida ikki xil: perspektiv va parallel proyeksiyalardan foydalaniladi. (3.4-rasm).



3.4-rasm. Proeksiyalashning ikki turi: *a* – perspektiv; *b* – parallel

Proeksiyalashning ikki turi ham ikki nuqta: ko`rish nuqtasi va kuzatish nuqtasining berilishini talab qiladi. *Ko`rish nuqtasi* (*viewpoint*) – bu kuzatuvchining ko`zi. *Kuzatish nuqtasi* (*view site*) – bu «ko`rish nuri» yo`nalishini aniqlovchi obyekt nuqtasi. Ko`rish nuqtasidan mo`ljal (цель)ga o`tkazilgan vektor kuzatuv yo`nalishini belgilaydi.

Perspektiv proyeksiya (*perspective projection*)da ko`rilayotgan obyektning hamma nuqtalari proyeksiya markazi bilan tutashtiriladi, u odatda ko`rish nuqtasi va mo`ljal (цель)ni tutashtiruvchi chiziqda yotadi¹. Ushbu chiziqlarning ekran bilan kesishish nuqtalari proyeksiyani hosil qiladi. Ekran ko`rish nuqtasi va mo`ljal (цель) orasida joylashadi. *Parallel proyeksiya* (*parallel projection*)da obyektning hamma nuqtalaridan chiziqlar kuzatuvchi yo`nalishida kuzatish yo`nalishiga parallel o`tkaziladi, bu chiziqlarning ekran bilan kesishish nuqtalari esa proyeksiyani shakllantiradi. Ekran, perspektiv proyeksiyalashdagi kabi, proyeksiyalash yo`nalishiga perpendikulyar joylashadi. Bunday proyeksiya ortogonal deyiladi².

Agar proyeksiyalanayotgan obyekt nuqtalarning koordinatalari x_v y_v z_v koordinatalar tizimida berilgan bo`lsa (3.4-rasm) yuqorida bayon qilingan metodlarning istalgan biridan olinadigan proyeksiyaning nuqtalari osonlik bilan hisoblanishi mumkin. Masalan, obyekt parallel proyeksiyasi nuqtalarining koordinatalari obyekt nuqtalarining mos X_v va Y_v koordinatalariga teng bo`ladi. x_v y_v z_v koordinatalar tizimi *kuzatuvchi koordinatalar tizimi* (*model coordinate system*) deb ataladi, chunki u kuzatish proyeksiyasini hisoblashni yengillashtiradi.

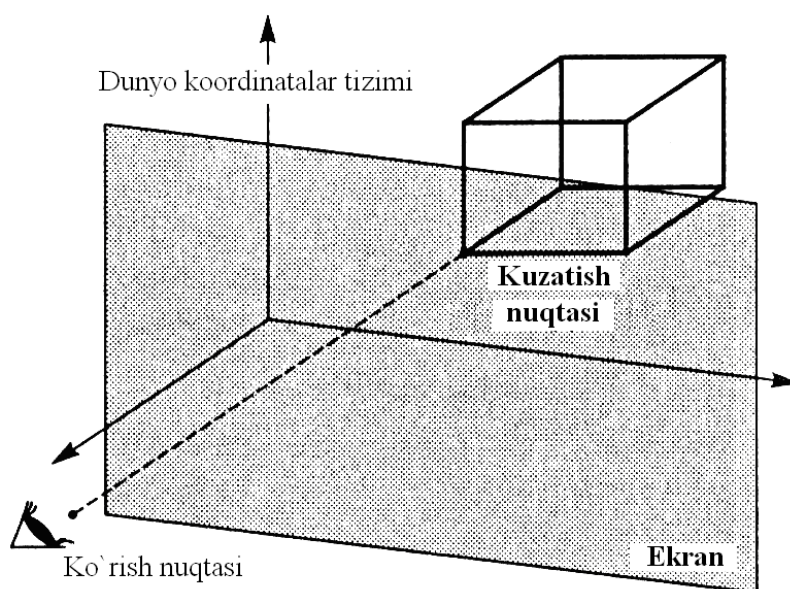
Kuzatuvchi koordinatalar tizimi qurilayotgan quyidagi xarakteristikalarga ega bo`lishi ta`minlanishi shart. Bu koordinatalar tizimining boshlanishi kuzatilayotgan nuqtada joylashadi z_v o`qi koordinata boshidan ko`rish nuqtasiga yo`naladi, y_v o`qi esa ekranning vertikal o`qiga parallel bo`ladi (3.4-rasmga qarang). Uchinchi o`q – x_v o`qi dastlabki ikki o`qlarning vektorli ko`paytmasi sifatida aniqlanadi.

¹ Aks holda proyeksiya egriburchakli deyiladi.

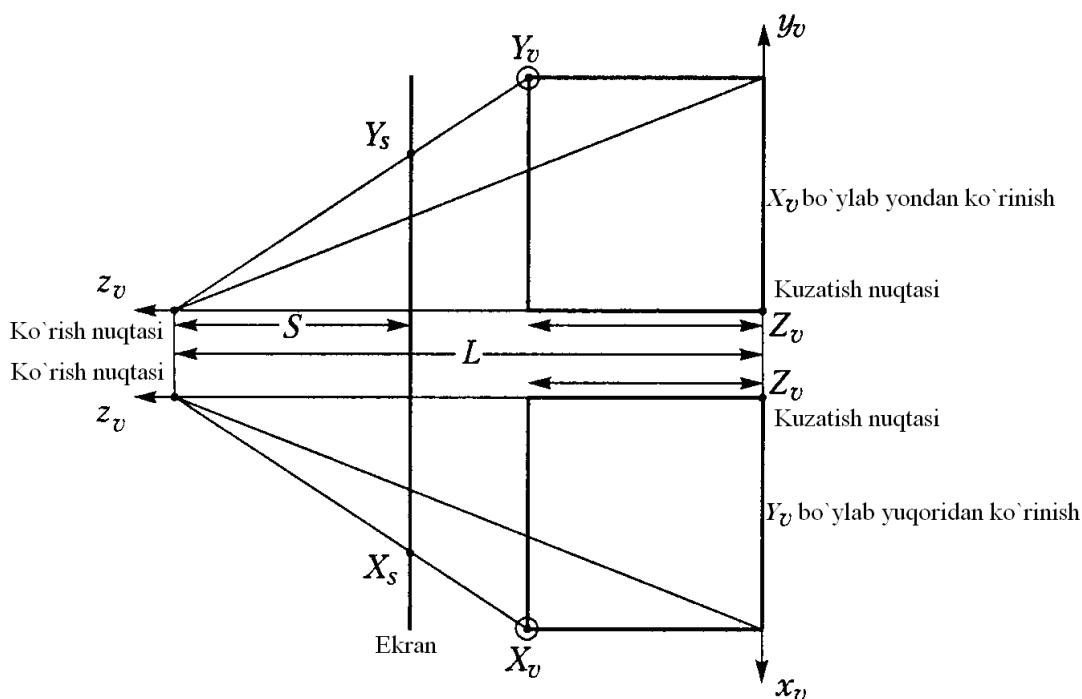
² Egriburchakli proyeksiyada ekran oriyentatsiyasi ixtiyoriy bo`ladi.

Ko`rsatilgan, uning koordinatalarini biz $X_v Y_v Z_v$ bilan belgilaymiz.

Ko`pchilik odamlarda fazodagi vertikal yo`nalish tabiiy tarzda ekrandagi vertikal yo`nalish tabiiy tarzda ekrandagi vertikal yo`nalish bilan assotsiyalanadi, shu sababli y_v o`qi vertikal vektorning fazadagi ekranga proyeksiyasi hisoblanadi. Grafik kutubxonalarining ko`pida foydalanuvchiga fazadagi *vertikal vektori* (*up vector*)ni dunyo koordinatalarida berishga to`g`ri keladi. Ko`rish nuqtasi va kuzatish nuqtasi holatlari ham dunyo koordinatalarida beriladi (3.5-rasm).



3.5-rasm. Ko`rish nuqtasi va kuzatish nuqtasi



3.6-rasm. Proeksiyani hisoblash

Kuzatuvchi koordinatalar tizimi aniqlangandan va bizni qiziqtirayotgan hamma obyektlar koordinatalari hisoblangandan so'ng faqat ularning proyeksiyalarining ekrandagi holatini hisoblab topish qoladi xolos. Parallel proyeksiya uchun buni bajarish juda osonligini biz bilamiz. Shu sababli biz proyektiv proyeksiyada koordinatalarni hisoblash protsedurasining bayoni bilan shug'ullanamiz. Yuqoridan va yondan ko'rinishlarini ko'rib chiqamiz (3.4,a va 3.6-rasmlar). Bizni qiziqtirayotgan nuqta rasmda ajratib ko'rsatilgan, uning koordinatalarini biz X_v U_v Z_v bilan belgilaymiz.

Uchburchaklar o'xshashligidan

$$X_s = \frac{S}{L - Z_v} X_v; \quad (3.1)$$

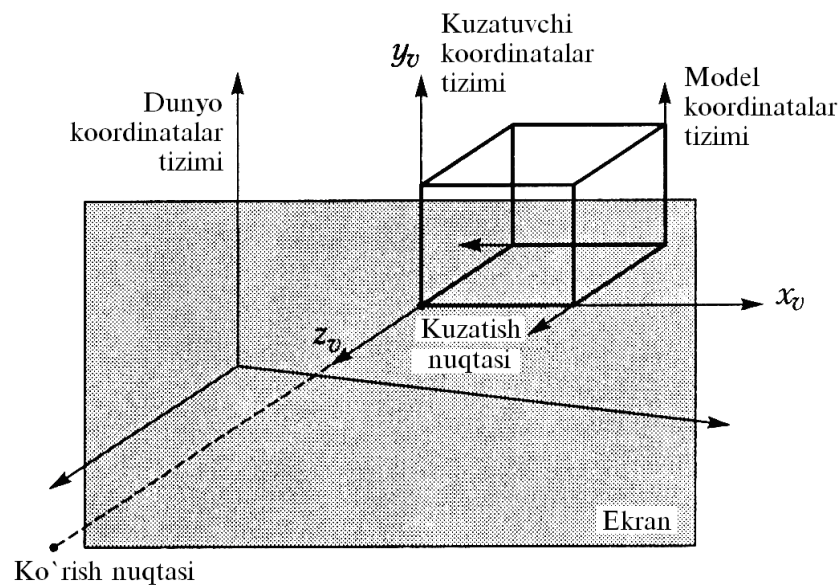
$$Y_s = \frac{S}{L - Z_v} Y_v. \quad (3.2)$$

(3.1) va (3.2) formulalarda X_s va Y_s – tanlangan nuqta proyeksiyasigacha bo'lgan masofalar. Masofalar z_v o'qi ekran bilan kesishadigan nuqtadan gorizont va vertikal yo'nalishlarda o'lchanadi. L – kuzatish nuqtasi va proyeksiya markazi orasidagi masofa, S – esa proyeksiya markazi va ekran orasidagi masofa. (3.1) va

(3.2) formulalar shuni ko`rsatadiki Z_v qiymati katta bo`lgan nuqtaning X_s va Y_s , qiymatlari katta bo`ladi, shu sababli uzoqdagi kesma, yaqindagi shu uzunlikdagi kesmaga nisbatan, kichikroq bo`lib tuyuladi. X_s va Y_s masofalar tasvir markazining istalayotgan holatini va uning monitordagi o`lchamlarini hisobga olgan holda qurilmaning virtual koordinatalariga aylantiriladi.

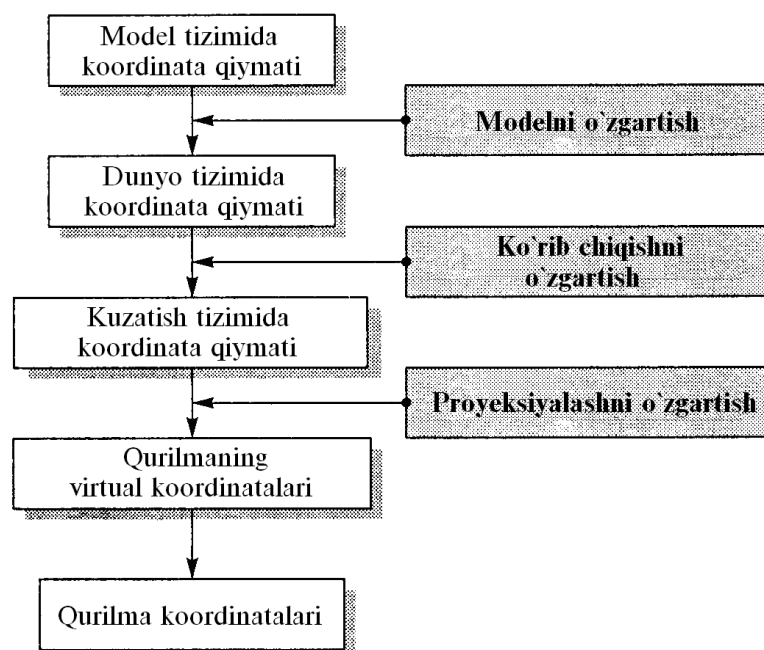
Yuqorida sanab o`tilgan koordinatalar tizimlari orasidagi nisbat 3.7-rasmda illyustratsiya qilingan.

Yuqorida qayd qilinganidek koordinatalar tizimlari bir-biri bilan o`zgartish matritsalarini orqali bog`langan. Har bir model koordinatalar tizimining dunyoviyga nisbatan holati va oriyentatsiyasi mos o`zgartishlar matritsasi bilan beriladi. Kuzatuvchi koordinata tizimi ham dunyoviyga nisbatan o`zgartishlar matritsasi yordamida agiqlanishi mumkin, buning uchun ko`rish va kuzatish nuqtalarining holati hamda vertikal vektori berilishi kerak. O`zgartishlar matritsasiidan foydalanib nuqta proyeksiyalarini hisoblash protsedurasi quyidagicha bo`ladi. Dastlab proyeksiyalanayotgan nuqtaning koordinatalari o`zgartishlar matritsasi yordamida modellan dunyoviyga o`tkaziladi, bu nuqtaning o`zi taalluqli bo`lgan modeldan dunyo koordinata tizimiga o`tganligini bildiradi. Bu operatsiya *modelni o`zgartirish (model transformation)* deyiladi. (3.8-rasm). So`ngra bu nuqtaning koordinatalari dunyo koordinata tizimidan kuzatuvchiga o`zgartiriladi. Bu operatsiya (3.8-rasmga qarang). *Kuzatishlarni o`zgartirish (viewing transformation)* deb ataladi. Va nihoyat, kuzatuvchi koordinatalar tizimidagi koordinatlar (3.1) va (3.2) formulalar bo`yicha X_s va Y_s qiymatlariga o`zgartiriladi, so`ngra qurilmaning virtual koordinatalariga aylantiriladi. Bu operatsiya (3.8-rasm) *proyeksiyani o`zgartirish (projection transformation)* deyiladi. Oxirida qurilmaning virtual koordinatalarini drayver nimdasturi oddiyga o`zgartiriladi. Natija 3.8-rasmda ko`rsatilgan.



3.7-rasm. Koordinatalar tizimi nisbati

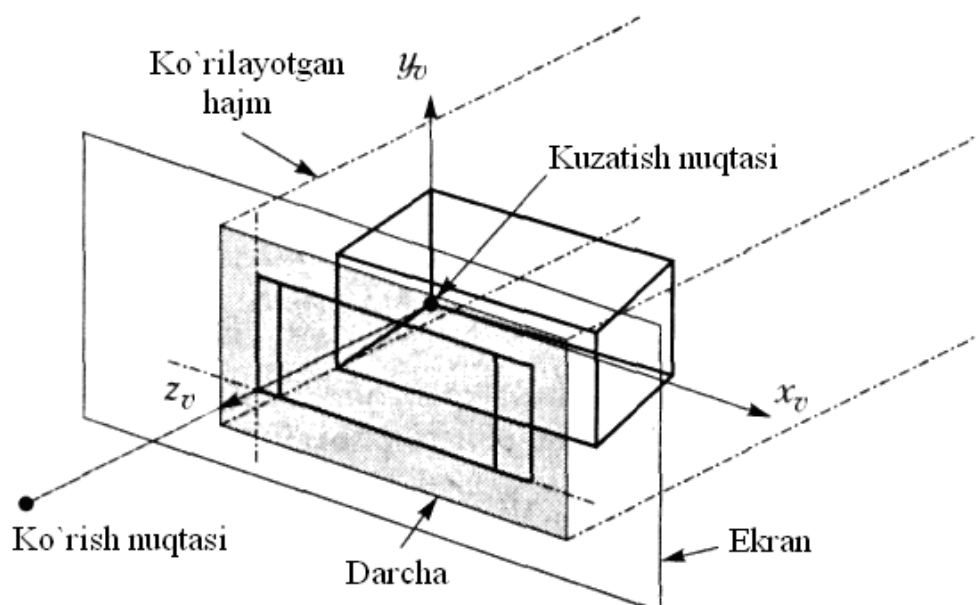
Bu o'zgartishlarning barchasi odatda grafik kutubxonaning ichida bajariladi, dasturchiga esa o'zgartishlarni o'tkazish uchun zarur bo'lgan ma'lumotlarni ko'rsatish qoladi xolos. Masalan, obyektlar translyatsiyasi va burilish modulni o'zgartishda hisobga olinadi, ko'rish nuqtasi, kuzatish nuqtasi va vertikal vektor holati – kuzatishni o'zgartirishda, proyeksiya turi, proyeksiya markazining ekranda joylashishi – proyeksiyalashni o'zgartirishda nisbatga olinadi. Lekin primitiv darajadagi grafik kutubxonalar dasturidan ushbu hamma o'zgartishlar uchun kodni mustaqil yozishni talab qilishi mumkin.



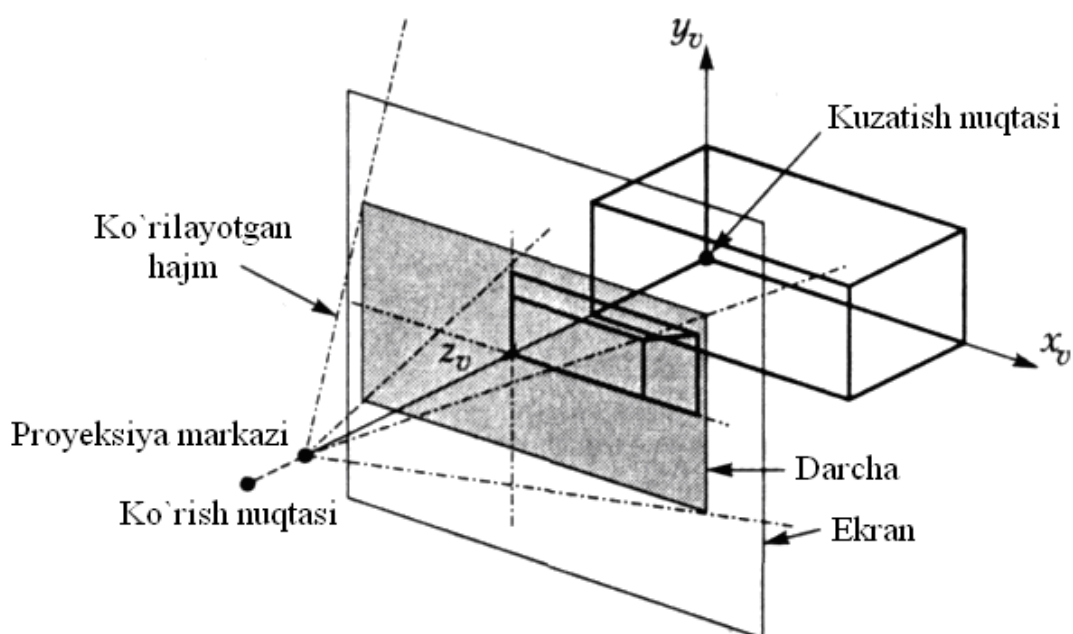
3.3. Darcha va ko`rsatish ekrani

Darcha (okno, window) atamasi tarmoq kompyuter muhitida ishchi stansiya monitori ekranining shunday qismini bildiradiki, u orqali foydalanuvchi ushbu tarmoqqa ulangan hisoblash resurslari bilan o`zaro ta`sirda bo`ladi. Kompyuter grafikasida bu atama boshqa ma`noga ega. Darcha – bu monitorga proyeksiyalanayotgan fazoning bir qismidir. Darchadan tashqarida joylashgan obyektlar monitorda paydo bo`lmaydi. Darcha odatda quyidagicha ta`riflanadi: u ekrandagi to`g`ri to`rtburchak bo`lib, unda X_v , Y_v qiymatlari ko`rib chiqish koordinatalar tizimida berilgan bo`ladi (3.9- va 3.10-rasmlar). Fazoning ko`rinadigan jabhasi *ko`rinadigan obyekt (viewing volume)* deyiladi va u proyeksiya turiga bog`liq. Parallel proyeksiya uchun bu jabha parallelopiped, perspektiv proyeksiya uchun esa – piramida shakliga ega bo`ladi.

Ko`rilayotgan hajm loyihalashda ancha murakkab tasvirni berishi mumkin, chunki unga kuzatuvchiga yaqin yoki undagi ancha uzoq joylashgan keraksiz obyektlar tushishi mumkin. Ba`zan bu hajmni ikki tekislik bilan cheklash qulay bo`ladi, ulardan biri yaqinda, ikkinchisi esa – uzoqda joylashadi (3.11-rasm). Perspektiv proyeksiya uchun yaqindagi va uzoqdagi tekisliklar qanday aniqlansa, parallel proyeksiya uchun ham shunday agiqlanadi.

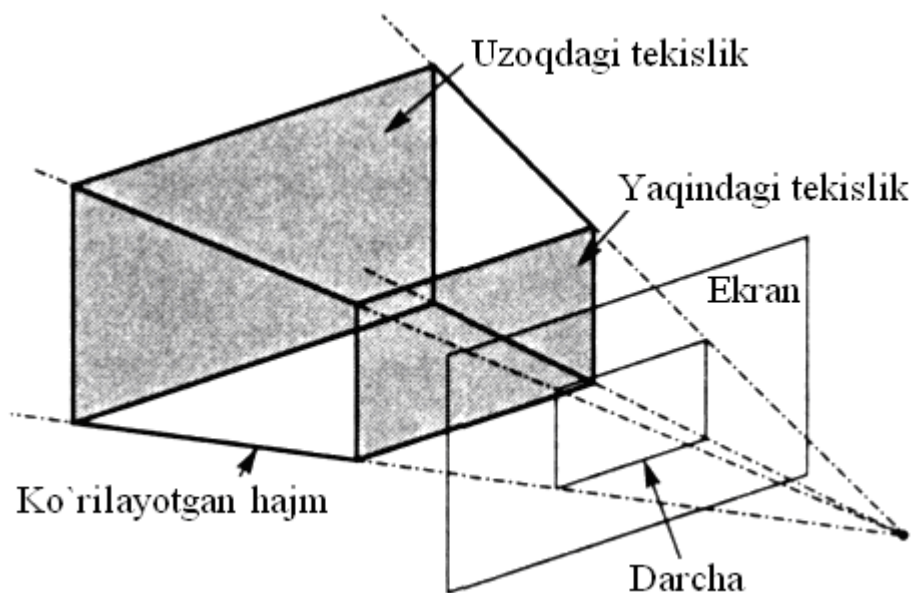


3.9-rasm. Parallel proyeksiya uchun darcha va ko`rilayotgan hajm

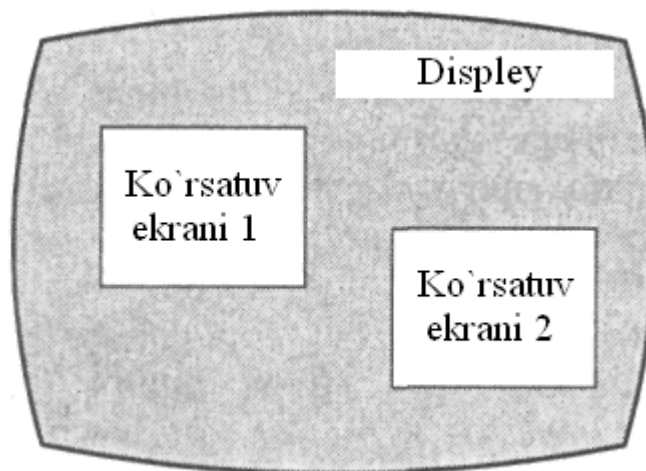


3.10-rasm. Perspektiv proeksiya uchun darcha va ko`rilayotgan hajm

Ko`rsatish ekrani (view port) – bu proyeksiyalanayotgan tasvir tasvirlanadigan ekranning bo`lagi (3.12-rasm). Bu bo`lakka ko`riladigan hajm proyeksiyalanadi, u «oddiy» darcha deyiladi. Tasvir translyatsiya va masshtablashdan tarkib toradi, ular ko`rsatuv ekrani markazi va monitor markazi orasidagi masofani hamda darcha va ko`rsatuv ekrani o`lchamlari orasidagi farqni hisobga oladi. Boshqacha aytganda, (3.1) va (3.2) formulalar bo`yicha olingan X_s va Y_s , qiymatlari shunday kattalashtirilishi yoki kichiklashtirilishi kerakki, natijada darcha markazi monitor markaziga emas, balki ko`rsatuv ekrani markaziga joylashsin. Bundan tashqari, ular darchaning to`rt chegaraviy nuqtasi ko`rsatuv ekranining to`rt chegaraviy nuqtalari bo`lib qolishi uchun, masshtablanishi kerak. Darcha tomonlarining nisbati ko`rsatuv ekrani tomonlari nisbatidek bo`lishi kerak, aks holda tasvir buziladi va masalan, doira ellipsga aylanib qoladi.



3.11-rasm. Yaqindagi va uzoqdagi tekisliklar



3.12-rasm. Ko`rsatuv ekranlari

Quyida OpenGL grafik kutubxonadan foydalanib darcha va ko`rsatuv ekranini hosil qiluvchi kod misoli keltirilgan. OpenGL da ko`rish darchasi ham uch o`lchovli hajm kabi aniqlanadi. Lekin natijada shunday bo`ladiki, go`yo ko`rish darchasining uchinchi o`lchovi hisobga olinmagandek.


```

static GLint viewport[] = {0, 0, 400, 400};
/* Видовой экран задается координатами окна1. Первый и второй аргументы
определяют положение левого нижнего угла окна просмотра, а третий
и четвертый – размер прямоугольника. */
static GLclampd depth_range[] = {0.0, 1.0};
/* Первый и второй аргументы представляют собой поправки к минимальному
и максимальному значениям, которые могут храниться в буфере глубины. */
static GLdouble viewing_volume[] = {-100.0, 100.0, -100.0, 100.0, -10.0, 100.0};
/* Горизонтальный и вертикальный размеры окна задаются в координатах
просмотра (аргументы 1-4). Пятый и шестой аргументы определяют
расстояние от экрана до ближней и дальней плоскостей соответственно. */
glOrtho(viewing_volume[0], viewing_volume[1], viewing_volume[2], viewing_volume[3],
viewing_volume[4], viewing_volume[5]);
/* Определяет тип проекции - параллельный - и создает матрицу
ортографического параллельного просматриваемого объема, после чего
умножает на него текущую матрицу. */
glViewport(viewport[0], viewport[1], viewport[2], viewport[3]);
/* Определяет прямоугольник окна, открытого диспетчером, после чего
отображает в это окно готовое изображение. Если glViewport не
используется, окно просмотра по умолчанию считается совпадающим со всем
открытым окном. */
glDepthRange(depth_range[0], depth_range[1]);
/* Определяет кодирование z-координат при преобразовании просмотра.
Значения z-координаты масштабируются этой командой так, чтобы
они лежали в определенном диапазоне. */

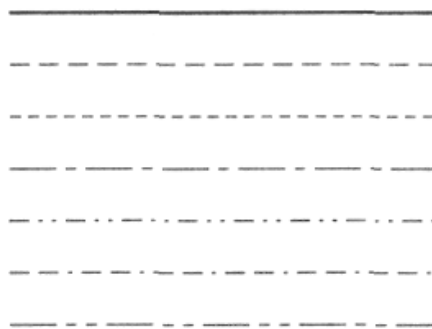
```

3.4. Primitivlar

Primitivlar (primitives) – bu grafikaning elementlari, ular grafik kutubxona tomonidan aks ettirilishi mumkin. Har bir kutubxonada o'zining primitivlari majmui mavjud, shuning uchun biz grafik kutubxonalarning ko'pchiligi quvvatlaydigan, faqat eng umumiy primitivlarnigina ko'rib chiqamiz.

3.4.1. Kesma

To'g'ri chiziq (линия – line) kesmasini aks ettirish uchun uning ikkala oxirining koordinatalari berilishi zarur. Grafik kutubxonalarning ko'pchiligida oxirgi nuqtalarning koordinatalari uch o'lchamli fazoda berilishi mumkin; ekran tekisligiga proyeksiyalash avtomatik tarzda amalga oshiriladi. Kesma atributlari: turi, qalinligi, rangi va boshqalar. Ko'pchilik grafik kutubxonalar quvvatlaydigan kesmalarning turlari 3.13-rasmda tasvirlangan. Chizmalarni avtomatlashtirilgan ravishda ishlab chiqish tizimlari uchun bu turdagi chiziqlarni quvvatlash albatta zarur, chunki ular mashinasozlik va arxitektura chizmalarida va elektr sxemalarida ishlatiladi.



3.13-rasm. Kesmalarning har xil turlari

GKS, PHIGS va OpenGL kutubxonalarida asosiy bazaviy funksiyalardan biri singan chiziq (*polyline*) bo`ladi, u bir-biri bilan tutashgan kesmalar majmuidan tarkib topadi. Singan chiziqni hosil qiluvchi kesmalar oxirlarining koordinatalarimatritsa ko`rinishida beriladi. Faqat bitta kesmadan tarkib topgan singan chiziq bo`lganda, matritsaga uning ikki oxirining koordinatalari joylashadi.

$$[P] = \begin{bmatrix} X_1 & Y_1 & Z_1 \\ X_2 & Y_2 & Z_2 \\ \vdots & \vdots & \vdots \\ X_n & Y_n & Z_n \end{bmatrix}.$$

Quyida PHIGS va OpenGL kutubxonalarida singan chiziqni qurish funksiyalaridan foydalanish misollari keltirilgan.

```

PHIGS
Pint      num_of_points = 10;
          /* Количество точек в ломаной. */
Ppoint3   point3[] = {
            {0.0, 0.0, 0.0},
            {10.0, 20.0, 15.0},.....
            {1.0, 3.0, 6.5}};
          /* Координаты концов отрезков ломаной. */
Ppolyline3(num_of_points, point3);
          /* Рисует заданную ломаную. */

OpenGL
GLdouble  point[][3] = {
            {0.0, 0.0, 0.0},
            {10.0, 20.0, 15.0},.....
            {1.0, 3.0, 6.5}};

          /* Координаты концов отрезков ломаной. */
glBegin(GL_LINE_LOOP);
  . glVertex3dv(&point[0][0]);
  . glVertex3dv(&point[1][0]);

```

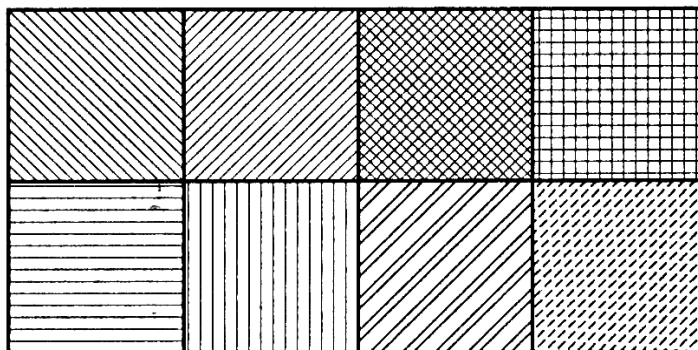
```

glVertex3dv(&point[9][0]);
glEnd();
/* Построение ломаной по заданным точкам (10 шт.) */

```

3.4.2. Ko`pburchak

Ko`pburchak – bu singan chiziqning o`zi, farqi bitta – cho`qqilar matritsalarining birinchi va oxirga qatorlari $[R]$ bir xil bo`lishi kerak (ularga mos nuqtalar ustma-ust tushadi). Singan chiziqni qurish funksiyasidan foydalanib, o`sha natijaga erishish mumkin, lekin maxsus funksiya yordamida qurilgan ko`pburchakni tizim ichki va tashqi qismlarga ega bo`lgan obyekt sifatida taniydi. Ko`pburchakning ichki yuzasi har xil turdagi shtrixovkalar bilan to`ldirilishi mumkin (3.14-rasm).



3.14-rasm. Turli shtrixovkalar bilan to`ldirishga misollar

Ichki qismining rangi (rangi, shtrixlash turi) hamda bu jabhani chegaralovchi singan chiziqning turi, qalinligi va rangi ko`pburchak atributi bo`lishi mumkin. Doira va to`g`ri burchakli to`rtburchaklarni qurish uchun ko`pburchaklarni qurish funksiyasidan foydalanish mumkin, lekin grafik kutubxonalarning ko`rchiligida ularni qurish uchun maxsus funksiyalar mavjud, ular ancha kam kiruvchi parametrlarni talab qiladi (masalan, doira markazi va radiusi yoki to`g`ri burchakli to`rtburchak diagonalining oxirlari). Bunga qaramasdan kutubxona ichida bu funksiyalar ko`pburchaklarni qurish funksiyasi orqali realizatsiya qilingan.

3.4.3. Marker

Markerlar odatda grafiklarda nuqtalarni ajratib ko`rsatish uchun qo`llanadi. Ko`pchilik grafik kutubxonalarda mavjud bo`lgan markerlar 3.15-rasmda ko`rsatilgan.

Marker turi atribut sifatida ko`rsatiladi. Polimarker, kesma kabi GKS va PHIGSlarda standart obyektidir. OpenGL markeri ochiqdan-ochiq quvvatlamaydi, lekin uni rastrli fayllarda va ularni ekranga chiqarishda saqlash mexanizmini taklif etadi. Shu sababli OpenGLda qurilgan grafik dastur platformaga ancha yaxshi ko`chadi. Quyida keltirilgan kod namunasi PHIGS va OpenGLda * (yulduzcha) markerini chiqarishni namoyish qiladi.



3.15-rasm. Markerlar misoli

```

PHIGS
pint          num_of_point = 10:
/* Количество маркеров для отображения. */
Ppoint3      point3[] = {
                {0.0, 0.0, 0.0},
                {10.0, 20.0, 15.0}, ... .
                {1.0, 3.0, 6.5}};
/* Координаты маркеров. */
pset_marker_type(PMK_STAR);
/* Тип маркера - звездочка. */
ppolymarker3(num_of_point, point3);
/* Построение маркеров по имеющимся данным. */

OpenGL
GLubyte asterisk[13] = {0x00, 0x00, 0x30, 0x18, 0x0c, 0x0c, 0x0c, 0x0c, 0x0c, 0x0c,
0x0c, 0x18, 0x30};1
/* Задание растрового изображения звездочки. Первой рисуется нижняя
строка, затем следующая и т. д. */
GLsizei      width = 8;
/* Ширина раstra в пикселах. */
GLsizei      height = 13;
/* Высота раstra в пикселах. */
GLfloat      origin_x = 0.0, origin_y = 0.0;
/* Задание начала отсчета системы координат раstra. Положительные
значения смещают начало отсчета вверх и вправо, отрицательные - вниз
и влево. */
GLfloat      incre_x = 10.0, incre_y = 0.0;
/* Поправки к положению раstra после вывода. */
GLfloat      white[3] = {1.0, 1.0, 1.0};
/* Задание цвета маркера. */
GLfloat      position[2] = {20.5, 20.5};
/* Координаты маркеров. */
glPixelStorei(GL_UNPACK_ALIGNMENT, 1);
/* Установка режима хранения пикселей, влияющего
на работу glBitmap. */
glColor3fv(white);
/* Установка белого цвета маркера. */
glRasterPos2fv(position);
/* Установка текущего положения раstra. */
glBitmap(width, height, origin_x, origin_y, incre_x, incre_y, asterisk);
/* Выводит растровую картинку, заданную в последнем аргументе,
представляющем собой указатель на эту картинку. Положение и размер
раstra указываются первыми четырьмя аргументами. */

```

3.4.4. Matn

Grafik kutubxonalarning ko`pchilik ikki turdagi matnni:

- tushuntirishlar uchun matnni (ekranli yoki ikki o`lchamli matn);
- uch o`lchamli matnni quvvatlaydi.

Tushuntirishlar uchun matn doim ekran tekisligida joylashadi, shu sababli u qanday burchakka burilgan bo`lmasin, uning shakli o`zgarmaydi. Uch o`lchamli matn uch o`lchamli fazodagi istalgan tekislikda joylashishi mumkin. Uning holati va oriyentatsiyasi dunyo koordinatalarida beriladi. Istalgan turdagi matn uchun shrift, balandlikning egniga nisbatan va harflar og`ishining burchagi hamda matn qatorining holati va yo`nalishi kabi parametrlar berilishi zarur. Matn ikki xil:

apparat va dastur simvollarini bilan taqdim etilishi mumkin. Dasturaviy shrift oldindan kompyuter xotirasida saqlanayotgan mos grafik dasturlar tomonidan quriladi. Uni qurishga apparat shriftlari simvollarini qurishga qaraganda ko`proq vaqt ketadi, lekin uning shakli har turli bo`lishi mumkin. Apparat shriftining simvollarini harfni shakllantaruvchi alohida kesmalardan tarkib topadi.

Quyida keltirilgan listing PHIGS va OpenGL kutubxonalari yordamida ABC matn qatorini qurishni namoyish qiladi. Simvollardan OpenGLda foydalanishdan oldin ularni rastr suratlari ko`rinishida saqlash lozim.

```

PHIGS
Pvector3    direction[2] = {
                {0.0, 0.0, 0.0},
                {10.0, 10.0, 10.0}};
/* Направление строки текста задается двумя радиус-векторами. */
Ppoint3     position = {5.5, 5.0, 5.0};
/* Положение строки текста. */
ptext3(&position,direction,"ABC");
/* Построение строки текста с указанными параметрами. */

OpenGL
GLubyte     a[13] = {0x00, 0x00, 0x3f, 0x60, 0xcf, 0xdb, 0xd3, 0xdd, 0xc3, 0x7e, 0x00,
0x00, 0x00};
GLubyte     b[13] = {0x00, 0x00, 0xc3, 0xc3, 0xc3, 0xc3, 0xff, 0xc3, 0xc3, 0xc3, 0x66,
0x3c, 0x18};
GLubyte     c[13] = {0x00, 0x00, 0xf3, 0xc7, 0xc3, 0xc3, 0xc7, 0xfe, 0xc7, 0xc3, 0xc3,
0xc7, 0xfe};
/* Растровые символы А, В и С. Растр задается снизу вверх
построчно. */
GLsizei     width = 8;
/* Ширина раstra в пикселах. */
GLsizei     height = 13;
/* Высота раstra в пикселах. */

GLfloat     origin_x = 0.0, origin_y=0.0;
/* Задание начала отсчета системы координат раstra.
Положительные значения смещают начало отсчета вверх и вправо,
отрицательные - вниз и влево. */
GLfloat     incre_x = 10.0, incre_y = 0.0;
/* Поправки к положению раstra после вывода. */
GLfloat     white[3] = {1.0, 1.0, 1.0};
/* Задание цвета текста. */
GLfloat     position[2] = {20.5, 20.5};
/* Координаты символов. */
glPixelStorei(GL_UNPACK_ALIGNMENT,1);
/* Установка режима хранения пикселей, влияющего
на работу glBitmap. */
glColor3fv(white);
/* Установка белого цвета символов. */
glRasterPos2fv(position);
/* Установка текущего положения раstra. */

```

```

glColor3fv(white):
    /* Установка белого цвета символов. */
glRasterPos2fv(position):
    /* Установка текущего положения растра. */

glBitmap(width, height, origin_x, origin_y, incre_x, incre_y, asterisk):
    /* Выводит растровую картинку, заданную в последнем аргументе,
    представляющем собой указатель на эту картинку. Положение и размер
    растра указываются первыми четырьмя аргументами. */

```

3.5. Grafikani kiritish

Yuqorida qayd qilinganidek, grafik dasturdan nafaqat raqamlar va matn qatorlarini, balki nuqta, kesma va ko`pburchaklar kabi grafik elementlari kiritishni quvvatlash talab qilinishi mumkin. Masalan, agar foydalanuvchi ekranda ko`pburchak yuzasini hisoblamoqchi yoki uning o`lchamlarini kattalashtirmoqchi bo`lsa, u dastlab ekranda ko`rinayotgan boshqa obyektlar orasidan uni qiziqtirayotgan ko`pburchakni ko`rsatishi lozim. Grafikani kiritish uchun fizik qurilmalarning ikki turi: lokator (koordinatalarni kiritish qurilmasi) va knopkadan foydalaniladi. *Lokator (locator)* grafik dasturga o`zining holati, ya`ni kursorning holati haqida informatsiyani uzatadi. *Knopka (button)* kursorning joriy holatida foydalanuvchining amali (ulash va uzish) haqida xabar beradi. Bugungi kunda grafik kiritishning eng ko`p tarqalgan qurilmasi – sichqonchadir, u ikkala funksiyani bajaradi. Sichqon konusining pastki qismidagi zo`ldir (sharik) koordinatalarni kiritish imkonini beradi, korpus yuqori qismidagi knopkalar esa dasturga foydalanuvchi amali haqida xabarni uzatadi. Grafik kiritish qurilmasi uch: so`rab chiqish, so`roq yuborish va tanlash rejimlarida ishlashi mumkin. *So`rab chiqish (sampling)* rejimida kiritish qurilmasining holatini, birinchi navbatda lokator holatini, doimiy hisoblar borish amalga oshiriladi. Masalan, siz sichqonni siljitib ekranda rasm chizsangiz, u so`rab chiqish rejimida ishlaydi. Sichqonning siljishi kursorning ekran bo`ylab uzluksiz siljishiga olib keladi.

So`roq yuborish (requesting) rejimida lokatorning holati faqat so`roq yuborish yuborilgandagina hisoblanadi, u odatda sichqok knopkasi bosilganda amalga oshadi. So`rab chiqish va so`roq yuborish rejimlari orasidagi farqni oydinlashtirish maqsadida ko`pburchak qurish jarayonini ko`rib chiqamiz,

bundako`pburchak cho`qqilarining eoodinatalari sichqon yordamida grafik berilgan bo`lsin. Bu holda biz kursor zarur joyda bo`lmagunicha sichqonni siljitamiz, bundan keyin knopkani bosamiz. Kursor ekran bo`ylab sichqon harakatiga muvofiq siljiydi, u bunda so`rab chiqish rejimida bo`ladi. Cho`qqilarning koordinatalari grafik dasturga so`roq yuborish rejimida uzatiladi. Bu rejimlar umumiy xossaga ega: grafik turga sichqon yoki kursor koordinatalari uzatiladi.

Tanlash (picking) rejimida grafik kiritish qurilmasi knopka bosilayotgan onda kursor ko`rsatayotgan ekran elementlarini indentifikasiya qiladi. Grafik elementlari dastur tuzilayotgan paytda dasturchi ularga bergan nomlar bo`yicha indentifikasiyalash mumkin. Vibor rejimi ekranda mavjud bo`lgan chizmani redaksiyalash (masalan, ko`pburchaklarni yo`qotish yoki ular cho`qqilarining koordinatalarini o`zgartirish uchun) juda qulay bo`ladi.

3.6. Displey fayli

Displey fayli (display list) – bu grafik kutubxona komandalarining guruhi bo`lib, u keyinchalik bajarilishi uchun saqlanadi. Grafik kutubxona komandalarining ko`pchiligi yoki displey faylida joylashishi yoki zudlik bilan bajarilishi mumkin. Displey fayli kutubxona komandalarini tartibga soli shva ularga ishlov berish qulayligi va samaradorligini ta`minlaydi. Misol tariqasida uy tasvirining ekran bo`ylab siljishini ko`rib chiqamiz. Faraz qilaylik, rasm bir necha yuz kesmalardan tarkib topgan bo`lsin. Agar bu kesmalar alohida-alohida mavjud bo`lganda edi, bizga siljish komandasini yuzlab marotaba – har bir kesma uchun alohida yozib chiqishga to`g`ri kelar edi. Ammo, agar rasmni tashkil qiluvchi kesmalarni qurish komandalari displeyli faylga jamlangan bo`lsa, u holda siljish komandasini atigi bir marta yozish kifoya bo`ladi. Grafik elementlarni displeyli faylga joylashtirish uchun bu faylni birinchi komandadan oldin ochish zarur, komanda faylga tushishi kerak va oxirgi komandadan keyin uni yonish lozim (quyidagi misolga qarang).

```

OpenGL
glNewList(AREA_FILL, GL_COMPILE_AND_EXECUTE);
/* Открытие дисплейного файла с именем AREA_FILL. */
    glBegin(GL_POLYGON);
        glVertex2fv(point1);
        glVertex2fv(point2);
        :
        :
    glEnd();
    /* Определение многоугольника. */
glEndList();
/* Закрытие дисплейного файла. */

```

OpenGL display fayli unumdorlikni, xususan, tarmoqda ishlaganda, optimallashtirishga mo'ljallangan; bunda alohida kompyuterdagi unumdorlik ortmaydi. Optimallashtirish shuning hisobiga ta'minlanadiki, display fayli ma'lumotlarning dinamik bazasi ko'rinishida emas, balki komandalar ro'yxati ko'rinishida saqlanadi. Boshqacha aytganda, yaratilgan display faylini o'zgartirib bo'lmaydi. Agar uni o'zgartirish mumkin bo'lganda edi, ro'yxat ichida qidirish va xotirani boshqarishga bo'lgan qo'shimcha sarflar tufayli unumdorlik pasayib ketar edi. Display faylining alohida qismlarini o'zgartirish xotiraning qayta taqsimlanishini talab qilar edi, bu esa uning fragmentasiyalanishga olib kelishi mumkin edi. Doimiy fayllari guruhlariga birlashmagan komandalarning oddiy ketma-ketligi kabi tez ishlaydi. OpenGL display fayllari unumdorlikni sezilarli darajada oshirish mumkin, ayniqsa OpenGL nimdasturlari tarmoq bo'ylab uzatilayotganda, chunki bu fayllar serverda saqlanadi, shu tufayli tarmoq grafigi qisqaradi.

Yaratilgan display fayliga quyidagi operatsiyalar qo'llanilishi mumkin:

- ko'p marta bajarilish (multiple execution) – bir faylning o'zi qayta-qayta ko'p marta bajarilishi mumkin;
- ierarxik bajarilish (hierarchical execution) – boshqa (zuvrat) display fayllarini chaqiruvchi (otaxon) display fayli ierarxik deyiladi. Ierarxik display fayllari alohida komponentlardan tarkib topgan obyektlar uchun, ayniqsa ba'zi komponentlar obyektga bir necha ekzemplardi kirgan bo'lsa, qulay bo'ladi;
- o'chirish (udalenie, deletion) – display fayli o'chirilishi mumkin.

3.7. O`zgartirish matritsasi

3.2 bo`limda bayon qilinganidek uch o`lchamli fazada nuqtalarni obyektga proyeksiyalash koordinatalarni bir tizimdan boshqasiga o`zgartirish talab qiladi. Dastlab obyekt nuqtalarining koordinatalarini model tizimidan dunyo tizimiga o`tkazish kerak. Obyektning joriy holati boshlang`ich holatga nisbatan odatda burili shva siljishlar orqali beriladi; boshlang`ich holatda model koordinatalar tizimi dunyoviy bilan ustma-ust tushgan bo`ladi. Demak, obyekt nuqtalarning dunyoviy koordinatalarini ularning boshlang`ich holatidan (bu holatda model koordinatalari dunyoviy bilan ustma-ust tushar edi) translyatsiya qilish va burash yo`li bilan olish mumkin. Grafik kutubxonalarining aksariyati bu o`zgartishlarni mustaqil ravishda bajaradi, dasturchiga esa faqat uni qiziqtirayotgan obyekt uchun siljish va buralishni berish qoladi xolos. Lekin dasturchi baribir o`zgartishlar qonunlarini bilishi zarur, chunki unga obyektlarni kerakli joylarda hatoliklarsiz chizishga to`g`ri keladi, bu ayniqsa obyektlar murakkab siljiganda murakkab masaladir. Bu bo`limda biz obyekt siljishi va buralishda uning nuqtalariga ta`sir qiluvchi o`zgartishlar matritsalarini ko`rib chiqamiz.

Obyektning joriy holatida uning hamma nuqtalarining dunyo koordinatalarini olib, biz bu nuqtalarning koordinatalarini kuzatuvchi tizimdan ikkinchisiga o`tkazish aks ettirish (*mapping*) deyiladi. Dunyo va kuzatuvchi koordinatalar tizimlari orasida aks ettirish odatda grafik kutubxona tomonidan mustaqil ravishda, dasturchi belgilangan ko`rish nuqtasi, kuzatish nuqtasi koordinatalari va vertikal vektori yo`nalishi (dunyo koordinatalari) bo`yicha bajariladi. Aks ettirish operatsiyasi uchun o`zgartishlar matritsasi 3.7.3. bo`limda ko`rib chiqiladi.

3.7.1. Translyatsiya

Boshlang`ich holatida model koordinatalar tizimi dunyoviy bilan mos kelgan obyekt x , u va z yo`nalishlarida mos ravishda a , b va s qiymatga translyatsiya qilinganida (3.16-rasm), yangi holatda obyekt nuqtalarining dunyo koordinatalari (X_w, Y_w, Z_w) quyidagi tarzda hisoblanadi:

$$\begin{aligned}
 X_w &= X_m + a; \\
 Y_w &= Y_m + b; \\
 Z_w &= Z_m + c.
 \end{aligned}
 \tag{3.3}$$

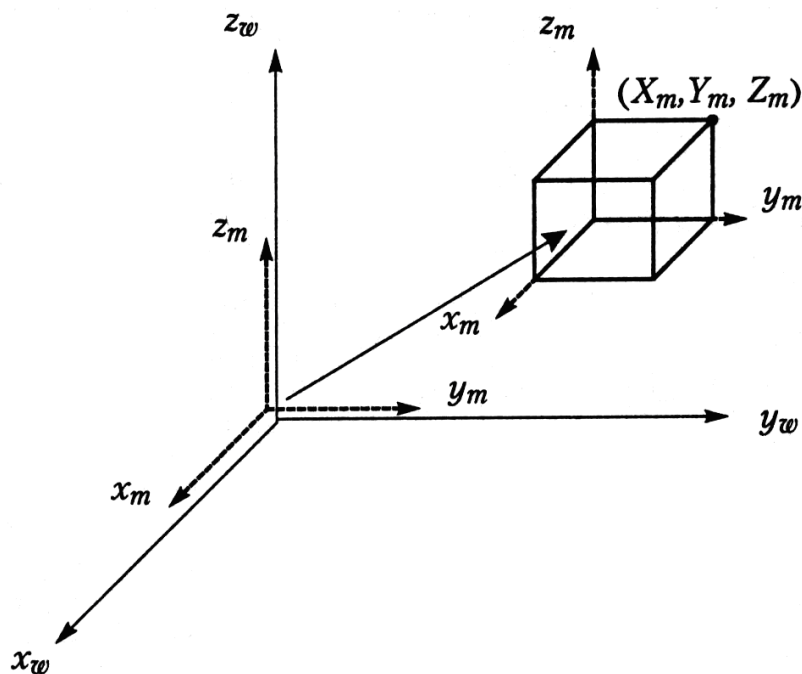
Bu formulada X_m , Y_m , Z_m raqamlari ham nuqtaning model koordinatalari bo`ladi.

(3.3) formula matritsali shaklda yozilishi mumkin¹:

$$\begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & a \\ 0 & 1 & 0 & b \\ 0 & 0 & 1 & c \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_m \\ Y_m \\ Z_m \\ 1 \end{bmatrix}$$

\Downarrow
 $Trans(a, b, c)$

(3.4)



3.16-rasm. Obyektni translyatsiya qilish

¹ Koordinatalar ham qator ko`rinishida taqdim etilishi mumkin. Unda o`zgartishlar matritsasi koordinatalar vektori ko`rinishida yoziladi. Bu holda o`zgartishlar matritsasi (3.4) formuladan transponirovka qilingan matritsa ko`rinishiga ega bo`ladi. (3.4) formulada OpenGL yozuvi haqidagi bitimga rioya qildik.

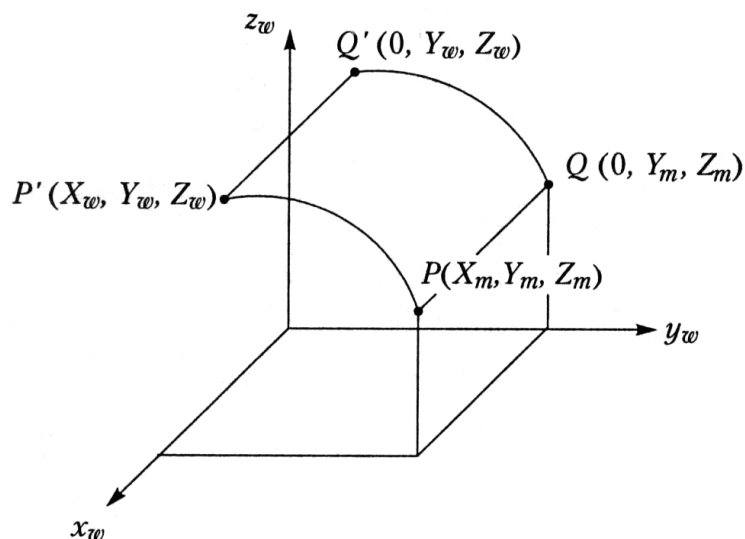
(3.4) va (3.3) formulalar bir-biriga ekvivalentligiga osonlik bilan inonish mumkin: buning uchun (3.4) formulani yoyilgan ko`rinishda yozish kifoya qiladi.

(3.3)dagi qo`shish operatsiyasi (3.4)da ko`paytma orqali, buning uchun uch o`lchamli vektor uchta skalyar o`rniga to`rtta skalyar orqali yoziladigan bir jinsli koordinatalardan foydalaniladi¹. Biz hosil qilgan matritsa *bir jinsli o`zgartish matritsasi (homogeneous transformation matrix)* deb ataladi. Bu holda o`zgartish – bu translyatsiyadir. Agar o`zgartishni, xususan translyatsiyani, ikki o`lchamli fazada nuqtacha qo`llash zarur bo`lsa, o`zgartishning bir jinsli matritsa 4×4 o`lchamli matritsadan uchinchi qator va uchinchi ustunni yo`qotish bilan 3×3 o`lchamli matritsaga aylanadi. Yangi matritsa 3×1 o`lchamli koordinatalar vektorinikidir, u 4×1 vektordan z koordinatasi yo`qotilishi natijasida olingan.

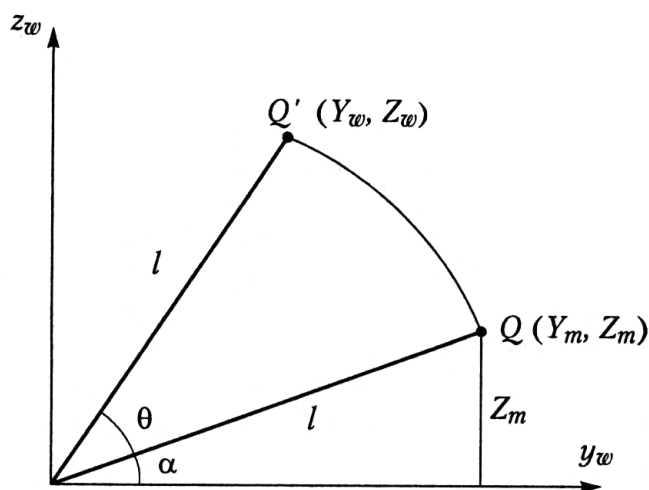
3.7.2. Aylantirish (burash)

Obyekt dunyo koordinatalar tizimining x o`qi atrofida o`zining model koordinatalar tizimi bilan birga 0 burchakka buralsin; model tizimi, oldingi misoldagi kabi, boshidan dunyo tizimi bilan ustma-ust tushgan bo`lsin (3.17-rasm). Yangi holatda obyekt nuqtasining dunyo koordinatalari (X_w, Y_w, Z_w) ushbu nuqtaning model tizimidagi joriy koordinatalari bilan ustma-ust tushadigan boshlang`ich dunyo koordinatalari (X_t, Y_m, Z_m) dan olinishi mumkin. 3.17-rasm yz tekisligiga proyeksiyalangandan so`ng (X_w, Y_w, Z_w) va (X_t, Y_m, Z_m) lar orasidagi nisbat yaqqol ko`zga tashlanadi. Proeksiyalash natijasi 3.18-rasmda ko`rsatilgan.

¹ Uch o`lchamli fazoning istalgan vektori $(x, u, z)^T$ mos bir jinsli koordinatalarda $(x_w, y_w, z_w, w)^T$ yozilishi mumkin, bu erda yuqoridagi indeks T transponirovka qilish operatsiyasini belgilaydi. w qiymati ixtiyoriy bo`lishi mumkinligi sababli, har bir vektor uchun bir jinsli koordinatalarda yozishning ko`p variantlari mavjud. (3.4) formulada $w = 1$ qiymatdan foydalanilgan.



3.17-rasm. X o`qi atrofida aylantirish



3.18-rasm. yz tekisligiga proyeksiyalash

3.18-rasmdan quyidagi tengliklarni osonlik bilan olish mumkin:

$$X_w = X_m \quad (3.5)$$

$$\begin{aligned} Y_w &= l \cos(\theta + \alpha) = l(\cos \theta \cos \alpha - \sin \theta \sin \alpha) = \\ &= l \cos \alpha \cos \theta - l \sin \alpha \sin \theta = Y_m \cos \theta - Z_m \sin \theta \end{aligned} \quad (3.6)$$

$$\begin{aligned} Z_w &= l \sin(\theta + \alpha) = l(\sin \theta \cos \alpha + \cos \theta \sin \alpha) = \\ &= l \cos \alpha \sin \theta + l \sin \alpha \cos \theta = Y_m \sin \theta + Z_m \cos \theta \end{aligned} \quad (3.7)$$

(3.5), (3.6) va (3.7) tengliklar matritsa shaklida yozilishi mumkin:

$$\begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \theta & -\sin \theta & 0 \\ 0 & -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_m \\ Y_m \\ Z_m \\ 1 \end{bmatrix} \quad (3.8)$$

(3.8) formulaning o`ng qismidagi matritsa $-x$ o`qi atrofida aylantirishning bir jinsli o`zgartish matritsasidir, uning qisqacha belgilanishi $-Rot(x, \theta)$. Translyatsiya matritsasiga o`xshab, ikki o`lchamli obyekt uchun aylanishning bir jinsli matritsasi 3×3 o`lchamga reduksiyalanadi.

u va z o`qlari atrofida aylanishning bir jinsli matritsalarini o`xshash tarzda olinadi va quyidagicha yoziladi:

$$Rot(y, \theta) = \begin{bmatrix} \cos \theta & 0 & \sin \theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin \theta & 0 & \cos \theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.9)$$

$$Rot(z, \theta) = \begin{bmatrix} \cos \theta & -\sin \theta & 0 & 0 \\ \sin \theta & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.10)$$

Biz duny koordinata o`qlari atrofida aylanishni bayon qiluvchi o`zgartishlar matritsalarini oldik. Istalgan ixtiyoriy o`q atrofida aylanishni x , u va z o`qlari atrofida aylanishlarga yoyish mumkin. Shunday qilib, ixtiyoriy o`q uchun o`zgartish matritsasi (3.8)-(3.10) matritsalarini bir-biriga ko`paytirish bilan olinadi.

Yuqorida qayd qilinganidek, bu bo`limda bayon qilinayotgan o`zgartishlar matritsalarini, odatda grafik kutubxonalarining mos nimdasturlari tomonidan hisoblanadi. Quyida keltirilgan kod PHIGS va OpenGL nimdasturlaridan foydalanishni illyustratsiya qiladi.

PHIGS

```
ptranslate3(Pvector* offset3, Pint* error_ind, Pmatrix3 result3);
/* offset3: вектор трансляции
   error_ind: код ошибки
   result3: вычисленная матрица преобразования1 */
protate_x(Pfloat angle, Pint* error_ind, Pmatrix3 result3);
protate_y(Pfloat angle, Pint* error_ind, Pmatrix3 result3);
protate_z(Pfloat angle, Pint* error_ind, Pmatrix3 result3);
/* angle: угол поворота
   error_ind: код ошибки
   result3: вычисленная матрица преобразования */
```

OpenGL

```
glTranslated(GLdouble offset_x, GLdouble offset_y, GLdouble offset_z);
/* Умножает текущую матрицу на матрицу трансляции объекта со
   смещениями offset_x, offset_y и offset_z по соответствующим осям. */

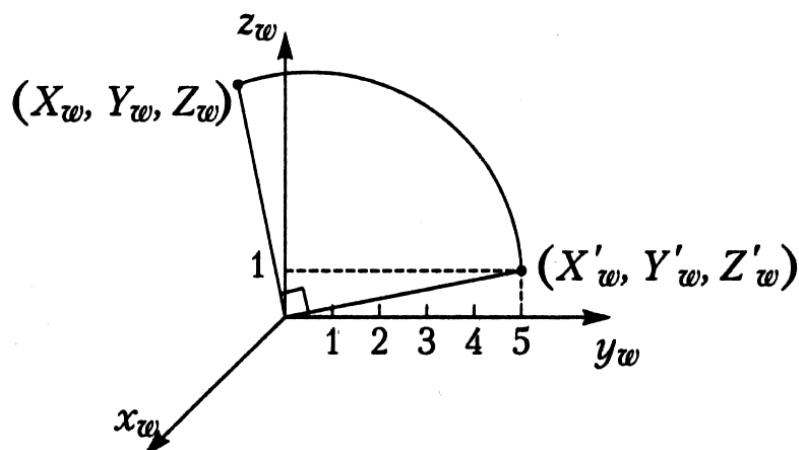
glRotated(GLdouble angle, GLdouble x, GLdouble y, GLdouble z);
/* Умножает текущую матрицу на матрицу поворота объекта на угол
   angle против часовой стрелки вокруг луча, проведенного из начала
   координат в точку (x, y, z). */
```

¹ PHIGSda o`zgartish matritsasi vektor-qatordan keyin qo`yiladi.

O`zgartishlar matritsasini qo`llashni oydinlashtiruvchi bir nechta misollarni ko`rib chiqamiz.

Misol 3.1

Obyekt uch o`lchamli fazada dunyo koordinatalar tizimining u o`qi yo`nalishida 5 birlikka translyatsiya qilinadi, so`ngra o`sha koordinatal tizimining x o`qi atrofida 90° ga buriladi. Agar obyekt nuqtalrining eordinatalari model tizimida $(0, 0, 1)$ qiymatlarga ega bo`lsa, translyatsiya va burilishdan keyin bu nuqtaning dunyo koordinatalari qanday bo`ladi?



Yechim

(X'_w, Y'_w, Z'_w) koordinatalari translyatsiya o`zgartirilgandan keyin o`uyidagi tarzda hisoblab topilishi mumkin:

$$\begin{bmatrix} X'_w & Y'_w & Z'_w \end{bmatrix}^T = \text{Trans}(0, 5, 0) \begin{bmatrix} 0 & 0 & 1 & 1 \end{bmatrix}^T = \begin{bmatrix} 0 & 5 & 1 & 1 \end{bmatrix}^T.$$

Bundan keyin buralishni o`zgartirish qo`llaniladi:

$$\begin{bmatrix} X_w & Y_w & Z_w & 1 \end{bmatrix}^T = \text{Rot}(x, 90^\circ) \begin{bmatrix} 0 & 5 & 1 & 1 \end{bmatrix}^T$$

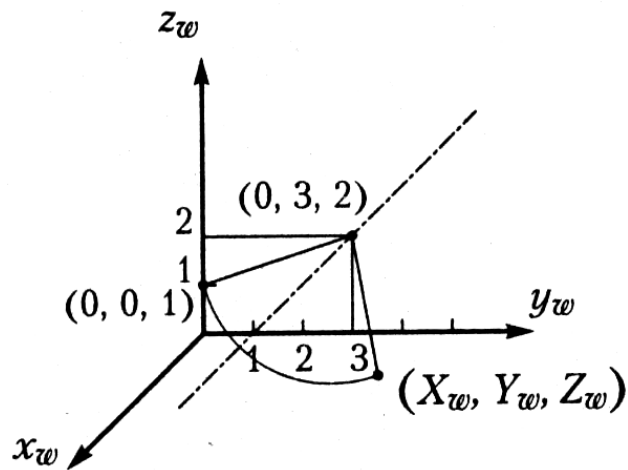
Demak, nuqta koordinatalari o`zgartishlardan keyin $(0, -1, 5)$ qiymatlarga ega bo`ladi. Oldingi ifodalarni birlashtirish mumkinligiga e`tibor bering:

$$\begin{bmatrix} X_w & Y_w & Z_w & 1 \end{bmatrix}^T = \text{Rot}(x, 90^\circ) \text{Trans}(0, 5, 0) \begin{bmatrix} 0 & 0 & 1 & 1 \end{bmatrix}^T$$

Oxirgi ifoda ayniqsa ko`p nuqtalarning koordinatalarini hisoblashda ancha qulay. Oxirgi holda translyatsiya va aylanish matritsalarini oldindan bir-biriga ko`paytiriladi va o`zgartirishning ekvivalent matritsasini beradi, u bizgi qiziqtirayotgan hamma nuqtalarga ta`sir qiladi. O`zgartirishlarning alohida matritsalarini mos ketma-ketlikda bir-biriga ko`paytirish yo`li bilan o`zgartirishlarning ekvivalent matritsasini hisoblash jarayoni *konkatenasiya* (*concatenation*) deb ataladi. Konkatenasiyani bajarish imkoniyati bir jinsli koordinatalar tizimidan foydalanishning afzalliklaridan biridir; unda translyatsiya qo`shish orqali emas, balki matritsali ko`paytirish orqali yoziladi.

Misol 3.2

Obyekt fazoda dunyo koordinatalari tizimining x o`qiga parallel bo`lgan va dunyo koordinatalari $(0, 3, 2)$ bo`lgan nuqtadan o`tuvchi o`q atrofida 90° ga buraladi. Agar obyekt nuqtasi $(0, 0, 1)$ model koordinatalariga ega bo`lsa, buralishdan keyin o`sha nuqtaning dunyo koordinatalari qanday bo`ladi?



Yechim

Biz faqat koordinatalar boshidan o`tuvchi o`qlarga nisbatan ko`rib chiqdik, shuning uchun endi bizga obyektning aylanish o`qi bilan birga siljitishga to`g`ri keladi. Aylanish o`qi koordinatalar boshidan o`yishi kerak, bunda obyektning ushbu o`qqa nisbatan holati saqlanib qolishi lozim. Obyektning o`q bilan birga (0, -3, -2) vektorga translyatsiya qilish aylanish o`qining dunyo koordinatasining x o`qi bilan ustma-ust tushishini beradi. So`ngra biz obyektning x o`qi atrofida 90°ga aylantiramiz, bundan keyin boshlang`ich holatga qaytish uchun obyektning qaytadan (0, 3, 2) vektorga joylashtiramiz.

Bu operatsiyalar quyidagi tarzda yozilishi mumkin:

$$\begin{bmatrix} X_w & Y_w & Z_w & 1 \end{bmatrix}^T = \text{Trans}(0, 3, 2) \text{Rot}(x, 90^\circ) \text{Trans}(0, -3, -2) \begin{bmatrix} 0 & 0 & 1 & 1 \end{bmatrix}^T$$

Bu formuladagi matritsalar ketma-ketligiga e`tibor bering. 3.1-misoldagi kabi o`zgartishlarni ketma-ket qo`llab, natijani osonlikcha tekshirish mumkin.

Ifodani yozib, quyidagini olamiz:

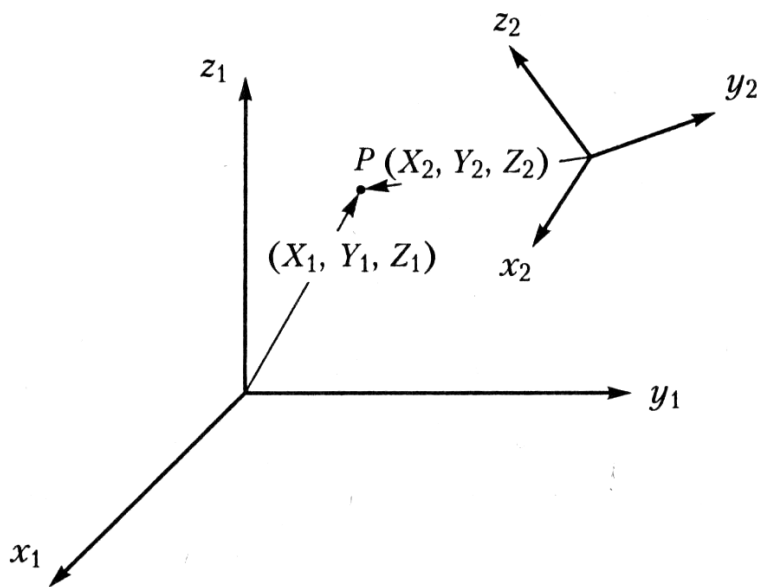
$$\begin{aligned} \begin{bmatrix} X_w & Y_w & Z_w & 1 \end{bmatrix}^T &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 3 \\ 0 & 0 & 1 & 2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos 90^\circ & -\sin 90^\circ & 0 \\ 0 & \sin 90^\circ & \cos 90^\circ & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & -3 \\ 0 & 0 & 1 & -2 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \\ &= \begin{bmatrix} 0 & 4 & -1 & 1 \end{bmatrix}^T \end{aligned}$$

Natija yuqorida keltirilgan illyustratsiyaga mos keladi.

3.7.3. Aks tasvir

Aks tasvir (mapping) – bu nuqtaning koordinatalarini qaysidir koordinatalar tizimida o`sha nuqtaning boshqa koordinatalar tizimida mavjud bo`lgan koordinatalari bo`yicha topishdir. Ikkita koordinatalar tizimlarini ko`rib chiqamiz (3.19-rasm). Faraz qilaylik, R nuqtaning x_2, y_2, z_2 koordinatalar tizimidagi koordinatalari (X_2, Y_2, Z_2) o`sha nuqtaning x_1, y_1, z_1 tizimidagi koordinatalari (X_1, Y_1, Z_1) bo`yicha hisoblab topilishi kerak bo`lsin. Yana faraz qilamiz, hisoblash o`zgartirish matritsasi T_{1-2} ni ma`lum koordinatalarga qo`llash bilan amalga oshiriladi:

$$\begin{bmatrix} X_2 & Y_2 & Z_2 & 1 \end{bmatrix}^T = T_{1-2} \begin{bmatrix} X_1 & Y_1 & Z_1 & 1 \end{bmatrix}^T. \quad (3.11)$$



3.19-rasm. Bir koordinata tizimidan ikkinchisiga aks ettirish

T_{1-2} matritsani ochiq ko`rinishda yozib, biz (3.11) formuladan quyidagi ifodani olamiz:

$$\begin{bmatrix} X_2 \\ Y_2 \\ Z_2 \\ 1 \end{bmatrix} = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_1 \\ Y_1 \\ Z_1 \\ 1 \end{bmatrix}. \quad (3.12)$$

(3.12) tenglamadagi nomalarni topish uchun, unga $X_1=0$, $Y_1=0$ va $Z_1=0$ muayyan qiymatlarni qo'yamiz, natijada quyidagilarni hosil qilamiz:

$$X_2 = p_x, Y_2 = p_y, Z_2 = p_z. \quad (3.13)$$

r_x, r_u va r_z x_2, y_2, z_2 tizimning hisob boshi koordinatalarini aniqlaydi deyish mumkin.

Endi (3.12) tenglamaga $X_1=1$, $Y_1=0$, $Z_1=0$ qiymatlarni qo'yib, quyidagini olamiz:

$$X_2 = n_x + p_x, Y_2 = n_y + p_y, Z_2 = n_z + p_z. \quad (3.14)$$

(3.14) formuladan (3.13)ni ayirib, quyidagi xulosaga kelish mumkin: n_x, n_y, n_z – bu x_1, y_1, z_1 koordinatalar tizimining x_1 o'qi bo'ylab yo'nalgan birlik vektori x_2, y_2, z_2 ning komponentlaridir. Demak, n_x, n_y, n_z koeffisientlarni koordinata tizimlarining o'zaro oriyentatsiyasini hisobga olgan holda osonlik bilan hisoblab topish mumkin.

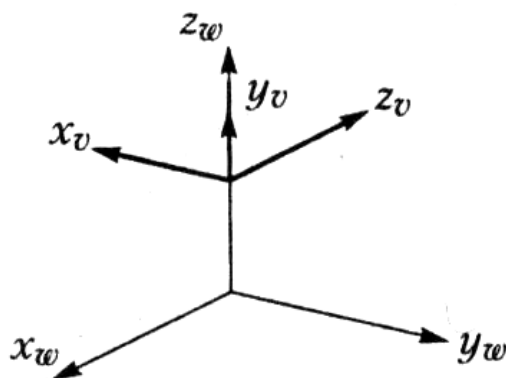
Shunga o'xshash, σ_x, σ_y va σ_z – y_1 o'qi birlik vektorining x_2, y_2, z_2 komponentlaridir, a_x, a_y va a_z – esa z_1 o'qining komponentlaridir.

Keyingi misol taklif qilinayotgan nazariyani amalda namoyish qiladi.

Misol 3.3

Berilgan ko'rish nuqtasi $(-10, 0, 1)$, kuzatish nuqtasi $(0, 0, 1)$ va vertikal vektori $(0, 0, 1)$ holatlari bo'yicha kuzatish koordinatalar tizimi quriladi (pastda keltirilgan rasimga qarang). E'tibor bering, vektorlarning hamma koordinatalari va komponentlari dunyo koordinatalar tizimida berilgan. Ko'rib chiqish koordinatalar tizimi va dunyo koordinatalar tizimlarining bir-biriga nisbatan joylashishini bilgan holda:

- o'zgartishlar matritsasi T_{w-v} ni;
- dunyoviy koordinatalari $(5,0,1)$ nuqtaning kuzatuvchi koordinatalar tizimidagi koordinatalarini hisoblang.



Yechim

T_{w-v} birinchi ustunining birinchi uchta raqami (ya'ni n_x , n_y va n_z) $(0 \ 0 \ -1)$ ga teng, chunki ular x_w o'qining x_v , y_v va z_v komponentlarini taqdim etadi. Shunga o'xshab, y_w o'qining x_v , y_v va z_v komponentlarini taqdim etuvchi o_x , o_y va o_z lar $(-1 \ 0 \ 0)$ ga, a_x , a_y va a_z lar esa $(0 \ 1 \ 0)$ ga teng bo'ladi. r_x , r_y va r_z – bu x_w , y_w va z_w tizim koordinatalari boshi x_v , y_v va z_v larning koordinatalaridir, shuning uchun mos ravishda 0, -1 va 0 ga teng. Natijada T_{w-v} matritsani olamiz.

$$T_{w-v} = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & -1 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

$(5, 0, 1)$ nuqtaning kuzatuvchi tizimdagi koordinatalarini bu koordinatalar vektoriga hozir hisoblab topilgan aks ettiruvchi matritsa T_{w-v} ni qo'llab topamiz.

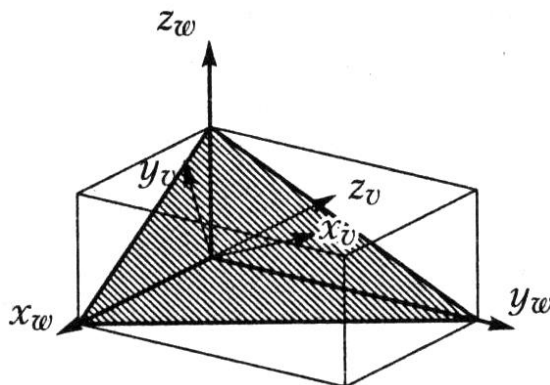
$$\begin{bmatrix} X_v \\ Y_v \\ Z_v \\ 1 \end{bmatrix} = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & -1 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 5 \\ 0 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -5 \\ 1 \end{bmatrix}$$

Demak, $(5, 0, 1)$ nuqta koordinatalari kuzatuvchi tizimida $(0, 0, -5)$ qiymatlarga ega bo'ladi, bunga keltirilgan rasmga qarab inonish mumkin.

Misol 3.4

Qurish va kuzatish nuqtalari mos ravishda $(5, 5, 5)$ va $(0, 0, 0)$ koordinatalarga ega, vertikal vektori esa $(0, 0, 1)$ ga teng tanlanadi. Proeksiya

izotermik. Aks ettirishni o'zgartirish matritsasi T_{w-v} va koordinatalari dunyoviy bo'lgan nuqtaning kuzatuvchi nuqtalarini hisoblash kerak.



Yechish

Kuzatuvchi koordinatalar tizimi yuqoridagi rasmda ko'rsatilganidek tasvirlanishi mumkin. Bu rasmdagi shtrixlangan uchburchak ekran tekisligiga parallel. Bu tekislikda x_v va y_v o'qlari yotadi.

T_{w-v} elementlarini hisoblab topish uchun, bizga x_w , y_w va z_w o'qlarining x_v , y_v va z_v komponentlarini olish zarur. Buning x_v , y_v va z_v o'qlarining birlik vektorlarini mos ravishda i_v , j_v va k_v harflari bilan belgilaymiz. x_w , y_w va z_w o'qlarining birlik vektorlari oddiygina i , j va k lar bilan belgilanadi.

Birlik vektori k_v kuzatish nuqtasidan ko'rish nuqtasiga yo'nalgan, shuning uchun

$$k_v = \frac{1}{\sqrt{3}}i + \frac{1}{\sqrt{3}}j + \frac{1}{\sqrt{3}}k.$$

3.2 bo'limda qayd qilinganidek, birlik vektori j_v vertikal vektorining ekranga proyeksiyasi kollinear bo'lishi kerak. Boshqacha aytganda uning yo'nalishi, vertikal vektoridan uning ekranga perpendikulyar tashkil etuvchisini ayirgandan keyin hosil bo'lgan vektor yo'nalishi bilan, ustma-ust tushadi. Vertikal vektorini u_p bilan belgilab, j_v uchun ifoda yozamiz.

$$j_v = \frac{u_p - (u_p \cdot k_v)k_v}{|u_p - (u_p \cdot k_v)k_v|} = \frac{-\frac{1}{3}i - \frac{1}{3}j + \frac{1}{3}k}{\left|-\frac{1}{3}i - \frac{1}{3}j + \frac{1}{3}k\right|} = -\frac{1}{\sqrt{6}}i - \frac{1}{\sqrt{6}}j + \frac{2}{\sqrt{6}}k.$$

Oxirgi birlik vektori i_v quyidagi vektor ko'paytma orqali hisoblanadi:

$$i_v = j_v \times k_v = \frac{1}{-\sqrt{2}}i + \frac{1}{\sqrt{2}}j.$$

Endi x_v o'qi x_w komponentasi p_x ni ko'paytma kabi hisoblaymiz:

$$i \cdot i_v = \frac{1}{-\sqrt{2}}.$$

Shunga o'xshash, p_u komponentasi quyidagiga teng:

$$i \cdot j_v = \frac{1}{-\sqrt{6}}.$$

p_z komponenta esa

$$i \cdot k_v = \frac{1}{\sqrt{3}}.$$

T_{w-v} matritsasining ikkinchi va uchinchi ustunlarini ham shu yo'sinda olamiz. r_x , r_y va r_z komponentalarni hisoblashga hojat yo'q, chunki bu misolda kuzatuvchi va dunyo tizimlari koordinatalarning boshi ustma-ust tushgan.

Demak, o'zgartishlar matritsasi quyidagi ko'rinishga ega bo'ladi:

$$T_{w-v} = \begin{bmatrix} -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 & 0 \\ -\frac{1}{\sqrt{6}} & -\frac{1}{\sqrt{6}} & -\frac{1}{\sqrt{2}} & 0 \\ \frac{1}{\sqrt{3}} & \frac{1}{\sqrt{3}} & \frac{1}{\sqrt{3}} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

(0, 0, 5) nuqtaning kuzatuvchi tizimidagi koordinatalarining qiymatlari

$$[X_v \ Y_v \ Z_v \ 1]^T = T_{w-v} \cdot [0 \ 0 \ 5 \ 1]^T = \begin{bmatrix} 0 & \frac{5\sqrt{6}}{3} & \frac{5\sqrt{6}}{3} & 1 \end{bmatrix}^T.$$

Izometrik proyeksiyada nuqtaning ekran koordinatalari kuzatuvchi koordinatalaridan bevosita olinadi:

$$\begin{pmatrix} 0 & \frac{5\sqrt{6}}{3} \end{pmatrix}.$$

Izometrik proyeksiya parallel proyeksiyalarga kiradi, shuning uchun z_w o`qidagi hamma nuqtalar ekran koordinatalarining u o`qiga proyeksiyalanadi. $(0, 0, 1)$ ga teng vertikal vektorining berilishi amalda shuni bildiradiki: proyeksiyalangandan keyin z_w o`q ekranda vertikal to`hri chiziq ko`rinishiga ega bo`ladi.

3.7.4. *Masshtab va ko`zgu kabi aks ettirish matritsalarini*

Yuqoridagi bo`limlarda ko`rib chiqilgan qzgartishlar matritsalaridan tashqari, ko`pincha masshtablash va ko`zgu kabi aks ettirish matritsalaridan ham foydalaniladi.

Obyektning x o`qi bo`ylab s_x , u o`qi bo`ylab s_y , po z o`qi bo`ylab s_z koeffitsiyentlar bilan *masshtablash* uchun quyidagi o`zgartirish matritsasini qo`llashadi:

$$\begin{bmatrix} X' \\ Y' \\ Z' \\ 1 \end{bmatrix} = \begin{bmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}. \quad (3.15)$$

Ikki o`lchamli obyektlar uchun masshtablag matritsasi 3×3 o`lchamgacha radiuslanadi; bu translyatsiya va aylantirish matritsalarini bilan amalga oshirilgan edi. Koordinatalar qiymatlarini o`zgartirmasdan, ko`rsatish ekrani o`lchamlarini o`zgartirib, masshtablash effektiga erishish mumkin.

Obyekt koordinatalar boshiga nisbatan masshtablanganda o`zgartirishlar matritsasi (3.15)dan foydalaniladi. Ko`pincha obyektning, uning (X_p, Y_p, Z_p) koordinatalarga ega bo`lgan R nuqtasiga nisbatan, masshtablash zarurati tug`iladi. Bu holda R nuqtaga dastlab translyatsiyani o`zgartirish $Trans(-X_p, -Y_p, -Z_p)$ qo`llanadi, u bu nuqtani koordinatalar boshiga siljitadi, so`ngra (3.15)dan masshtablash matritsasi qo`llanadi va bundan obyekt $Trans(X_p, Y_p, Z_p)$ amali bilan boshlang`ich holatga qaytariladi.

xy ko`zgu tekisligiga nisbatan *aks ettirishga* quyida keltirilgan o`zgartirishlar matritsasi yordamida erishish mumkin. O`zgartirish $-z$ koordinatasining ishorasini o`zgartirishdir.

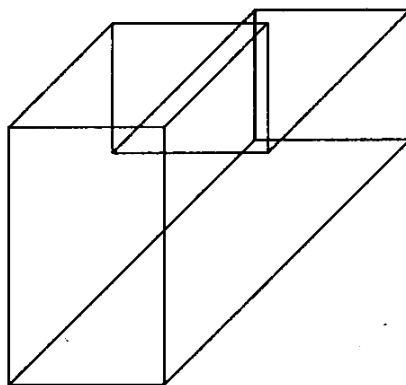
$$\begin{bmatrix} X' \\ Y' \\ Z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}. \quad (3.16)$$

Boshqacha aks ettirishlar (xz va yz tekisliklariga nisbatan) uchun o'zgartishlar matritsalarini shunga o'xshash chiqariladi.

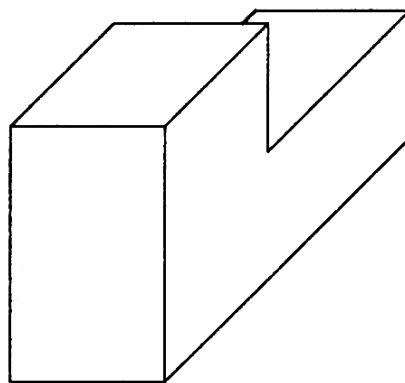
3.8. Ko`rinmaydigan chiziqlar va sirlarni yo`qotish

Agar ekranda faqat ko`rinadigan chiziqlar va sirtlar bo`lsa, proyeksiya ancha ravshanlashadi. *Ko`rinmas chiziqlarni yo`qotish (hidden-line removal)* kuzatuvchiga ko`rinmaydigan kesmalar aksini blokirovka qilish bilan amalga oshiriladi, *ko`rinmas sirlarni yo`qotish (hidden-surface removal)* esa – sirlarga nisbat o`sha amalning o`zidir. 3.20- va 3.21-rasmlar ko`rinmas chiziqlarni yo`qotishni illyustratsiya qiladi.

Bu protsedura obyektning qabul qilishni sezilarli darajada osonlashtiradi.



3.20-rasm. Ko`rinmaydigan chiziqlar yo`qotilishidan oldingi tasvir



3.21-rasm. Ko`rinmaydigan chiziqlar yo`qotilgandan keyingi tasvir

Ko`rinmaydigan chiziqlar va sirlarni yo`qotuvchi ko`p dasturaviy algoritmlar chop qilingan, ularning muammolari hisoblash samaradorligini oshirish va o`zlarining algoritmlarini qo`llash uchun obyektlar diapazonini kengaytirishga intilishgan. Lekin o`chirish z -bufer deb ataladigan grafik qurilma yordamida eng yaxshi bajariladi, shu sababli bugungi kunda bu mavzudagi tadqiqotlar deyarli olib borilmayapti.

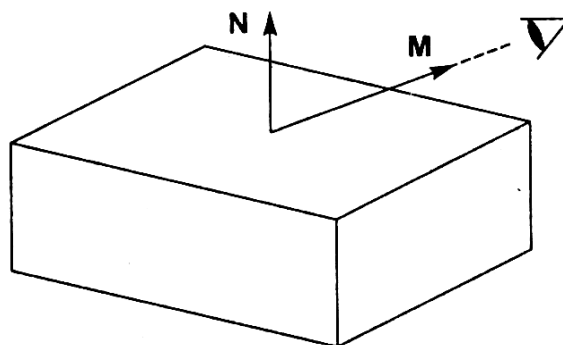
Bu bo`limda biz dasturaviy yo`l bilan ko`rinmas chiziq va sirlarni yo`qotuvchi bir nechta tipik algoritmlarni ko`rib chiqamiz hamda shu maqsadda z -bufera metodidan foydalanishni o`rganamiz.

3.8.1. Ko`rinmaydigan qirralarni o`chirish algoritmi

Ko`rinmaydigan qirralarni o`chirish algoritmi (*back-face removal algorithm*) quyidagi prinsipga asoslangan: obyektning qirradi faqat shu holda ko`rinish mumkinki, agar bu qirraga bo`lgan tashqi normal` vektori kuzatuvchi tomoniga yo`nalgan bo`lsa. Aks holda qirra ko`rinmaydi. Masalan, 3.22-rasmda tasvirlangan brusning yuqori qirradi, agar tashqi normal` vektori N , qirradagi nuqtadan kuzatuvchiga o`tkazilgan M vektori yo`nalishda, musbat tashkil etuvchiga ega bo`lsa, ko`rinuvchi hisoblanadi.

Bularni matematik ifodasi:

- agar $\mathbf{M} \cdot \mathbf{N} > 0$ bo`lsa, sirt qiriladi;
- agar $\mathbf{M} \cdot \mathbf{N} = 0$ bo`lsa, sirt kesma bo`lib proyeksiyalanadi;
- agar $\mathbf{M} \cdot \mathbf{N} < 0$ bo`lsa, sirt ko`rinmaydi.

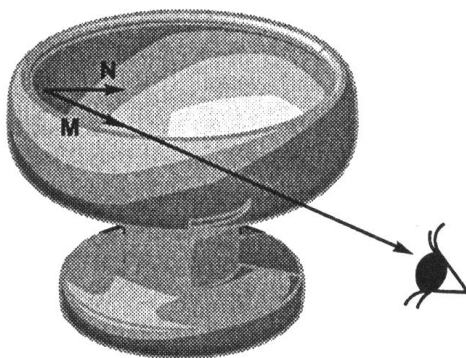


3.22-rasm. Qirra ko`rinishini belgilovchi vektorlar

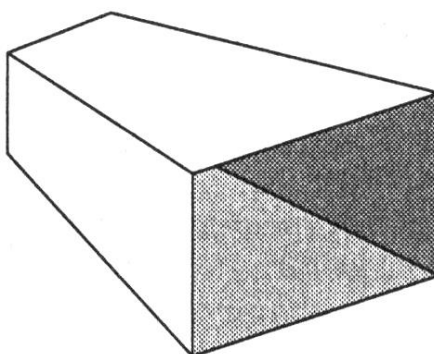
Bu algoritm tekis sirtlar bilan cheklangan obyektga oson qo`llanadi, chunki sirt chegarasida normal vektori N o`zgarmasdir. Lekin botiq¹ obyektga algoritmni qo`llab bo`lmaydi, chunki kuzatuvchiga yo`nalgan qirra o`sha obyektning boshqa qirrasini bilan berkilgan bo`lishi mumkin.

Birinchi qabariq obyektlar bo`lganda ham shu muammo vujudga keladi, ular bir-birlarining qirralarini berkitishlari mumkin. Demak, ko`rinmaydigan qirralarni yo`qotish algoritmi faqat bitta qabariq obyektga qo`llanilishi mumkin. Bundan tashqari, tashqi normal` vektori bir ma`noda aniqlanmaydigan obyektlar uchun ham bu algoritmni qo`llab bo`lmaydi (masalan, 3.24-rasmga qarang).

¹Agar obyektning kamida ikki qirrasini 180° dan katta bo`lgan ichki burchak ostida tutashsa, bunday obyekt botiq deyiladi. Agar hamma qirralar 180° dan kichik bo`lgan ichki burchaklarda tutashsa, obyekt qabariq deyiladi.



3.23-rasm. Botiq obyekt misoli



3.24-rasm. Tashqi normal` vektori bir ma`noda aniqlanmaydigan obyekt

Agar obyekt sirtlari tekis bo`lmasa N qiymati bir qirra chegarasida qaysi nuqta tanlanishiga qarab o`zgarib turadi. $M \cdot N$ ko`paytma ishorasi ham o`zgarishi mumkin. Bu shuni bildiradiki, bir qirraning o`zida ham ko`rinadigan va ham ko`rinmaydigan uchastkalar bo`ladi. Shuning uchun qirra $M \cdot N = 0$ tenglik bajariladigan egri chiziq bo`ylab ikki qismga bo`linishi kerak. Bu egri chiziq *siluet chizig`i* (*silhouette line*) deyiladi. Siluet chizig`i bo`ylab qirra bo`lingandan keyin qirraning har bo`lagida $M \cdot N$ ishorasi o`zgarmas bo`ladi. Protsedura oson tuyulishi mumkin, lekin siluet chizig`ini hisoblab topish ancha murakkabdir vash u tufayli ko`rinmaydigan qirralarni yo`qotish algoritmining asosiy afzalligi – realizatsiya qilish osonligi yo`qoladi.

Hamma qirralar ko`rinadigan yoki ko`rinmaydigan sifatida tasniflangandan so`ng, ekranga ko`rinadigan qirralarning qobirg`alari chiqariladi, natijada ko`rinmaydigan chiziqlarsiz rasm hosil bo`ladi. Agar ko`rinmaydigan sirtlarsiz rasm hosil qilish lozim bo`lsa, ko`rinadigan sirtlar tanlangan ranglarga bo`yaladi.

3.8.2. Ko`rinmaydigan chiziqlarni o`chirish algoritmi

Chuqurligi bo`yicha sortirovka qilish ko`rinmaydigan sirtlarni o`chirish uchun qo`llaniladi. Ko`rinmaydigan qirralar algoritmi yashirilgan chiziqlar bilan rasm qurishga imkon beradi, lekin umumiy holda ko`p cheklanishlarga ega. Ko`rinmaydigan qirralar algoritmi ko`p obyektarga ishlatilganda ko`rinmaydigan chiziqlarning atigi 50%ga yaqini o`chiriladi. Bizga shunday algoritm kerakki, u ko`rinmaydigan chiziqlarning hammasini obyektlar soni, ularning qabariqligi va ularda egri chizikli sirtlar borligiga qaramasdan o`chirsin.

Shunday algoritmlarning biri quyidagi tarzda ishlaydi. Har bir obyektning har bir qobirg`asi¹ qandaydir obyektning qirralari² bilan bepkitilganligi tekshiriladi. Tekshirilmagan sirtlar qolmaguncha, qoburg`alarning berk qismlari ketma-ket o`chirilib boradi. Hamma qobirg`alarning qolgan qismlari ekranga chiqariladi.

Algoritmni realizatsiya qilish bir nechta bosqichlardan iborat.

1. Kuzatuvchiga yo`nalgan sirtlar, qolgan boshqa hamma sirtlardan,

ko`rinmaydigan qirralar algoritmi yordamida alohida guruhga ajratiladi. Ajratib olingan sirtlar FACE-TABLE massivida saqlanadi. Kuzatuvchidan yo`nalgan qirralarni hisobga olish talab qilinmaydi, chunki ularning o`zi yashiringan, shu sababli boshqa qirralarning qoburg`alarini berkitishmaydi. Har bir qirra uchun Z_v ning maksimal va minimal qiymati saqlanadi. Egri chiziqli sirtlar (ko`rinmaydigan qirralar algoritmidagi kabi) stiluet chiziqlari bo`yicha ajratiladi, bu sirtlarning ko`rinadigan qismlari esa tekis qirralar bilan birga FACE-TABLE massivida saqlanadi. ploskimi granyami.

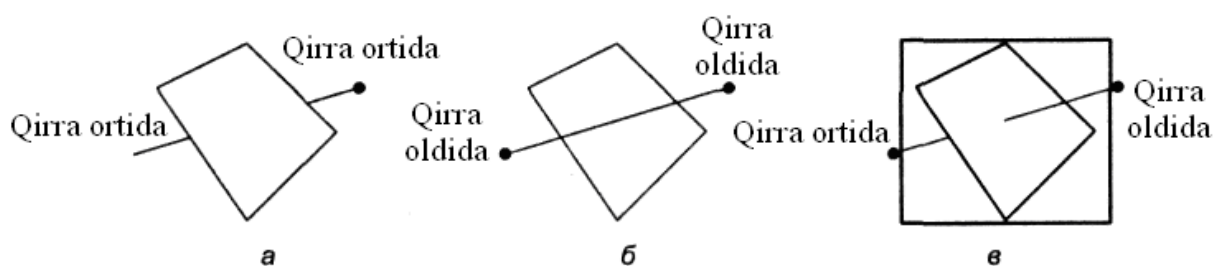
¹ Obyekt ichki hajmini cheklovchi qo`shni sirtlar kesishadigan egri chiziq obyekt qoburg`asi deyiladi.

² Obyekt hajmini cheklovchi sirtlar qirralar deyiladi. Istalgan qirraning yuzasi cheklangan bo`ladi, chunki hamma qirralar qoburg`alar bilan cheklanadi.

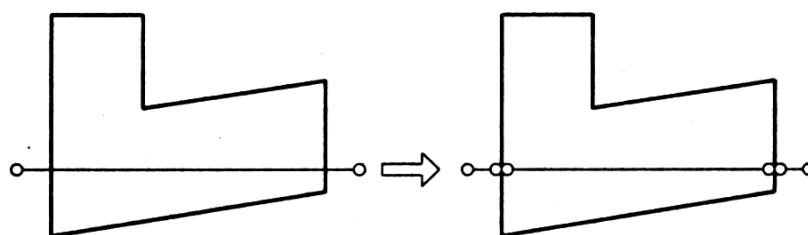
2. Qirralarning qoburg`alari FACE-TABLE massividan qolgan barcha qoburg`alardan ajratib olinadi va alohida ro`yxatga yig`iladi. FACE-TABLEga kirmaydigan boshqa qirralarning qoburg`alarini qaramaslik mumkin, chunki ular ko`rinmaydilar. So`ngra ro`yxatdagi har bir qoburg`aning FACE-TABLEdagi qirra tomonidan berkitilayotganligi tekshirtiriladi.
3. Qoburg`a va qirra Z_v qiymatlarining diapozonlarini taqqoslab, qirra tomonidan berkitilgan qoburg`a aniqlanadi. Bunda uch xil hol bo`lishi mumkin (3.25-rasm). 3.25,a-rasmdagi holda qoburg`aning hamma Z_v qiymatlari qirraning minimal Z_v qiymatlaridan kichik, ya`ni qirra qoburg`a oldida joylashgan. 3.25,b-rasmdagi holda qoburg`aning Z_v qiymatlari qirraning maksimal Z_v qiymatlaridan katta, ya`ni qirra qoburg`a orqasida joylashgan. 3.25,v-rasmdagi holda qirra va qoburg`a Z_v qiymatlarining diapazonlari bir-birini qoplaydi, ya`ni qoburg`aning bir

qismi qirra ortida, ikkinchi qismi esa – uning oldida joylashgan. Agar qoburg`a tekshirilayotgan qirra oldida joylashgan bo`lsa, FACE-TABLE massividan keyingi qirra tanlanadi va qoburg`a endi bu qirra bilan solishtiriladi. Agar qoburg`a qirra ortida bo`lsa yoki uni teshib o`tsa, qo`shimcha amallar bajarishga to`g`ri keladi.

4. Qoburg`a va qirra ekranga proyeksiyalanadi, bundan keyin proyeksiyalarning bir-birini qoplash tekshiriladi. Agar qoplanish bo`lmasa, demak, qoburg`a tekshirilayotgan qirrani berkitmayotgan bo`ladi. FACE-TABLE massividan keyingi qirra tanlanadi va 3-bandga muvofiq tekshiriladi. Agar proyeksiyalar bir-birini qoplasa, qoburg`a, u tekshirilayotgan qirrani teshib o`tayotgan nuqtada, ikki qismga bo`linadi (3.26-rasm). Qoburg`aning berkitilgan qismlari tashlab yuboriladi, ko`rinadigan qismlari esa ro`yxatga qo`shiladi. So`ngra 3-band ro`yxatdagi yangi elementlar uchun takrorlanadi. dastlabki qoburg`a ro`yxatdan o`chiriladi.
5. FACE-TABLEdagi hamma qirralar tekshiruvidan o`tgan qoburg`alar ko`rinadigan hisoblanadi va ekranga chiqariladi.



3.25-rasm. Qirra va qoburg`aning mumkin bo`lgan uch holati



3.26-rasm. Qoburg`ani bo`laklarga bo`lish

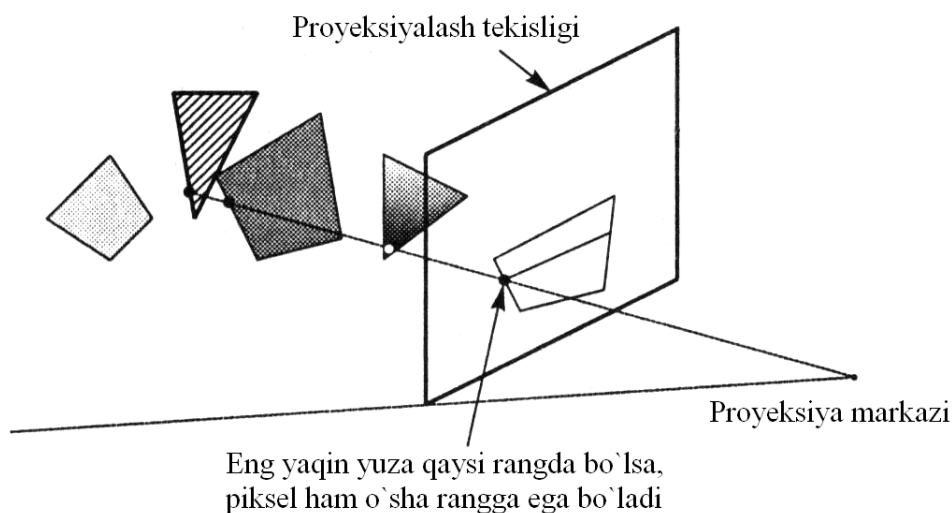
3.8.3. z-bufer metodi

Chuqurlik bo`yicha sortirovka qilish qanday prinsipga asoslangan bo`lsa, z-bufer metodi ham shunday prinsipga asoslangan: qolganlarga nisbatan kuzatuvchiga eng yaqin joylashgan element proyeksiyasi ekranning istalgan uchastkasida paydo bo`ladi. Bu erda element deganda nuqtalar, egri chiziqlar va sirtlar tushuniladi. Bu metod z-bufer deb nomlanadigan xotira jabhasidan foydalanilishini taqazo qiladi. Bu buferda har bir piksel uchun proyeksiyasi ushbu piksel bilan tasvirlanadigan element Z_v qiymati (ya`ni kuzatuvchi tizimida z koordinatasi) – bu obyektning kuzatuvchidan uzoqlashganligini o`lchovidir. z-bufer hajmi piksellar miqdori bilan belgilanadi; piksellarning har biri uchun moddiy son (veshestvennoe chislo)ni saqlash talab qilinadi.

Normalining vektori kuzatuvchidan yo`nalgan qirralar unga ko`rinmaydi, shuning uchun ekranga faqat normalining vektori kuzatuvchiga yo`nalgan qirralargina proyeksiyalanadi xolos. Lekin chuqurligi bo`yicha sortirovka qilish metodidan farqli o`laroq, bu holda proyeksiyalash tartibi ahamiyatga ega emas. Quyida keltirilgan algoritm bayonin o`qiganingizdan keyin buni tushunasiz.

Dastlab ixtiyoriy tanlangan sirt proyeksiyalanadi va sirt proyeksiyasi piksellariga mos keluvchi z-bufer xotirasi yacheykalariga ushbu piksellar probrazlari bo`lgan sirt nuqtalari koordinatalarining qiymatlari Z_v yoziladi. Piksellar birinchi sirt rangiga bo`yaladi. So`ngra keyingi sirt proyeksiyalanadi va uning proyeksiyasiga taalluqli bo`lgan hamma bo`yalmagan piksellar ikkinchi sirt rangiga bo`yaladi. Agar piksellar bo`yalib bo`lgan bo`lsa, ularga mos Z_v qiymatlari joriy sirt nuqtalarining Z_v qiymatlari bilan solishtiriladi. Agar qaysidir pikselning saqlangan qiymati Z_v joriyga nisbatan katta bo`lsa (ya`ni oldingi sirdagi nuqta joriy sirdagi nuqtaga nisbatan kuzatuvchiga yaqinroq bo`lsa), piksel rangi o`zgarmaydi. Aks holda piksel joriy sirt rangiga bo`yaladi. Boshidan z-bufer uzoqdagi tekislikka mos Z_v koordinatalari bilan inisiallanadi (3.11-rasmga qarang), shuning uchun birinchi sirt proyeksiyasining piksellari avtomatik ravishda uning rangiga bo`yaladi. Bu protsedurani mavjud tekisliklarning hammasi uchun

takrorlab, biz ekranning hamma piksellarini eng yaqindagi sirtlar rangiga bo'yaymiz (3.27-rasm).



3.27-rasm. z-bufer metodining asoslari

Yuqoridagi bayonga muvofiq z -bufera metodi, chuqurlik bo'yicha sortirovka qilish metodi kabi, asosan yashiringan sirtlarni o'chirish uchun mo'ljallangan. Lekin z -bufera metodi katta bo'lmagan o'zgartishlar bilan rasmlarni yashiringan chiziqlar bilan qurish imkonini ham beradi. Dastlab hamma sirtlar ekranga proyeksiyalanadi, bunda piksellar fon rangiga bo'yaladi (ekranda hech narsa paydo bo'lmaydi). Bunda z -bufer Z_v ning to'g'ri qiymatlari bilan to'ladi. So'ngra ekranga sirtlar qoburg'alari proyeksiyalanadi. Bunda qoburg'alarining Z_v qiymatlari, oldinga bosqichda topilgan, kuzatuvchiga eng yaqin sirtlarning Z_v qiymatlari bilan taqqoslanadi. Z_v ning yangi qiymatlari dastlabkisidan katta bo'lgan piksellargina bo'yaladi. Shunday qilib, qoburg'alarining sirtlari yashirgan uchastkalari ekranda paydo bo'lmaydi. Bu protsedura yashirilgan chiziqlarsiz to'g'ri rasmni beradi, lekin ba'zi chegaraviy chiziqlar haddan tashqari ingichka bo'lib qolishi mumkin, chunki bu chiziqlarning ba'zi piksellarini mso qoburg'alar tomonidan cheklangan sirtlarga taalluqli bo'ladi. Qoburg'alar proyeksiyalanayotganda obyektning butunicha yaqinroqqa surib, bu muammoni osonlik bilan hal qilishi mumkin.

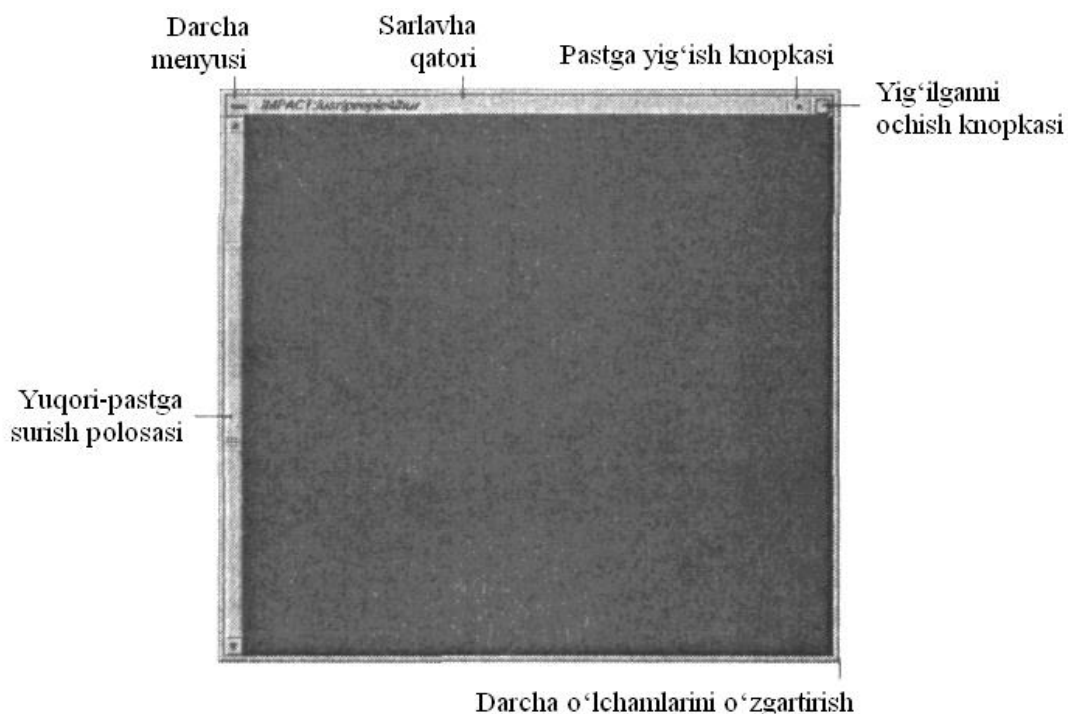
3.9. Foydalanuvchining grafik interfeysi

Ilgari qayd qilinganidek, mavjud *CAD/SAM/SAE* dasturlarining ahamiyatli xususiyati – ularning foydalanuvchi bilan grafik kiritish va grafik chiqarish vositalari orqali muloqotda bo`lishdir. Boshqacha aytganda, dasturaviy ta`minot menyu yoki belgilarni aks ettirish uchun darchani ochish (muloqot jabhasi) hamda menyu punktlari va belgilari qandaydir funksiyalarni etkazish imkoniyatiga ega bo`lishi kerak. Bunday imkoniyatlarni taqdim etuvchi dasturaviy ta`minot *foydalanuvchining grafik interfeysi (graphical user interface – GUI)* deb ataladi. Dasturchi muayyan grafik kutubxona bazasida o`zining qo`l bola grafik interfeysini qurishi mumkin. Bu interfeysning kamchiligi – uni foydalanilgan grafik kutubxonani quvvatlamaydigan ishchi stansiyalarga ko`chirish imkoniyati bo`lmaydi. Shu sababli har bir yangi grafik ishchi stansiyalar uchun interfeysni qaytadan yozib chiqishga to`g`ri keladi. (ushbu bobning keyingi bo`lagi X window tizimiga bog`langan). X window ga asoslangan foydalanuvchining ikkita tipik interfeysi Open Look va OSF/Motif deyiladi. Open Look ni Sun Microsystems korporasiyasi, OSF/Motif ni esa qolgan hamma ishlab chiqaruvchilar quvvatlaydi.

Open Look yoki OSF/ Motif laardan foydalanib ilova yozayotgan dasturchi darchalarni ochish, menyuni qurish va menyu bandlari chiqaradigan topshiriqlarni bajarish kabi operatsiyalarni bajarish uchun darcha dispetcheri funksiyalariga murojaat qilishi mumkin¹. Motif darcha dispetcherining grafik ekrani 3.28-rasmda ko`rsatilgan.

¹ Windows 98/NT boshqaruvida ishlaydigan personal kompyuterlarda shunga o`xshash imkoniyatlarni Microsoft bazaviy klasslari majmui (Microsoft Foundation Classes – MFC) taqdim etadi.

Open Look va Motif larning ilovalarni ishlab chiqish uchun yaxshiligi shundaki, ular X window tizimiga asoslangan. Bu tizimning afzalligi keyingi bo`limda ko`rib chiqiladi.

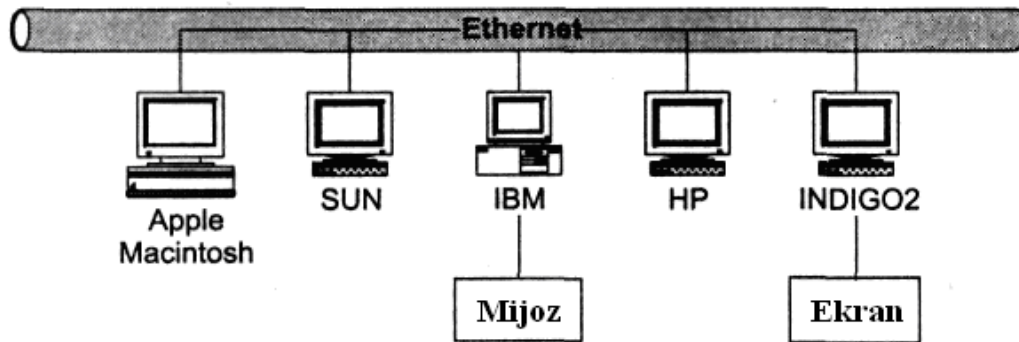


3.28-rasm. Motif dispetcherining darchasi

3.10. X window tizimi

X window (yoki oddiychasiga X) tizimini ishlab chiqish 1983 yilda Massachusetts Texnologiyainstitutida «Afina proyeksi» kod nomi ostida boshlandi. Darcha interfeysi Stenford universitetiga 80-yillarning boshida W nomi ostida yaratilgan operatsion tizim asosida ishlab chiqildi. 1986 yilda X window tizimining birinchi kommersiya versiyasi sotuvga chiqdi, u X10 deb atalardi. Keyinchalik X11R5 chiqarilganligi e`lon qilindi.

X window tizimi tarmoqqa ulangan ishchi stansiyalarda ilovacha darchalarni ochish va yopish imkonini berdi. Kiritish va chiqarish operatsiyalari ham istalgan ishchi stansiyada bajarilishi mumkin edi. «Darcha» so`zi bu bo`limda 3.3-bo`limdagiga nisbatan biroz boshqacharoq ma`noga ega. Bu erda darcha deyilganda ishchi stansiya ekranining alohida bo`lagi tushuniladi, u orqali foydalanuvchi tarmoq hisoblash resurslari bilan muloqotda bo`ladi. Masalan, biz INDIG02 ishchi stansiyasida ikkita darcha ochishimiz mumkin (3.29-rasm) va ularning biridan SUN kompyuterida bajariladigan dasturni kiritish va chiqarish portlari sifatida foydalanishimiz mumkin. Bor vaqtning o`zida INDIG02 ning o`zida bajariladigan dasturni kiritish va chiqarish uchun foydalanish mumkin.



3.29-rasm. Tarmoq muhitida X dan foydalanish

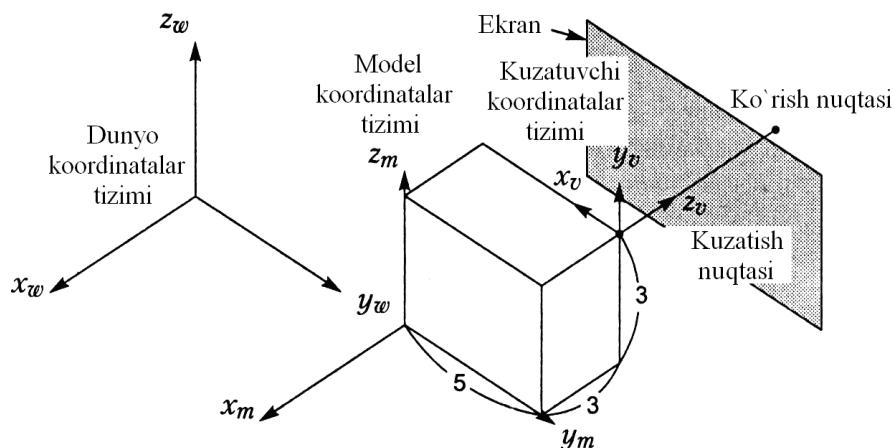
Bunday tasnifdagi masalalarni bajarish uchun X window tizimi klientlardan so`rovlarni qabul qila olishi, boshqa ishchi stansiyalarga so`rovlarni jo`nata olishi va ekran darchalari bilan kiritish-chiqarish grafik operatsiyalarini bajara olishi kerak. Darcha tarmoqdagi istalgan kompyuterda joylashishi mumkin. Darcha ochilishi kerak bo`lgan ishchi stansiyadagi grafika bilan operatsiyalarni bajarish qobiliyatiga ega bo`lgan X window serveri ishlashi lozim. Klient so`rovlari maxsus funksiyalardan foydalanib yoziladi, ular Xlib kutubxonasida saqlanadi. Xlib kutubxonasi ilova bajariladigan ishchi stansiyada o`rnatilishi kerak. So`rovni tarmoq bo`ylab qayta jo`natishni X window tizimining asosiy kodi amalga oshiradi. Umuman, tizimga sanab o`tilgan hamma komponentlar, jumladan, X-server va Xlib kiradi.

Demak, X window uchun ilova quyidagi afzalliklarga ega. Birinchidan, ishchi stansiyalarning birida bajarilayotgan ilova boshqa ishchi stansiyada ochilgan darcha orqali grafik kiritish-chiqarishlarni amalga oshirish mumkin. Ikkinchidan, bir ishchi stansiyada bir nechta darchalar ochilishi mumkin, ular orqali foydalanuvchi har xil mashinalarning hisoblash resurslari bilan ishlash imkoniyatiga ega bo`ladi. Va nihoyat, X window tizimi uchun yozilgan ilova operatsion tizimga va ishchi stansiyalariga bog`liq bo`lmaydi. Foydalanuvchining grafik interfeysi ham shu afzalliklarga ega, u ham ilova hisoblanishi mumkin. Open Look va Motif interfeyslari haqida ham shu gaplarni aytish mumkin.

Savollar va topshiriqlar

1. Qurilma drayveri komandalaridan bevosita foydalanib yozilgan grafik dasturlar qanday kamchiliklarga ega bo`ladi?
2. Tushuntirib bering, nima uchun qaysidir grafik kutubxonaga asoslangan grafik dastur faqat grafik qurilmalarning cheklangan sonidagina bajarilishi mumkin.
3. Grafik qurilmada holatni berish uchun qurilmaning oddiy koordinatalar tizimidan emas, balki virtual tizimdan foydalanishning asosiy sababi nimada?
4. Nima sababdan obyekt shaklini berish uchun ushbu obyektning model koordinatalar tizimidan foyalaniladi?
5. Sahna har bir obyektining holati va oriyentatsiyasi qaysi tarzda berilishini tushuntirib bering.
6. Obyekt nuqtasining koordinatalarini model tizimdan ekran tizimiga o`zgartirish protsedurasini qisqacha bayon qiling.
7. Kompyuter grafikasidagi «Darcha» atamasining ma`nosini tushuntiring.
8. «Ko`rib chiqish darchasi» atamasining ma`nosini tushuntiring.
9. Tanlash va so`rab chiqish rejimlari orasidagi farqni tushuntiring.
10. Ekran fayli bilan qaysi operatsiyalar bajarilishi mumkin?
11. Qurish nuqtasi va kuzatish nuqtasi $(1, 1, 2)$ va $(0, 1, 2)$ dunyo koordinatalariga ega. Vertikal vektori $(0, 0, 1)$ koordinatalarga ega.
 - 1) Ekran, kuzatuvchi koordinatalar tizimi, ko`rish nuqtasi va kuzatish nuqtalarining o`zaro joylashishini ko`rsatuvchi eskizni chizing.
 - 2) Dunyo koordinatalarini kuzatuvchi koordinatalariga o`tkazishni amalga oshiruvchi o`zgartishlar matritsasi T_{w-v} ni yozing.
 - 3) O`zgartishlar matritsasi yordamida dunyo koordinatalari $(5, 1, 2)$ bo`lgan nuqtaning kuzatuvchi koordinatalarini aniqlang.
12. $(2, 2)$ nuqtani $(3, 4)$ nuqtaga nisbatan xu tekisligida soat miliga qarshi yo`nalishda 30° sha burash uchun qaysi o`zgartish matritsalarini qaysi tartibda qo`llanilishi kerak? $(2, 2)$ nuqtaga o`zgartishlar matritsasini qo`llab, buralgandan keyingi nuqtaning koordinatalariga hisoblab toping.

13. Rasmda koordinatar tizimlarining o`zaro holati ko`rsatilgan. Quyida keltirilgan savollarga javob bering.



- 1) Obyekt nuqtasi model tizimida $(-3, 0, 3)$ koordinatalarga ega. Bu nuqtaning dunyo koordinatar tizimidagi koordinatalari (X_w, Y_w, Z_w) , agar o`zgartishlar matritsasi T_t berilgan bo`lsa, quyidagi keltirilgan formuladan topilishi mumkin.

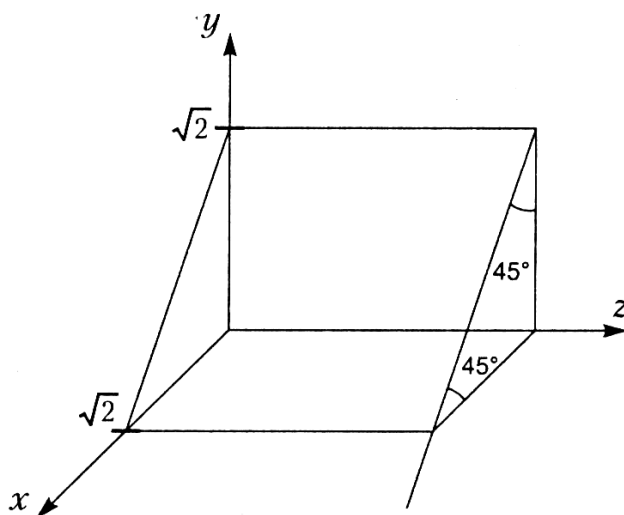
$$\begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} = T_m \begin{bmatrix} -3 \\ 0 \\ 3 \\ 1 \end{bmatrix}.$$

Keltirilgan formula bo`yicha o`zgartishlar matritsasi T_t ni hisoblab toping va nuqtaning dunyoviy koordinatarini aniqlang. Bu masalada, agar dunyo koordinatar tizimi $(0, 2, -1)$ vektorga translyatsiya qilinsa, model koordinatar tizimi dunyo tizimi bilan ustma-ust tushadi.

- 2) Kuzatish koordinatar tizimi rasmda ko`rish nuqtasi $(-10, 7, 2)$, kuzatish nuqtasi $(-3, 7, 2)$ va vertikal $(0, 0, 1)$ koordinatalari bilan tavsiflanadi; bular dunyo koordinatar tizimida berilgan. Nuqtaning dunyo koordinatalari (X_w, Y_w, Z_w) ni kuzatuvchi (X_v, Y_v, Z_v) ga o`tkazuvchi o`zgartishlar matritsasini hisoblang.

$$\begin{bmatrix} X_v \\ Y_v \\ Z_v \\ 1 \end{bmatrix} = T_v \cdot \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}$$

14. Dunyo koordinatalar tizimidan kuzatuvchi tizimga o'tishni bajaruvchi o'zgartishlar matritsa T_{w-v} ni hisoblang; boshlang'ich shart: ko'rish nuqtasi va kuzatish nuqtasi koordinatalari (dunyo tizimida) mos ravishda (4, 5, 6) va (0, 0, 0) ga teng. Dunyo koordinatalar tizimining z o'qi vertikal vektori bilan ustma-ust tushgan.
15. Cho'qqilarining koordinatalari $A(0, 0, 0)$, $V(1, 1, 0)$ va $S(0, 1, 2)$ bo'lgan qabariq obyektning uch burchakli qirrasini tasavvur qiling. Ko'rinmaydigan qirralarni yo'qotish algoritmidan foydalanib, ushbu qirra ko'rish nuqtasi $V(0, 1, 5)$ dan ko'rinish-ko'rinmasligini aniqlang. $D(2, 2, 2)$ – ushbu obyekt cho'qqilaridan biridir, deb tasavvur qiling.
16. Ikki o'lchamli obyektning A va V nuqtalari S va D nuqtalarga siljimoqda, bu boshlang'ich shakl o'zgarishga olib keladi. Jismning hamma nuqtalariga qo'llanilishi lozim bo'lgan o'zgartishlar matritsalarini to'g'ri tartibda sanab chiqing. Nuqtalar koordinatalarining qiymatlari: $A(2, 2)$, $B(5, 5)$, $S(5, 2)$, $D(7, 2 + \sqrt{3})$.
17. xz tekisligiga perpendikulyar bo'lgan tekislik quyida keltirilgan rasmda ko'rsatilganidek joylashgan.



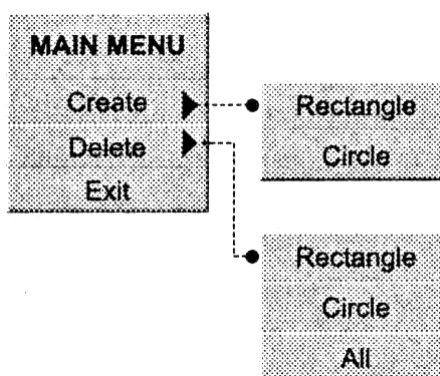
R nuqtaning aks tasviri R^* ni bu tekislikka nisbatan quyidagi keltirilgan formula bo'yicha hisoblab topish mumkin. T_{r-r^*} elementlar o'zgartishlar matritsasi orqali ifodalagsin, ularga $T_{Trans}(a, b, c)$, $Rot(x, \alpha)$, $Rot(y, \beta)$, $Rot(z, \gamma)$ lar hamda xz , yz va xz tekisliklariga nisbatan aks ettirish matritsalarini kiradi.

$$[X^* \ Y^* \ Z^* \ 1] = T_{p-p^*} \cdot [X \ Y \ Z \ 1].$$

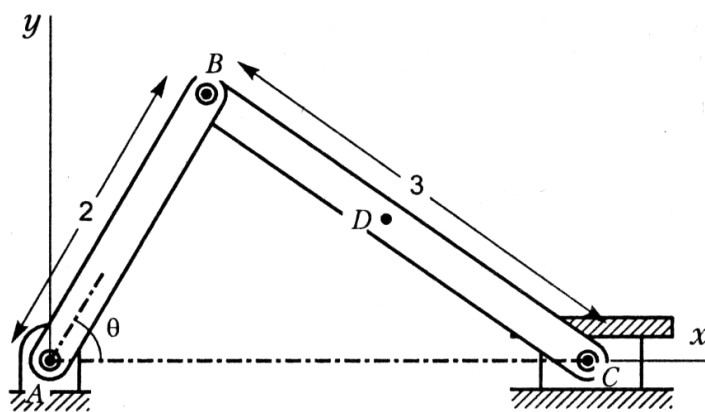
18. Istalgan grafik kutubxonadan foydalanib, quyidagi amallarni bajaradigan grafik dasturni yozing.

- 1) Koordinatalar tizimini va uning markazida kubni aks ettiring. Ko'rish va kuzatish nuqtalarining holatini, vertikal vektor yo'nalishini va kub o'lchamini o'zingiz mustaqil ravishda bering.
- 2) Sichqon, chap knopkasi bosilganda kub u yo'nalishida +5 birlikka, z yo'nalishida +5 birlikka translyatsiya qilinadi. O'rta knopka bosilganda boshlang'ich holatga (koordinatalar tizimi markazidagi) qaytadi.

19. Quyida keltirilgan suzib chiquvchi (vsplivayushiy) menyu bilan ishlaydigan ikki o'lchamli grafik redaktorni yozing.



20. Etaklovchi zveno AV ta'siri ostida krivoship-shatunli mexanizm aylanganda shatun VS ning o'rtasidagi nuqta D trayektoriyasini chizuvchi grafik dasturni yozing. (Quyida keltirilgan rasmga qarang). Qurilgan trayektoriyadan etaklovchi zvenoning shunday θ burchagini aniqlangki, unda trayektoriyaga urinma gorizontali bo'lib qolsin.



21. Ilovaning grafik interfeysi X window tizimida yozilganda olinadigan afzalliklarni bayon qiling.

4 – BOB. CHIZMALARNI AVTOMATLASHTIRILGAN ISHLAB CHIQUISH TIZIMLARI

Yuqoridagi boblarda qayd qilinganidek, *chizmalarni avtomatlashtirilgan ishlab chiqish tizimi (computer – aided drafting system)* – bu ishlab chiquvchiga interaktiv rejimda mashinasozlik, arxitektura, muhandislik chizmalarini, elektr sxemalarini va yana ko`p boshqa turdagi chizmalarni yaratish va o`zgartirish imkonini beruvchi dasturaviy mahsulotdir. Bu dastur, bundan tashqari, tayyor chizmalarni va ularning o`zgarishlarini saqlab, ma`lumotlar bazasini yangilab turadi. Demak, chizmalarni avtomatlashtirilgan ishlab chiqish tizimi bilan ishlash matn protsessoridan foydalanishga o`xshaydi. Yagona farq shundaki, chiqishda foydalanuvchi matnli xujjat emas, balki chizma oladi. Matn protsessorida mavjud xujjat bazasida yangi xujjatni juda tez tayyorlash mumkin bo`lgani kabi, chizmalarni avtomatlashtirilgan ishlab chiqish tizimida ham mavjud chizmani o`zgartirib, yangi chizmani olish mumkin. Matn protsessori yoki chizmalarni avtomatlashtirilgan ishlab chiqish tizimining yangi xujjat yoki chizmani tayyorlashdagi afzalliklari beqiyosdir. Lekin mavjud xujjatlar va chizmalarni o`zgartirishda ularning afzalliklari yanada yorqinroq namoyon bo`ladi.

Keyingi bo`limlarda chizmalarni avtomatlashtirilgan ishlab chiqish tizimlarining ko`pida mavjud bo`lgan eng tipik funksiyalarni ko`rib chiqamiz. Har bir tizimda funksiyalarni chaqirishning muayyan komandalari o`ziniki bo`lishi mumkin, shu sababli zarurat tug`ilganda Sizga mos tizimning foydalanuvchi qo`llanmasiga murojaat qilishga to`g`ri keladi.

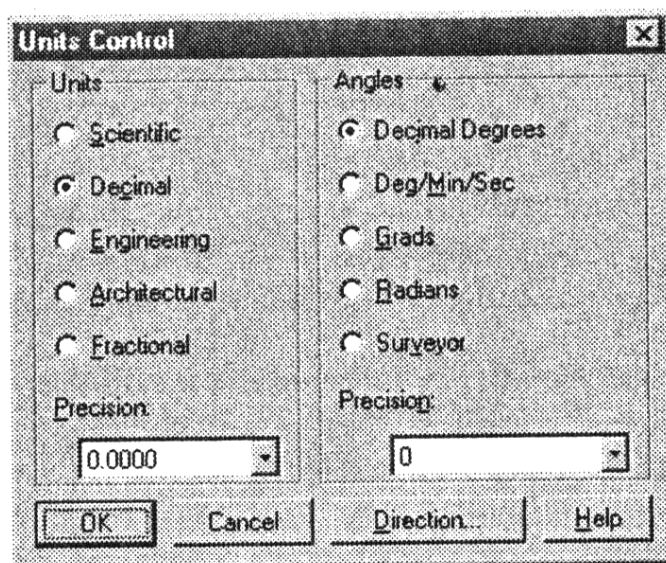
4.1. Chizma parametrlarini o`rnatish

Chizmalarni avtomatlashtirilgan ishlab chiqish tizimlari bilan ishlashni o`lchash birliklari, chizma o`lchamlari, to`r (setka) va qatlam (sloy)Lar parametrlarini o`rnatishdan boshlash lozim. Chizmalarni tez va aniq qurish uchun, bu parametrlar to`g`ri qiymatlarga ega bo`lishi kerak. Chizmani to`rsiz va

qatlamlarsiz ham qurish mumkin, lekin bunga ko`p vaqt ketadi, olingan chizmani o`zgartirish esa juda qiyin bo`ladi.

4.1.1. O`lchash birliklari

Foydalanuvchi masofalar va burchaklarni o`lchashning format va birliklar aniqligini tanlab olishi kerak. Masofalarni o`lchash birliklari ilmiy, o`nli, kasrli, muhandis va arxitektura birliklarida taqdim etilishi mumkin. Burchaklarni o`lchash birliklari – bu graduslar, graduslar/minutlar/sekundlar, radianlar va geometrik birliklardir. 4.1-rasmda chizmalarni avtomatlashtirilgan ishlab chiqishning eng tarqalgan tizimlaridan biri – AutoCAD Release 14 dasturining birliklarni tanlash dialogli darchasi ko`rsatilgan.



4.1-rasm. AutoCAD Release 14 da o`lchash birliklarini o`rnatish dialogli darchasi

4.1.2. Chizma o`lchamlari

Chizmani qog`ozda chizayotganda, Siz varaq chegaralaridan tashqariga chiqqa olmaysiz. Xuddi shu kabi grafik qurilma bilan ishlaganda ham chizma ma`lum chegaralarga ega bo`lishi kerak, chunki bu chizma qachondir baribir cheklangan o`lchamli qog`ozga chiziladi. Demak, foydalanuvchi oldindan chizma o`lchamlarini o`rnatishi kerak. Bu operatsiyani bajaruvchi komandalar ketma-ketligi AutoCAD uchun quyida keltirilgan.

Command: limits

Reset Model Space Limits

ON/OFF/<Lower left corner><0.00.0.00>:10.10

Upper right corner<12.00.9.00>:300.200

Chizma o`lchamlarini tanlashda odatda quyidagi omillar hisobga olinadi [15]:

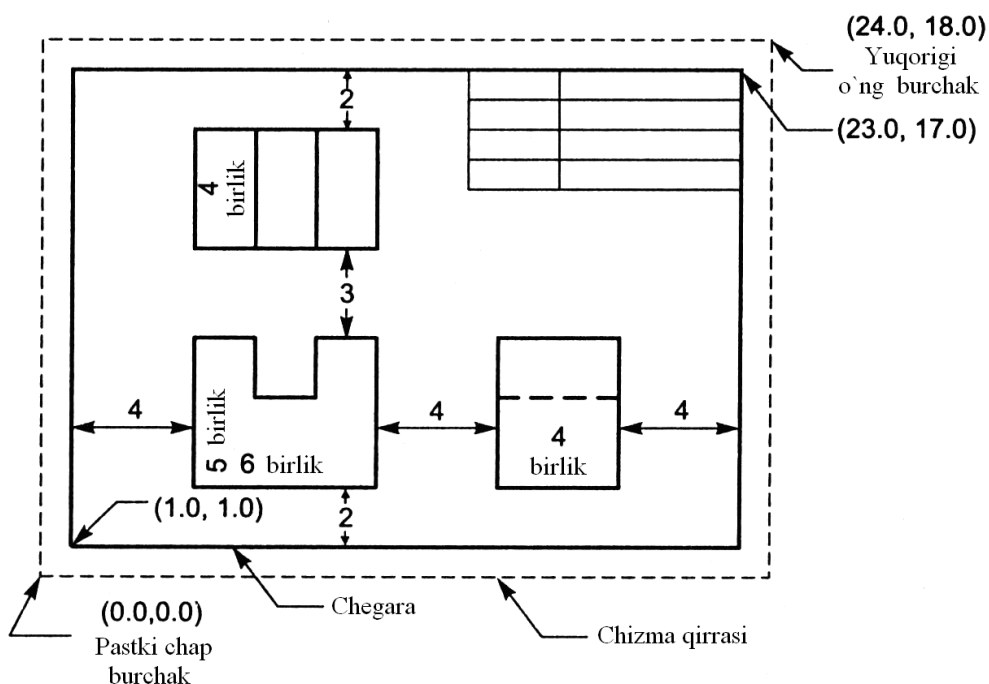
- chizmaning amaldagi (fakticheskiy) o`lchami;
- o`lchamlar, izohlar, materiallar ro`yxatlari va boshqa zarur ma`lumotlarni belgilash uchun joy;
- turli ko`rinishlar orasidagi masofa (chizma tiqilib ketgandek ko`rinmasligi lozim);
- ramka va sarlavha (agar ular nazarda tutilgan bo`lsa) uchun joy.

Chizma o`lchamlarini berishdan oldin, zarur bo`lgan joyni birinchi yaqinlashuvda aniqlash uchun, chizma eskizini qurish tavsiya etiladi. Masalan, agar qandaydir obyekt uchun oldidan ko`rinish o`lchamlari 6x5 birlikka, yon tomondan ko`rinish o`lchamlari 4x5 birlikka va yuqoridan ko`rinish o`lchamlari 6x4 birlikka teng bo`lsa, cheklashlarni shunday o`rnatish kerakki, hamma chizma va unga taalluqli bo`lgan hamma ma`lumotlar formatga sig`sin. Faraz qilaylik, Siz oldidan va yon tomondan ko`rinishlar orasidagi masofa 4 birlikni, oldidan va yuqoridan ko`rinishlar orasidagi masofa 3 birlikni tashkil etishini istayapsiz. Chizma chegaralarigacha masofa chapdan – 4 birlikni, o`ngdan – 4, yuqoridan – 2 va pastdan – 2 birliklarni tashkil qilsin (4.2-rasm). Ko`rsatilgan qiymatlar shunday tanlanadiki, tayyor chizma garmonik ko`rinishga ega bo`lsin.

O`zingiz uchun ko`rinishlar o`lchamlari va ular orasidagi masofalar hamda chizma chegaralari va bu chegaralardan qog`oz chetigacha bo`lgan masofalarni belgilaganingizdan keyin, siz chizma o`lchamlarini quyidagicha hisoblab topishingiz mumkin:

- gorizontaal bo`yicha o`lcham = $1+4 + 6 + 4 + 4 + 4 + 1 = 24$;
- vertikal bo`yicha o`lcham = $1 + 2 + 5 + 3 + 4 + 2 + 1 = 18$.

Demak, chizma o`lchamlari 24x18 birlikni tashkil qiladi. E`tibor bering, biz chizma chegaralari va qog`oz chegaralari orasidagi masofani bir birlikka teng qilib oldik.



4.2-rasm. Chizma o`lchamlarini berish

4.2-rasmdagi chizma 1:1 masshtabda bajarilgan. Lekin, agar siz chizmaning qattiq nusxasini olish uchun chizmani chop qilishni istasangiz, sizga qog`oz o`lchamlariga qarab uni kattalashtirish yoki kichiklashtirishga to`g`ri keladi. Umuman, varaq o`lchami chizma chegaralarini, shrift o`lchamini, chizma masshtabini, chiziq qalinligi masshtabini va chizmaning boshqa parametrlarini belgilaydi. Varaqning standart o`lchamlari har xil masshtablardagi chizmalarning mos chegaralari (millimetrlarda) 4.1-jadvalda keltirilgan.

4.1-javdal

Birliklarning metrik tizimi

| Varaq o`lchami | Qog`oz o`lchami | Chizma chegarasi (M1:1) | Chizma chegarasi (M1:5) | Chizma chegarasi (M5:1) |
|----------------|-----------------|-------------------------|-------------------------|-------------------------|
| A4 | 210x297 | 210, 297 | 1050, 1485 | 42, 59,4 |
| A3 | 297x420 | 297, 420 | 1485, 2100 | 59,4 84 |
| A2 | 420x594 | 420, 594 | 2100, 2970 | 84, 118,8 |
| A1 | 594x841 | 594, 841 | 2970, 4205 | 118,8, 168,2 |

| | | | | |
|----|----------|-----------|------------|--------------|
| A0 | 841x1189 | 841, 1189 | 4205, 5945 | 168,2, 237,8 |
|----|----------|-----------|------------|--------------|

4.1.3. Qatlam

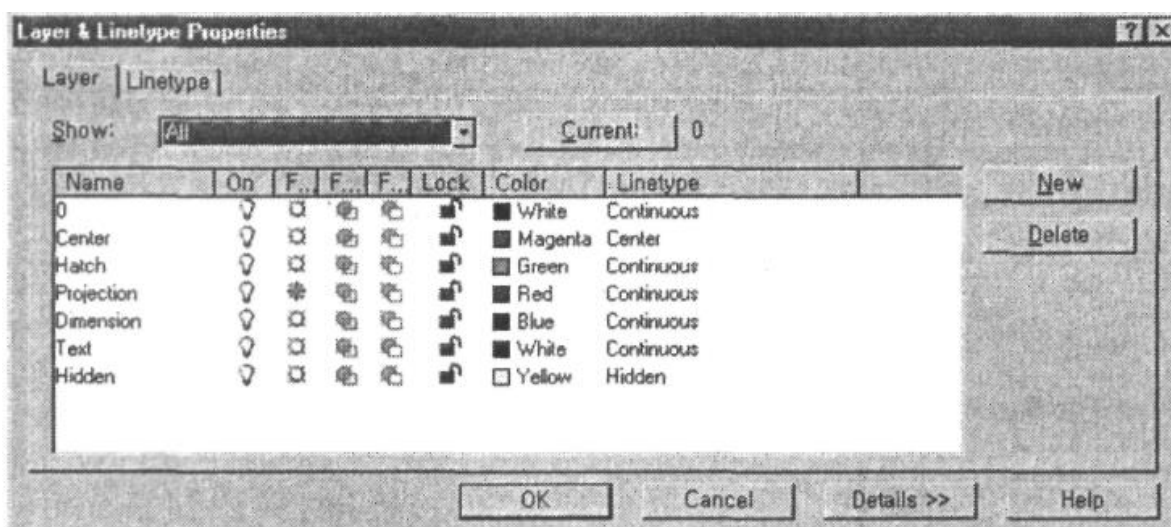
Agar chizma yetarli darajada murakkab bo`lsa, uni ko`p qatlamlarga bo`lish qulaylik keltiradi. Binoning qavatma-qavat planini va truboprovodlar sxemasini alohida qatlamlarga taqsimlab, siz vazifangizni ancha soddalashtirasiz. Boshqacha aytganda, hamma obyektlarni o`z ichiga olgan katta chizmada hamma operatsiyalarni bajarishga qaraganda, bu operatsiyalarni alohida qatlam bilan bajarish ancha oson bo`ladi. Lekin sizda turli qatlamlardagi elementlar (masalan, trubalar va bino devorlari)ning o`zaro joylashishi haqida tasavvur hosil qilish uchun ko`rib chiqish rejimlari orasida hamma chizmani butunlay ko`rish imkoniyati bo`lishi lozim. Qatlamlarga bo`lish ularni bir-birining ustiga tushirish imkoniyatini saqlaydi, bunda grafik operatsiyalar bajarilayotganda chizma murakkablashib ketmaydi. Siz ushbu onda ishlayotgan qatlam aktiv hisoblanadi, qolgan hamma qatlamlar ushbu onda aktiv hisoblanmaydi. Istalgan boshqa fon kabi, aktiv bo`lmagan qatlamlarning grafik elementlari tanlash va o`chirish kabi grafik operatsiyalarga sezgir bo`lmaydi. Shu sababli chizma murakkabligi, go`yo siz bitta-yagona aktiv qatlam bilan ishlaganingiz kabi, qoladi.

Ko`p qatlamli pechatli plitalar alohida qatlamlarning chizmalarini qurishda qatlamlarga bo`lish funksiyasidan samarali foydalanish mumkin. Bu holda har bir qatlam, boshqa qatlamlarga bog`liq bo`lmagan holda, qurilishi mumkin, lekin qatlamlarning bir-biriga nisbatan holati haqida ma`lumot olish uchun bir-biri bilan qaysidir tarzda bog`langan qatlamlarni ekranga chiqarish imkoni qoladi. Qatlamlarga bo`lish konstruksiya alohida detallarining chizmalarini qurish uchun ham qulay. Agar har bir detal yig`ma chizmaning alohida qatlamida chizilsa, kerakli qatlamni aktivlashtirib, istalgan detal chizmasini osonlikcha olish mumkin.

4.3-rasmda AutoCAD Release 14 da qatlamlarni boshqarish dialog darchasi ko`rsatilgan.

4.1.4. To`r va bog`lash

Chizma qog`ozda chizilganida yordamchi chiziqlardan keng foydalaniladi, ular oldindan reysshina yordamida chizib olinadi. Ular chizma chiziqlarini qurishni qurishni osonlashtiradi va ularni ancha aniqroq qiladi. Chizmalarni avtomatlashtirilgan tarzda ishlab chiqish tizimlarida to`r chiziqlari ham shu vazifani bajaradi. To`rning gorizontal va vertikal chiziqlari berilgan zichlik (razreshenie)da bir-biridan teng masofalarda chiziladi, chizma chiziqlari esa ularning ustidan chiziladi. Chizmalarni avtomatlashtirilgan tarzda ishlab chiqish tizimlarining ba`zilarida faqat to`r chiziqlari kesishadigan joylarda nuqtalar quriladi.



4.3-rasm. AutoCAD Release 14 da qatlamlarni boshqaruvchi dialogli darcha

To`r chizig`i ustidan to`g`ri chiziq o`tkazish uchun, uning ikki oxirining holatini berish kerak. Ularning koordinatalarini klaviaturadan kiritish yoki kursorni kerakli joyga o`rnatib va knopkani bosib sichqon bilan ko`rsatish mumkin. Eslab ko`ring (3-bobga qara), sichqon lokator rejimida bo`lganida, kursor uning harakatini kuzatib boradi. Nuqtaning ikkinchi usulda ko`rsatilgan holati inson qo`li titrashi va sichqon mexanizmi noaniqligi tufayli yetarli darajada aniq bo`lmasligi mumkin. Bu muammoni hal qilish uchun kursor bog`lanishni to`r chiziqlarining eng yaqindagi kesishish nuqtasiga ulash mumkin. Sichqon knopkasi bosilganda kompyuter ushbu kesishish nuqtasining aniq koordinatalarini qabul qiladi.

Koordinatalar berilishi aniqligini to`r ruxsati (разрешение) aniqlaydi, uni esa foydalanuvchi o`zi istaganicha o`rnatishi mumkin. Bu funksiya *bog`lash* (*snapping*) deyiladi. To`rni ulash komandasi AutoCAD dasturida quyidagi ko`rinishga ega.

Command: grid

Grid spacing(X) or ON/OFF/Snap/Aspect<0>:0.75 /* rasstoyanie mejdu liniyami setki ustanavlivaetsya ravnim 0.75 ekrannoy edinisi */

4.2. Chizmachilikning bazaviy funksiyalari

4.2.1. To`g`ri chiziq

Chizmalarni avtomatlashtirilgan ishlab chiqish tizimlarida kesmalarni qurishning ko`p usullari mavjud. Ularning ichida eng ko`p tarqalgani – ikki oxirgi nuqtalar bo`yicha qurishdir. Nuqta holati har xil usulda berilishi mumkin. Oldingi bo`limda biz ikki metodni: koordinatalarni klaviaturadan kiritishni va sichqon knopkasini lokator rejimida bosishni taklif qildik. Bundan tashqari siz ekranda mavjud bo`lgan nuqtalardan birini tanlab, kesmaning oxirgi nuqtasini ko`rsatishingiz mumkin.

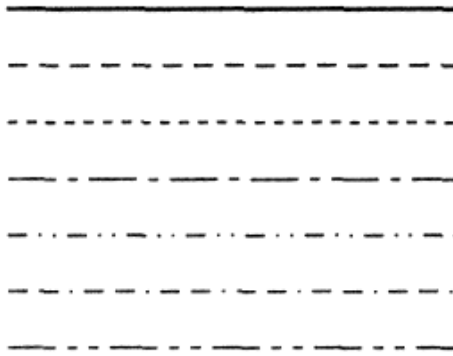
Kesmani uning ikkala oxirlarini ochiq ko`rsatmasdan ham qurish mumkin. Usullardan biri – ko`rsatilgan nuqtadan mavjud egri chiziqqa urinma o`tkazishni tizimdan so`rashdir. Bu holda faqat bitta nuqta aniq ko`rsatiladi, ikkinchi nuqtani esa tizim mustaqil ravishda aniqlaydi. Chiziq atributlari sifatida uning turi va qalinligi ko`rsatilishi mumkin. Chizmalarni avtomatlashtirilgan ishlab chiqish tizimlarining ko`pchiligi tomonidan quvvatlanadigan chiziqlarning turlari 4.4-rasmda keltirilgan. AutoCAD Release 14 da kesmani qurish quyidagi tarzda bajariladi:

Command: line

From point: 1. 1

To point: 5. 2

To point: return



4.4-rasm. Chiziqning har xil turlari

4.2.2. Aylana va aylana yoyi

Aylanani berishning eng oddiy usuli – uning markazi va radiusi uzunligini ko`rsatishdir. Boshqa usul – aylananing o`zida uchta nuqta berishdir. Chizmalarni avtomatlashtirilgan ishlab chiqish tizimlarining ko`pchiligi aylanani boshqa usullar bilan yaratish imkonini ham beradi. Masalan, tizim ikkita to`g`ri chiziqqa urinma bo`lgan aylanani chizishi mumkin. Istalgan holda sizga mos obyektlarni tanlab olishga to`g`ri keladi.

Aylana yoyi – bu aylananing xususiy holatidir, u (oddiy aylanaga beriladigan parametrlardan tashqari) boshlanish va tugash nuqtalarini berish bilan aniqlanadi. AutoCAD Release 14 da aylana quyidagi tarzda quriladi:

Command: circle

3P/2P/TTR/<Center point>: 5. 5

Diameter7<Radius><current>: 3

Aylana yoyi quyida ko`rsatilganidek quriladi. Bizning holda yoy berilgan uchta nuqtadan o`tadi.

Command: arc

Center/<Start point>: 7. 4

Center7End/<Second point>: 6. 5

End point: 6. 3

4.2.3. Splayn

Splaynlar ixtiyoriy egri chiziqlarni chizish uchun ishlatiladi, chizmachilikda bu amal lekalo yordamida qo`lda bajariladi. Foydalanuvchi egri chiziqda nuqtalarni ko`rsatadi, tizim esa ushbu nuqtalardan o`tadigan interpolyasion egri chiziqni quradi. Hosil bo`lgan egri chiziq odatda uchinchi tartibli tenglama bilan taqdim etiladi. Ba`zan egri chiziqlar berilgan nuqtalar bo`yicha qurilishi mumkin, ular egri chiziqni aniqlaydi, lekin unda yotishi shart emas.

4.2.4. O`chirish

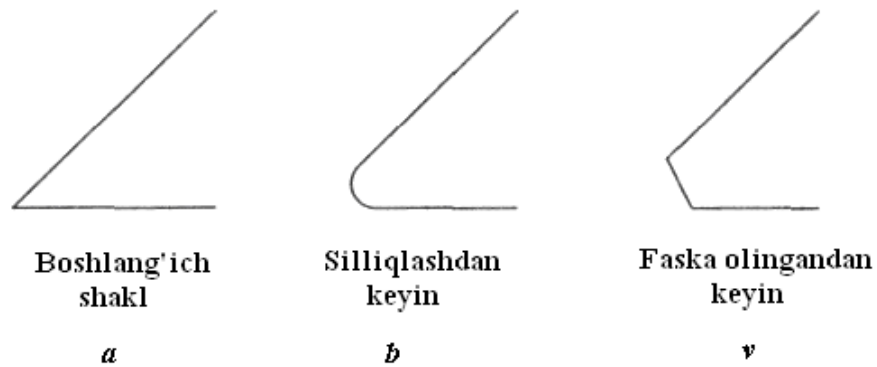
O`chirish funksiyasi chizmachilikda qog`ozda o`chiradigan rezinka kabi ishlaydi. Siz nuqta, kesma va egri chiziq kabi grafik elementlarni tanlaganingizda, ular ekrandan yo`qoladi. Tanlash rejimi 3-bobda bayon qilingan edi.

4.2.5. Silliqlash va faskalarni olish

Silliqlash va yumaloqlash (filleting, rounding) – bu ikki kesishuvchi kesma orasida aylana yoyini qurishdir (4.5, *a*-rasm), bunda qurilgan yoy ikkala kesmaga urinma bo`ladi (4.5, *b*-rasm). Silliqlash botiq burchaklar uchun, yumaloqlash esa – qabariq burchaklar uchun ishlatiladi. *Faskalarni olish (chamfering)* – silliqlash kabi, lekin bunda yoy o`rniga to`g`ri chiziqli kesma quriladi (4.5, *v*-rasm).

Silliqlash va faskalarni olish quyidagi tartibda bajariladi.

1. Silliqlash radiusi yoki faska o`lchami ko`rsatiladi.
2. Kesishuvchi ikki kesma tanlanadi. Silliqlash yoki faska kesishish nuqtasi yaqinida quriladi.
3. Silliqlash yoki faska qurilgandan keyin boshlang`ich kesmalarning kerakmas qismlari o`chiriladi. Ba`zi tizimlarda o`chirish avtomatik bajariladi, boshqalarida esa buni qo`lda qilishga to`g`ri keladi.



4.5-rasm. Silliqlash va faska olish

AutoCAD da silliqlashni quyidagi tarzda bajarish mumkin:

Command: fillet

(TRIM mode) Current fillet radius = 10.00

Polyline/Radius/Trim/<Select first object>: g

Enter fillet radius<current>: 3

(TRIM mode) Current fillet radius - 3.00

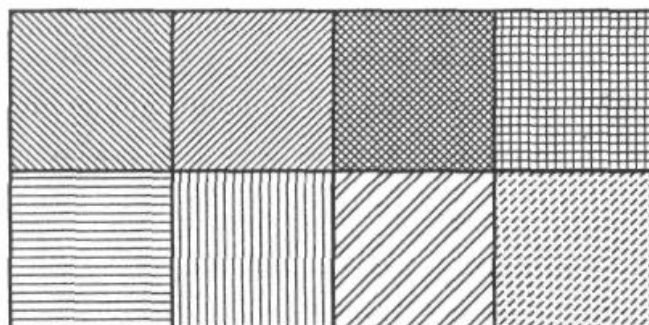
Command: fillet

Polyline/Radius/Trim/<Select first object>:/*Vibiraetsya perviy otrezok.*/*

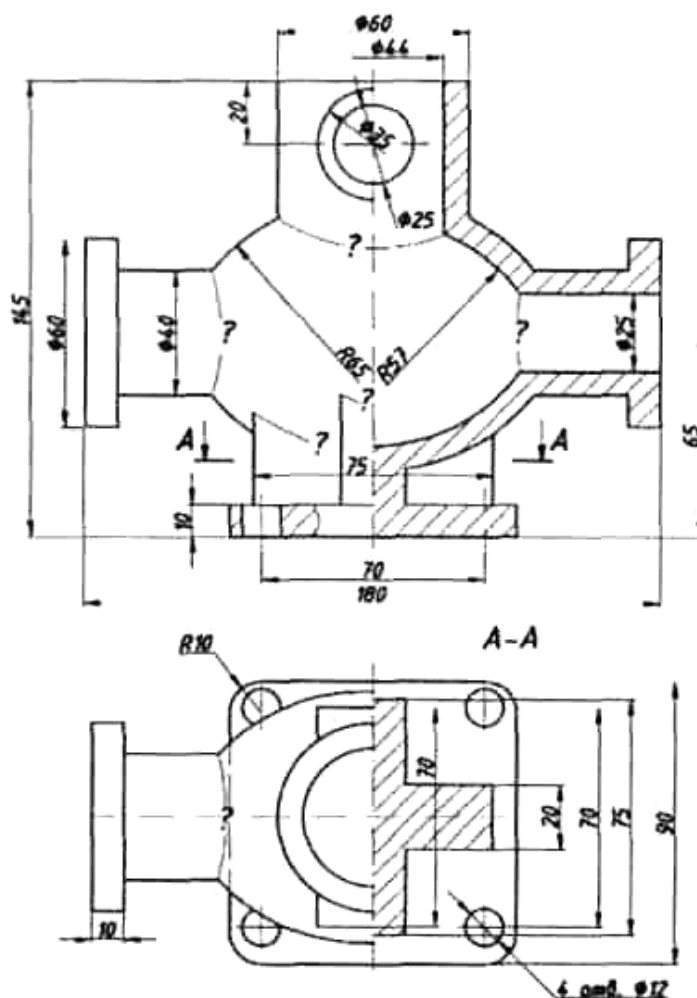
Select second line: /* Vibiraetsya vtoroy otrezok. */

4.2.6. Shtrixlash

Shtrixlash deb berk ko`pburchakli qandaydir shablon bilan to`ldirishga aytiladi. Shtrixlash mashinasozlik chizmalarida kesimlarni belgilash va arxitektura chizmalarida turli materiallarni ajratish uchun ishlatiladi. Chizmalarni avtomatlashtirilgan ishlab chiqish tizimlarining ko`pida taqdim etiladigan ba`zi eng tipik shablonlar 4.6-rasmda keltirilgan.



4.6-rasm. Shtrixlash namunalari



4.7-rasm. O'lchamlari belgilangan, shtrixlangan va ilovali chizma

Shtrixlash berk ko'pburchaklarni ko'rsatishdan boshlanadi. Bu operatsiya har xil yo'sinda bajarilishi mumkin. Ba'zi tizimlarda ko'pburchakni tashkil qiluvchi hamma kesmalarni ko'rsatib chiqishga to'g'ri keladi. Boshqa tizimlarda kesmalardan birini ko'rsatish kifoya qiladi, qolganlarining hammasini tizim avtomatik topadi. Agar ko'pburchak ichida shtrixlanmaydigan uchastkalar bo'lsa, uning chegaralari ko'rsatilishi kerak. Shtrixlash – bu chizmachining unumdorligini oshiruvchi chizmalarni avtomatlashtirilgan ishlab chiqish tizimining funksiyalaridan biridir. 4.7-rasmda kesimlarning elementlari shtrixlangan chizma ko'rsatilgan.

4.3. Yordamchi funksiyalar

4.3.1. Kopyialash

Matn protsessorida «Kesib olinsin» va «O`rniga qo`yilsin» operatsiyalari qanday ishlasa, kopyialash funksiyasi ham shunday ishlaydi. Grafik elementlar majmui ajratilishi, buferda saqlanishi va o`sha yoki istalgan boshqa chizmaning istalgan joyiga kiritib o`rnatilishi mumkin. Grafik elementlarni tanlash ularni kerakli o`lchamdagi to`g`ri burchakli to`rtburchak bilan atrofini chizib chiqish yo`li bilan amalga oshiriladi. Ko`rib chiqish darchasi berilishida ekranda to`g`ri burchakli to`rtburchak qanday chizilsa, bu holda ham shunday chiziladi. To`rtburchak chegaralari kesib o`tadigan grafik elementlar, foydalanuvchi istagiga ko`ra kopiya olinayotgan majmuiga kiritilishi yoki undan o`chirilishi mumkin. Tanlab olingan obyektlar qaerga o`rnatilishi kerak bo`lsa, kursor o`sha nuqtaga o`rnatiladi. Agar chizmada takrorlanadigan elementlar, masalan ko`p xonali uylarning arxitektura chizmalari kabi chizmalarda mavjud bo`lsa, kopyialash funksiyasi qulaylik yaratadi. Kopyialash detallar chizmalarini ishlab chiqishda ham foydalidir, chunki siz butun qurilma chizmasining bir qismini kopyialashingiz va bundan keyin hosil bo`lgan chizmani aniqlashtirishingiz mumkin.

Aks tasvir kopyialashning xususiy holati bo`ladi, u o`q simmetriyasiga ega bo`lgan shakllarni qurish imkoniyatini beradi. bir yoki bir nechta simmetriya o`qlariga ega bo`lgan obyektlarni qurishda bu funksiya foyda beradi. chizmalarni avtomatlashtirilgan ishlab chiqish tizimlarining ko`pi takrorlanadigan obyektlarni tartib bilan joylashtiradigan qo`shimcha funksiyalarni taqdim etadi. Masalan, ba`zi tizimlar aylana bo`yicha ma`lum qadamda joylashgan boltlarning bir nechta kallaklarini chizishi mumkin, buning uchun siz dastlab ulardan birini chizishingiz va kerakli parametrlarni berishingiz kerak.

AutoCAD Release 14 da kopyialash funksiyasi quyidagi tarzda chaqiriladi.

Command: copy

Select objects: /* Vibiraetsya nabor graficheskix elementov. */

<Base point or displacement/Multiple: /* Ukazivaetsya pervaya tochka. */

Second point of displacement: /* Ukazivaetsya vtoraya tochka vektora smesheniya. */

4.3.2. Darcha

Murakkab chizma bilan ishlaganda ba`zan uning bir qismini kattalashtirish zarurati tug`ilishi mumkin. Ko`pincha zarur bo`lgan grafik elementni, agar u qo`shnilariga qo`shilib ketgan bo`lsa, ajratib olish qiyin bo`ladi. Bu muammoni, chizmaning sizni qiziqtirayotgan elementi joylashgan qismini kattalashtirib, yechish mumkin. O`sha o`lchamdagi ko`rib chiqish darchasiga aks ettiriluvchi kichik darchadan foydalanish, grafik elementlar raqamli parametrlarini o`zgartirmasdan, kattalashtirish effektini beradi. kopyalanuvchi jabha qanday aniqlansa (oldingi bo`limga qara), darcha ham shunday – uning diagonalining ikkita oxirgi nuqtalarini berish bilan aniqlanadi.

AutoCAD Release 14 da darcha quyidagi keltirilgan komanda bilan o`zgartiriladi.

Command: zoom

All/Center/Dynamic/Extents/Left/Previous/Vmax/Window/<Scale(X/Xp)>:w

First corner: /* Ukazivaetsya pervaya tochka diagonali. */

Second corner: /* Ukazivaetsya vtoraya tochka diagonali. */

4.3.3. Simvollar

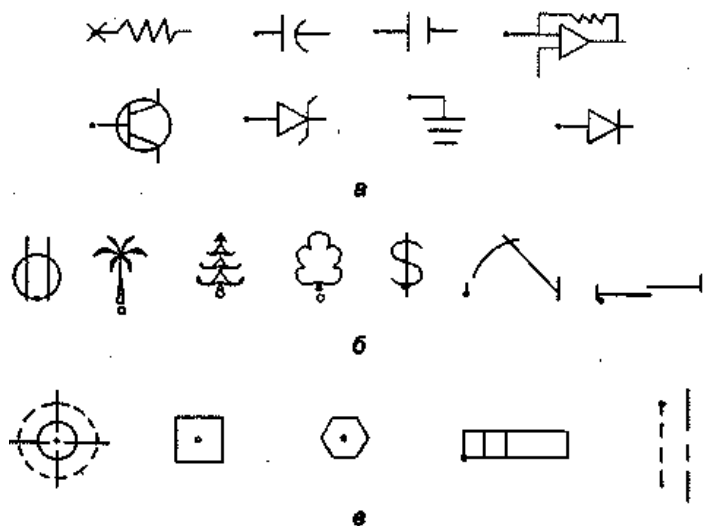
Tez-tez foydalanib turiluvchi shakl (figura)lar simvollar ko`rinishida saqlanishi va xotiradan istalgan onda, chizmaning kerakli joyiga qo`shib qo`yilishi uchun, chaqirib olinishi mumkin. Masalan, boltlar va gaykalar kabi standart komponentlarning shakllari hamda sirt g`adir-budirliklari va qo`yimlar simvollar ko`rinishida saqlansa va kerak bo`lgan istalgan onda chaqirib olinib, chizmaga qo`yilsa, mashinasozlik chizmalarini yaratish ancha osonlashadi. Simvollar bilan ishlash funksiyalari kopyalash funksiyalari kabi ishlaydi va taxminan o`sha tarzda realizatsiya qilinadi.

AutoCAD Release 14 da simvolni quyidagicha chizish mumkin:

Command: block

Block name (or ?): /* Imya soxranennogo simvola */

Elektr sxemalarida hamda arxitektura va mashinasozlik chizmalarida tez-tez foydalaniladigan tipik simvollar 4.8-rasmda keltirilgan.



4.8-rasm. Chizmalarda foydalaniladigan tipik simvollar

a – elektr sxemalarida; *b* – arxitektura chizmalarida; *v* – mashinasozlik chizmalarida

4.3.4. Makrodesturlash

Makrodesturlash – bu grafik komandalar majmuini bir nom ostida birlashtirishdir. Agar grafik komandalar *makrodestur* (*macro program*) deb nomlanuvchi dasturga birlashtirilsa, ularga oddiy kompyuter tillaridan ba`zi shartli va arifmetik operatorlar qo`shilishi mumkin. Grafik komandalarning kiruvchi parametrlari o`zgaruvchilar sifatida aniqlanishi mumkin, bu mikrodesturga turli qiymatlar beri shva har xil chizmalar olish imkonini beradi. bunday makrodestur *parametrik dastur* (*parametric program*) deyiladi, chunki u quradigan chizma mos parametrlarga berilgan qiymatlarga bog`liq bo`ladi. Vintlar chizmalarini qurish avtomatik dasturi parametrik dasturga yaxshi misol bo`ladi. Foydalanuvchi yuklanish xarakteristikalarini kiritadi, dastur bu xarakteristikalar bo`yicha vintlar o`lchamlarini hisoblaydi, so`ngra hisoblab topilgan o`lchamlarni hisobga olgan holda ularning chizmalarini quradi. Vintlar o`lchamlarini hisoblash imkoniyatini

beruvchi arifmetik operatorlar hamda vint chizmasini quruvchi grafik komandalar bunday parametrik dasturlarga kiradi. Makrodasturlash funksiyasining ahamiyati juda katta, chunki u chizmalarni avtomatlashtirilgan ishlab chiqish kommersiya tizimlarini muayyan ilovalarga moslashtirish imkonini beradi. kompaniya tomonidan ishlab chiqilgan parametrik dasturlarning har xilligi, amalda ushbu kompaniya tomonidan unda mavjud bo`lgan chizmalarni avtomatlashtirilgan ishlab chiqish tizimidan samarali foydalanish mezonini bo`lishi mumkin.

4.3.5. O`lchashlar

O`lchash funksiyasi tayyo yoki qurilayotgan chizma bo`yicha hisoblarni bajarish imkonini beradi. tizim istalgan jabha yuzasini, ikki kesma orasidagi burchakni, grafik elementlar orasidagi minimal masofani va boshqa parametrlarni aniqlash imkonini beradi. agra chizmani chizish va loyihalash baravariga chizmalarni avtomatlashtirilgan ishlab chiqish tizimida bajarilayotgan bo`lsa, bu funksiya juda foydali bo`lishi mumkin. Masalan, loyihalovchi olingan konstruksiya issiqlik uzatish yuzasi talablariga yoki xizmat ko`rsatish uchun minimal joy talablariga muvofiqligini tekshirishi mumkin.

AutoCAD Release 14 da o`lchash quyidagicha amalga oshiriladi.

Command: dist

First point: /* Vibiraetsya pervaya tochka. */

Second point: /* Vibiraetsya vtoraya tochka. */

Distance = <Rasschitannoe rasstoyanie>

Angle in XY plane - <Ugol v ploskosti XY>

Angle from XY plane - <Ugol k ploskosti XY>

Delta X = <Raznost` no X>. Delta Y = <Raznost` po Y>, Delta Z = <Raznost` po Z>

4.3.6. Qo`shimcha funksiyalar

Yuqoridagi bo`limlarda bayon qilingan funksiyalardan tashqari yordamchi funksiyalar mavjud, ular ma`lumotlar bazasida chizmalarni yangilab turish,

ma`lumotlar bazasidan chizmalarni olish va materiallar ro`yxatini tuzish imkonini beradi.

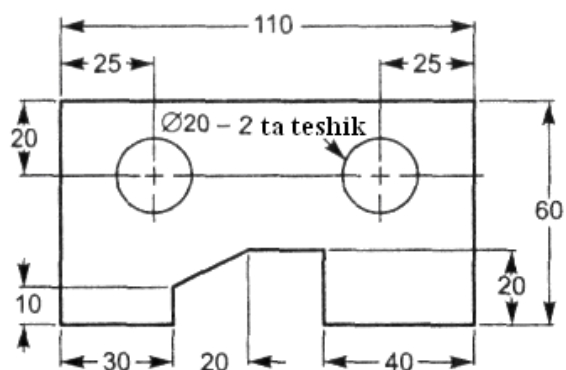
4.4. Chizmalar fayllarining mosligi

Ilgari bu chizma faylini ma`lumotlar bazasida saqlash va u erdan fayllarni boshqa bo`lim xodimlari olib olishi imkoniyati chizmalarni avtomatlashtirilgan ishlab chiqish tizimining afzalligi ekanligini bayon qilgan edik. Agar hamma xodimlar chizmalarni avtomatlashtirilgan ishlab chiqishning bitta tizimida ishlashayotgan bo`lishsa va boshqa xodimlar tomonidan qilingan fayllarni o`qishda qiynalishmasa bu afzallikdan amalda osonlik bilan foydalanish mumkin. Lekin bir kompaniyaning har xil bo`limlari bir-birlarining fayllarini o`qish qobiliyatiga ega bo`lmagan turli tizimlarda ishlashsa, bu afzallik yo`qotiladi. Agar gap turli ishlab chiqaruvchilarning tizimlari haqida ketsa, muammo yanada murakkablashadi. Bu holda o`zaro muloqotning yagona real metodi – bu chizmalarni qog`ozda chizish bo`lib qoladi.

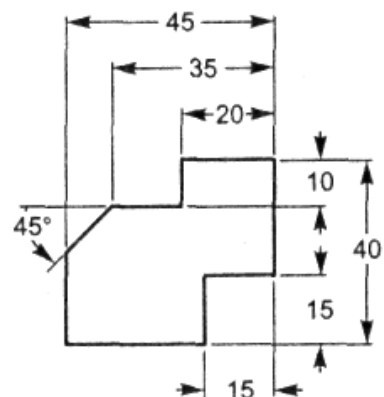
Bu muammoni hal qilish uchun chizmalarni avtomatlashtirilgan ishlab chiqish tizimlarini ishlab chiqaruvchilarning hammasidan fayllarni standartlashtirilgan formatda saqlashni talab qilishdir. Hozirgi paytda eng keng tarqalgan standartlashtirilgan format – bu *Amerika Milliy standartlar instituti (American National Standards Institute – ANSI)* qabul qilingan format – *Initial Graphics Exchange Specification (IGES)*dir. Amalda chizmalarni avtomatlashtirilgan ishlab chiqishning hamma kommersiya tizimlari IGES ni quvvatlaydi. Demak, bir tizimda yaratilgan fayllar boshqa tizimga o`tkazilishi mumkin. Lekin ba`zi simvollar uchun korrektili o`tkazish muammosi ham yechilmagan. IGES dan tashqari DXF formati –AutoCAD chizmalarining formati mavjud, u AutoCAD populyarligi hisobiga «amaldagi» standart bo`lib bormoqda.

Savollar va topshiriqlar

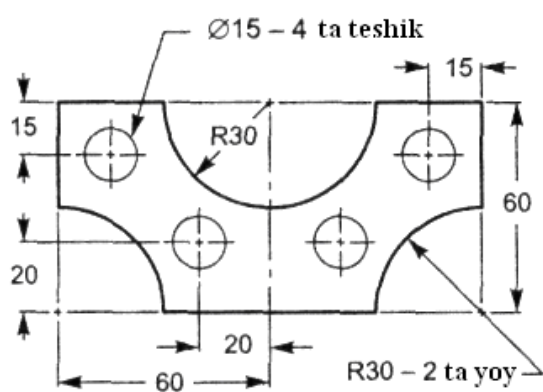
1. Chizmalarni avtomatlashtirilgan ishlab chiqish tizimi yordamida quyida keltirilgan chizmalarni kopyialang. Sizga o`lchamlarni qo`yib chiqish shart emas.



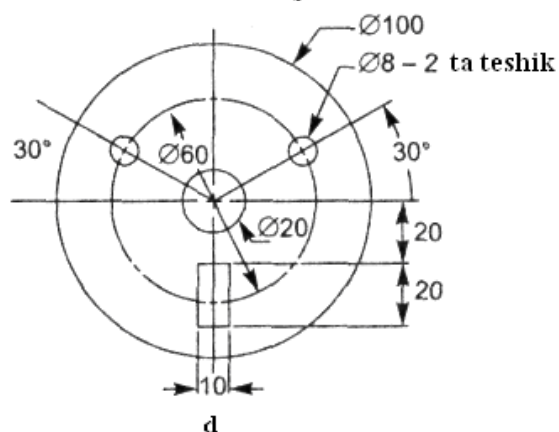
a



b

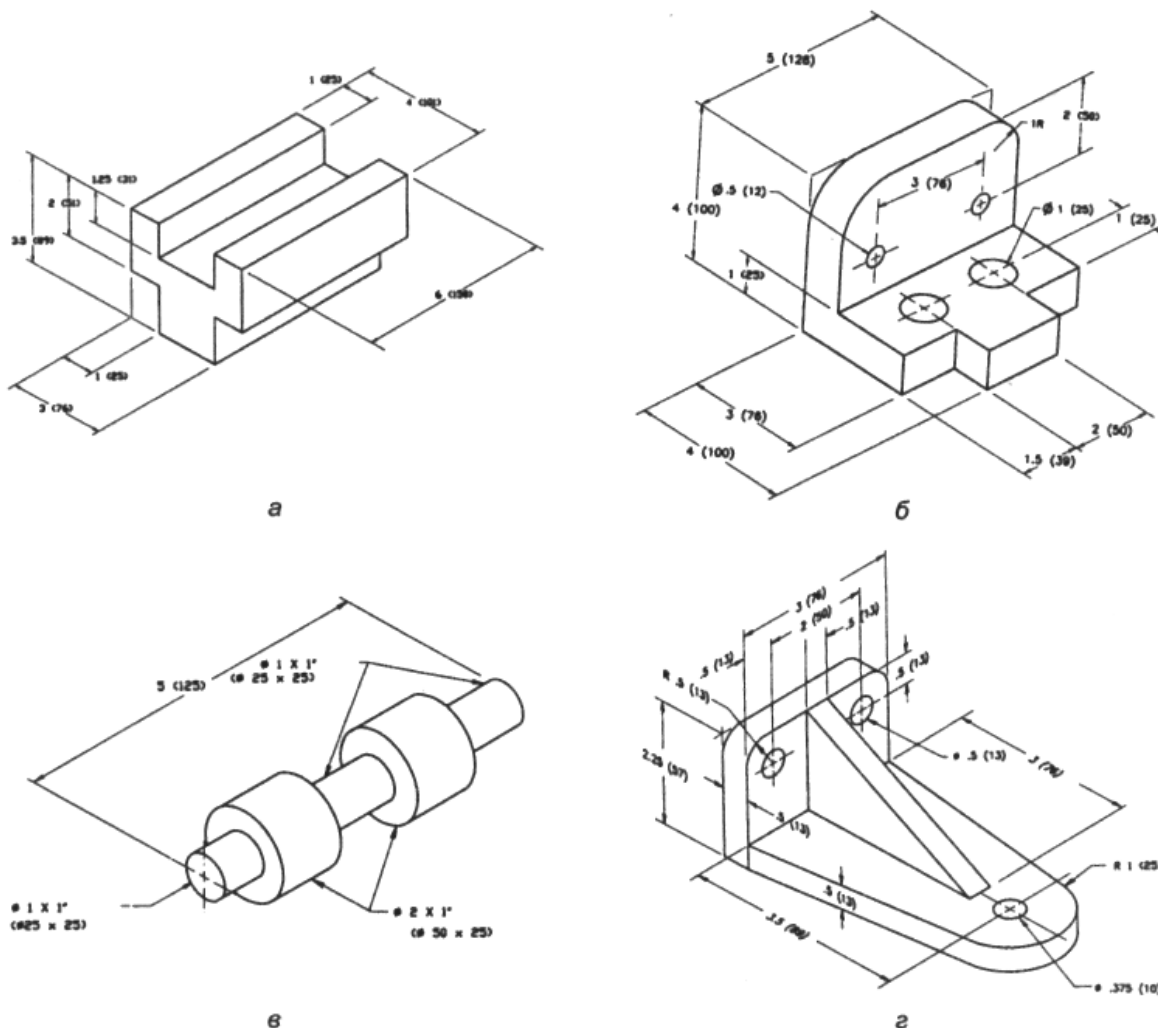


c

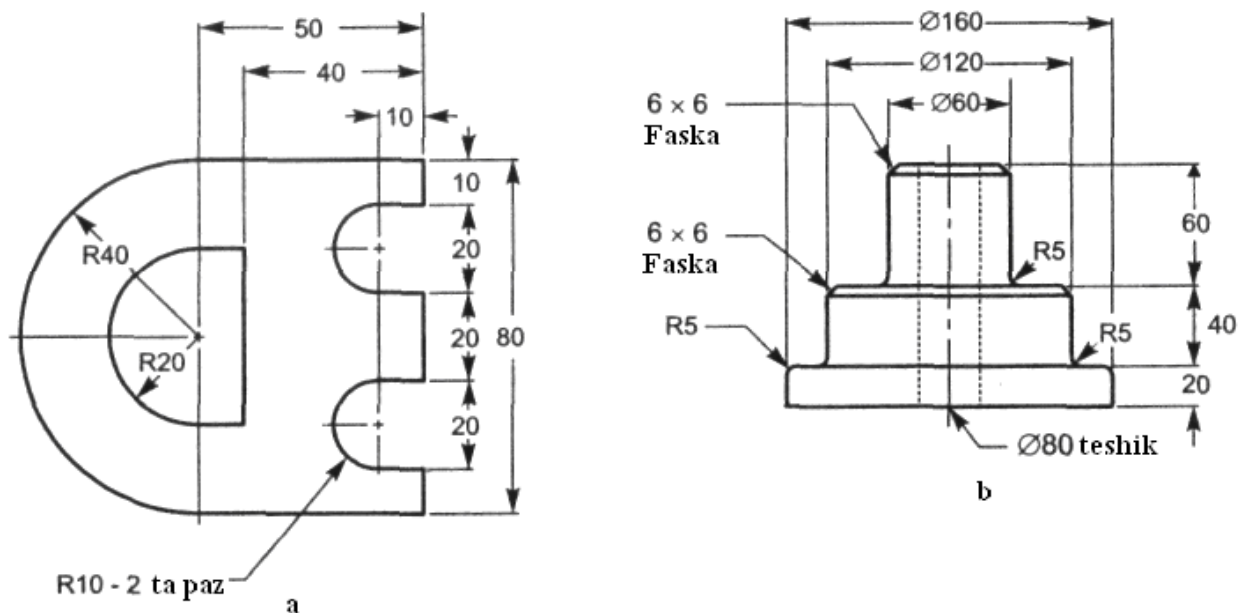


d

2. Quyidagi obyektlar uchun yuqoridan, oldindan va o'ngdan ko'rinishini quring. Sizga o'lchamlarni qo'yib chiqish shart emas.

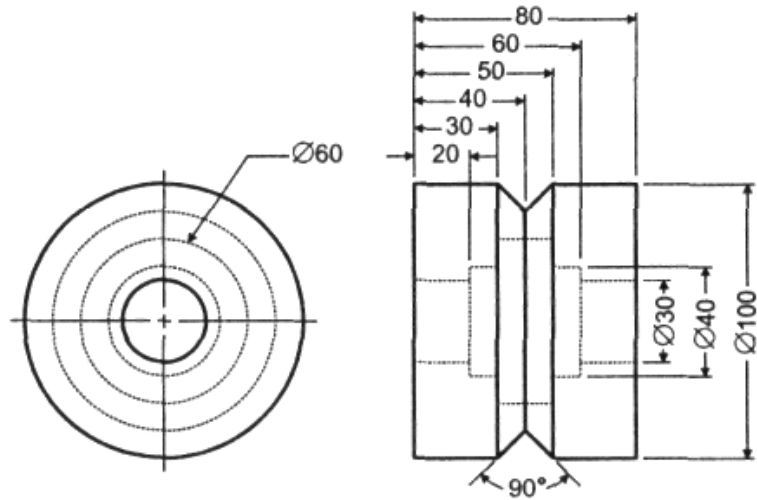


3. Quyida keltirilgan chizmalardan kopiya oling.

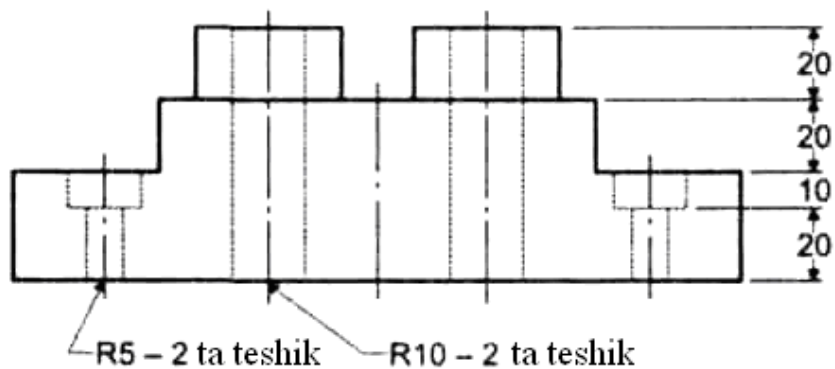
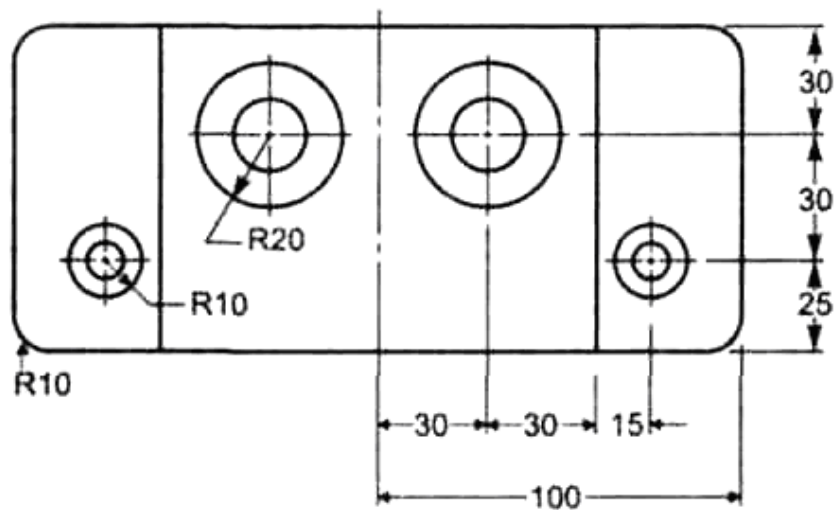


4. Rasmda keltirilgan chizmalarni yarating.

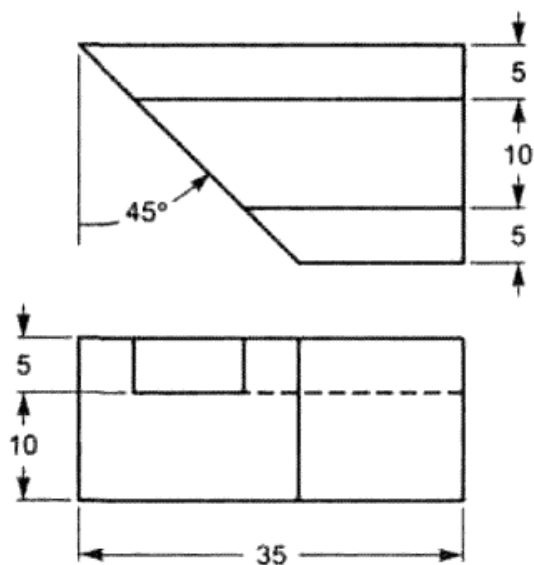
1) O`ngdan ko`rinishni oldidan ko`rinishning vertikal o`rta chizig`i bo`ylab olingan kesim bilan almashtiring.



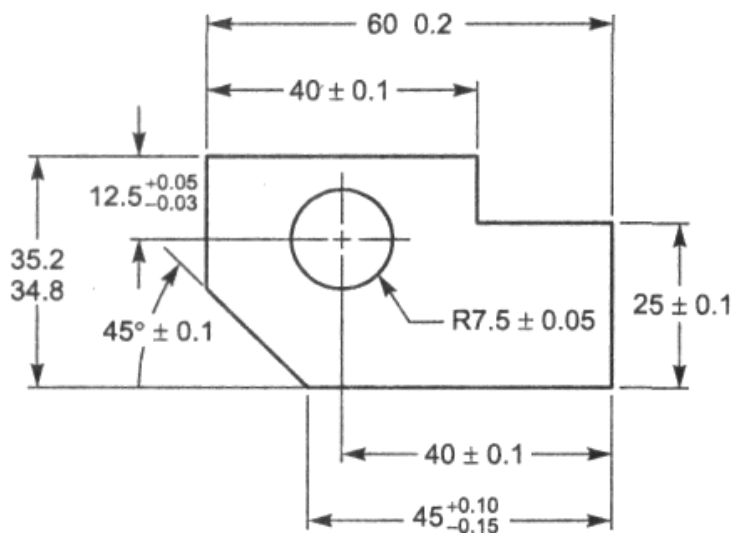
2) Olddan ko`rinishni qisman kesim ko`rinishida taqdim eting va yuqoridan ko`rinishda mos kesuvchi tekislikni ko`rsating. Zarur bo`lgan hamma izohlarni qo`shing.



3) Og`ma sirtning qo`shimcha ko`rinishini yarating.



4. Quyida keltirilgan chizmani o`lchamlari va qo`yimlari bilan birga kopyalang.



GLOSSARIY

Avtomatlashtirilgan loyihalash (CAD) – bu loyihalarni yaratish, o`zgartirish, tahlil qilish va optimallashtirishni osonlashtirish uchun kompyuter tizimlaridan foydalanishga asoslangan texnologiyadir.

Avtomatlashtirilgan ishlab chiqarish (CAM) – bu korxonaning ishlab chiqarish resurslari bilan bevosita yoki bilvosita interfeys orqali ishlab chiqarish operatsiyalarini rejalash, boshqarish asoslangan texnologiyadir.

Avtomatlashtirilgan konstruksiyalash (computer-aided engineering – CAE) – bu CAD geometriyasini tahlil qilish, modellashtirish va mahsulot konstruksiyasini takomillashtirish uchun uning xususiyatlarini o`rganish uchun kompyuter tizimidan foydalanishdir.

Aks ettirish (mapping) – koordinatalarni bir tizimdan ikkinchisiga o`tkazishdir.

Grafik dasturlash (graphics programming) – «inshoning» kirishi va chiqishida ko`p hollarda grafik informatsiya bo`ladi.

Darcha (window) – bu monitorga proyeksiyalanayotgan fazoning bir qismi bo`lib, u orqali foydalanuvchi ushbu tarmoqqa ulangan hisoblash resurslari bilan o`zaro ta`sirda bo`ladi;

Display fayli (display list) – bu grafik kutubxona komandalarining guruhi bo`lib, u keyinchalik bajarilishi uchun saqlanadi.

Kompyuterda dasturlash (computer programming) – «inshoni» kompyuter komandalari tilida belgilangan grammatika qoidalariga muvofiq yozish.

Kuzatish nuqtasi (view site) – obyektning «ko`rish nuri» yo`nalishini aniqlovchi nuqtasi; ko`rish nuqtasidan mo`ljalga o`tkazilgan vektor kuzatish yo`nalishini belgilaydi.

Qurilmaning virtual koordinatalar tizimi (virtual device coordinate system) – hamma ishchi stansiyalar uchun hisob boshlanadigan nuqtani, o`qlarning yo`nalishi va masshtabini fiksatsiya qiladi.

Qurilmaning koordinatalar tizimi (device coordinate system) – ekranda nuqtaning holatini belgilaydi; bu tizim gorizontal o`q u va vertikal o`q v dan tarkib topadi.

Ko`rinmas sirtlarni yo`qotish (hidden-surface removal) – kuzatuvchiga ko`rinmaydigan sirtlar aksini blokirovka qilish.

Ko`rinmas chiziqlarni yo`qotish (hidden-line removal) – kuzatuvchiga ko`rinmaydigan kesmalar aksini blokirovka qilish.

Ko`rish nuqtasi (viewpoint) – bu kuzatuvchining ko`zi.

Ko`rsatish ekрани (view port) – proyeksiyalanayotgan tasvir tasvirlanadigan ekranning bo`lagi; bu bo`lakka ko`riladigan hajm proyeksiyalanadi, u «oddiy» darcha deyiladi.

Makrodasturlash – bu grafik komandalar majmuini bir nom ostida birlashtirishdir.

Parallel proyeksiya (parallel projection) – obyektning hamma nuqtalarida chiziqlar kuzatuvchidan kuzatish yo`nalishiga parallel o`tkaziladi, bu nuqtalarning ekran bilan kesishish nuqtalari proyeksiyani shakllantiradi.

Perspektiv proyeksiya (perspective projection) – ko`rilayotgan obyektning hamma nuqtalari proyeksiya markazi bilan birlashtiriladi, u odatda ko`rish nuqtasini mo`ljal bilan birlashtiruvchi chiziqda yotadi.

Primitivlar (primitives) – bu grafikaning elementlari, ular grafik kutubxona tomonidan aks ettirilishi mumkin.

Singan chiziq (polyline) – bir-biriga birikkan kesmalar majmui.

Foydalanuvchining grafik interfeysi (graphical user interface – GUI) – bu dasturaviy ta`minot menyu yoki belgilarni aks ettirish uchun darchani ochish (muloqot jabhasi) hamda menyu punktlari va belgilari qandaydir funksiyalarni etkazish imkoniyatiga ega.

Chizmalarni avtomatlashtirilgan ishlab chiqish tizimi (computer-aided drafting system) – bu ishlab chiquvchiga interaktiv rejimda mashinasozlik, arxitektura, muhandislik chizmalarini, elektr sxemalarini va yana ko`p boshqa

turdagi chizmalarni yaratish va o`zgartirish imkonini beruvchi dasturaviy mahsulotdir.

ADABIYOT

1. **Глушаков С.В., Лобяк А.В.** AutoCAD 2008. Самоучитель / изд. 2-е, доп и перераб. – М.: АСТ: АСТ МОСКВА: Хранитель, 2008. – 448 с.
2. **Двигатели внутреннего сгорания.** В 3 кн. Кн. 3. Компьютерный практикум: Учеб./ В. Н. Луканин, М.Г. Шатров, А. Ю. Труш и др.; Под ред. В.Н. Луканина. – М.: Высшая школа. 1995.
3. **Корячко В.П., Курейчик В.М., Норенков И.П.** Теоретические основы САПР. – Минск.: Вышэйшая школа. 1987.
4. **Красильникова Г., Самсонов В., Тарелкин С.** Автоматизация инженерно-графических работ. – СПб.: Изд-во «Питер», 2000. – 256 с.
5. **Кунву Ли.** Основы САПР (CAD/CAM/CAE). –СПб.: Питер, 2004. –560 с.
6. **Левицкий В.С.** Машиностроительное черчение и автоматизация выполнения чертежей. – М.: Высш. шк., 1998. – 433 с.
7. **Максимей И.В.** Имитационное моделирование на ЭВМ. М.; 1988.
8. **Петров А.В., Черненький В.М.** Проблемы и принципы создания САПР. – М.: Высшая школа. 1990.
9. **Пятаев А.В.** Автокад. – Т.: ТГАИ. 2008. – 74 с.
10. **Ткачев Д.А.** AutoCAD 2007. – СПб.: Питер, Киев, ВНУ, 2007. – 464 с.
11. **Тулаев Б.Р.** Основы автоматизированного проектирования. Учебное пособие. – Т.: ТашГТУ. 2004.
12. **Хрящев В., Шипова Г.** Моделирование и создание чертежей в системе AutoCAD. – СПб.: ВНУ, 2006. – 224 с.
13. **Bendsoe, M.P., Diaz, A., and Kikuchi, N.** «Topology and Generalized Layout Optimization of Elastic Structures», In Bendsoe and Scares (eds.), Topology Design of Structures, Kluwer Academic Publishers, Dordrecht, Holland, 1992.
14. **Kohn, R.V. and Strang, G.** «Optimal Design and Relaxation of Variational Problems», Communic. Pure and Appl. Math., Vol. 39, (Part 1), pp. 113-137; (Part 2), pp. 139-182; (Part 3), pp. 333-350, 1986.

- 15. Tickoo, S.** AutoCAD: A Problem-Solving Approach, Release13 DOS, Delmar, Albany, NY, 1995.
- 16. Tulaev B.R. Zakirova N.S.** Basic of computer and design. The textbook. –Т., 2005. – 131p.
- 17. Zeid, I.** CAD/CAM Theory and Practice, McGraw-Hill, New York, 1991.

MUNDARIJA

| | |
|---|-----------|
| MUQADDIMA | 3 |
| 1 – BOB. ALTGA KIRISH | 5 |
| 2 – BOB. ALT KOMPOENTLARI | 9 |
| 2.1. APPARAT TA`MINOTI | 9 |
| 2.2. APPARAT VOSITALARINING KONFIGURATSIYASI | 12 |
| 2.3. DASTURAVIY KOMPOENTLAR | 14 |
| 2.4. WINDOWS BAZASIDA ALT | 15 |
| 3 – BOB. GRAFIK DASTURLASHNING ASOSIY KONSEPSIYALARI | 19 |
| 3.1. GRAFIK KUTUBXONALAR | 19 |
| 3.2. KOORDINATLAR TIZIMLARI | 21 |
| 3.3. DARCHA VA KO`RSATISH EKRANI | 29 |
| 3.4. PRIMITIVLAR | 32 |
| 3.4.1. <i>Kesma</i> | 32 |
| 3.4.2. <i>Ko`pburchak</i> | 34 |
| 3.4.3. <i>Marker</i> | 35 |
| 3.4.4. <i>Matn</i> | 36 |
| 3.5. GRAFIKANI KIRITISH | 38 |
| 3.6. DISPLAY FAYLI | 39 |
| 3.7. O`ZGARTIRISH MATRITSASI | 41 |
| 3.7.1. <i>Translyatsiya</i> | 41 |
| 3.7.2. <i>Aylantirish (burash)</i> | 43 |
| 3.7.3. <i>Aks tasvir</i> | 49 |
| 3.7.4. <i>Masshtab va ko`zgu kabi aks ettirish matritsalarini</i> | 54 |
| 3.8. KO`RINMAYDIGAN CHIZIQLAR VA SIRT LARNI YO`QOTISH | 55 |
| 3.8.1. <i>Ko`rinmaydigan qirralarni o`chirish algoritmi</i> | 56 |
| 3.8.2. <i>Ko`rinmaydigan chiziqlarni o`chirish algoritmi</i> | 58 |
| 3.8.3. <i>z-bufer metodi</i> | 61 |
| 3.9. FOYDALANUVCHINING GRAFIK INTERFEYSI | 63 |
| 3.10. X WINDOW TIZIMI | 64 |
| 4 – BOB. CHIZMALARNI AVTOMATLASHTIRILGAN ISHLAB CHIQUISH TIZIMLARI | 71 |
| 4.1. CHIZMA PARAMETRLARINI O`RNATISH | 71 |
| 4.1.1. <i>O`lchash birliklari</i> | 72 |
| 4.1.2. <i>Chizma o`lchamlari</i> | 72 |
| 4.1.3. <i>Qatlam</i> | 75 |
| 4.1.4. <i>To`r va bog`lash</i> | 76 |
| 4.2. CHIZMACHILIKNING BAZAVIY FUNKSIYALARI | 77 |
| 4.2.1. <i>To`g`ri chiziq</i> | 77 |
| 4.2.2. <i>Aylana va aylana yoyi</i> | 78 |
| 4.2.3. <i>Splayn</i> | 79 |
| 4.2.4. <i>O`chirish</i> | 79 |
| 4.2.5. <i>Silliqlash va faskalarni olish</i> | 79 |
| 4.2.6. <i>Shtrixlash</i> | 80 |
| 4.3. YORDAMCHI FUNKSIYALAR | 82 |
| 4.3.1. <i>Kopiyalash</i> | 82 |
| 4.3.2. <i>Darcha</i> | 83 |
| 4.3.3. <i>Simvollar</i> | 83 |
| 4.3.4. <i>Makrodasturlash</i> | 84 |
| 4.3.5. <i>O`lchashlar</i> | 85 |
| 4.3.6. <i>Qo`shimcha funksiyalar</i> | 85 |
| 4.4. CHIZMALAR FAYLLARINING MOSLIGI | 86 |
| GLOSSARIY | 91 |

ОГЛАВЛЕНИЕ

| | |
|--|-----------|
| ПРЕДИСЛОВИЕ | 3 |
| ГЛАВА 1. ВВЕДЕНИЕ В САПР..... | 5 |
| ГЛАВА 2. КОМПОНЕНТЫ САПР | 9 |
| 2.1. АППАРАТНОЕ ОБЕСПЕЧЕНИЕ | 9 |
| 2.2. КОНФИГУРАЦИЯ АППАРАТНЫХ СРЕДСТВ..... | 12 |
| 2.3. ПРОГРАММНЫЕ КОМПОНЕНТЫ | 14 |
| 2.4. САПР НА БАЗЕ WINDOWS | 15 |
| ГЛАВА 3. ОСНОВНЫЕ КОНЦЕПЦИИ ГРАФИЧЕСКОГО ПРОГРАММИРОВАНИЯ | 19 |
| 3.1. ГРАФИЧЕСКИЕ БИБЛИОТЕКИ | 19 |
| 3.2. СИСТЕМЫ КООРДИНАТ | 21 |
| 3.3. ОКНО И ВИДОВОЙ ЭКРАН | 28 |
| 3.4. ПРИМИТИВЫ | 31 |
| 3.4.1. Отрезок | 31 |
| 3.4.2. Многоугольник | 33 |
| 3.4.3. Маркер | 33 |
| 3.4.4. Текст..... | 35 |
| 3.5. ВВОД ГРАФИКИ..... | 36 |
| 3.6. ДИСПЛЕЙНЫЙ ФАЙЛ..... | 37 |
| 3.7. МАТРИЦА ПРЕОБРАЗОВАНИЯ | 39 |
| 3.7.1. Трансляция..... | 39 |
| 3.7.2. Вращение | 41 |
| 3.7.3. Отображение | 47 |
| 3.7.4. Другие матрицы преобразования..... | 51 |
| 3.8. УДАЛЕНИЕ НЕВИДИМЫХ ЛИНИЙ И ПОВЕРХНОСТЕЙ | 52 |
| 3.8.1. Алгоритм удаления невидимых граней | 53 |
| 3.8.2. Алгоритм удаления невидимых линий..... | 56 |
| 3.8.3. Метод z-буфера..... | 58 |
| 3.9. ГРАФИЧЕСКИЙ ИНТЕРФЕЙС ПОЛЬЗОВАТЕЛЯ | 60 |
| 3.10. СИСТЕМА X WINDOW | 61 |
| ГЛАВА 4. СИСТЕМЫ АВТОМАТИЗИРОВАННОЙ РАЗРАБОТКИ ЧЕРТЕЖЕЙ | 67 |
| 4.1. НАСТРОЙКА ПАРАМЕТРОВ ЧЕРТЕЖА | 67 |
| 4.1.1. Единицы измерения | 68 |
| 4.1.2. Размеры чертежа | 68 |
| 4.1.3. Слой..... | 70 |
| 4.1.4. Сетка и привязка..... | 71 |
| 4.2. БАЗОВЫЕ ФУНКЦИИ ЧЕРЧЕНИЯ | 73 |
| 4.2.1. Прямая линия | 73 |
| 4.2.2. Окружность и дуга окружности | 73 |
| 4.2.3. Сплайн..... | 74 |
| 4.2.4. Удаление | 74 |
| 4.2.5. Скругление и снятие фасок | 75 |
| 4.2.6. Штриховка..... | 76 |
| 4.3. ВСПОМОГАТЕЛЬНЫЕ ФУНКЦИИ | 77 |
| 4.3.1. Копирование | 77 |
| 4.3.2. Окно | 78 |
| 4.3.3. Символы..... | 78 |
| 4.3.4. Макропрограммирование..... | 79 |
| 4.3.5. Измерения..... | 80 |
| 4.3.6. Дополнительные функции..... | 81 |
| 4.4. СОВМЕСТИМОСТЬ ФАЙЛОВ ЧЕРТЕЖЕЙ..... | 81 |
| ГЛОССАРИЙ..... | 86 |

TABLE OF CONTENTS

| | |
|--|-----------|
| THE FOREWORD | 3 |
| CHAPTER 1. INTRODUCTION IN BAD | 5 |
| CHAPTER 2. COMPONENTS BAD | 9 |
| 2.1. HARDWARE MAINTENANCE | 9 |
| 2.2. A CONFIGURATION OF HARDWARE | 12 |
| 2.3. PROGRAM COMPONENTS | 14 |
| 2.4. BAD ON THE BASIS OF WINDOWS | 15 |
| CHAPTER 3. THE BASIC CONCEPTS OF GRAPHIC PROGRAMMING | 19 |
| 3.1. GRAPHIC LIBRARIES | 19 |
| 3.2. SY0STEMS OF CO-ORDINATES | 21 |
| 3.3. A WINDOW AND THE SPECIFIC SCREEN | 28 |
| 3.4. PRIMITIVE THINGS | 31 |
| 3.4.1. <i>A piece</i> | 31 |
| 3.4.2. <i>A polygon</i> | 33 |
| 3.4.3. <i>A marker</i> | 33 |
| 3.4.4. <i>The text</i> | 35 |
| 3.5. DRAWING INPUT | 36 |
| 3.6. A DISPLAY FILE | 37 |
| 3.7. A MATRIX OF TRANSFORMATION | 39 |
| 3.7.1. <i>Translation</i> | 39 |
| 3.7.2. <i>Rotation</i> | 41 |
| 3.7.3. <i>Display</i> | 47 |
| 3.7.4. <i>Other matrixes of transformation</i> | 51 |
| 3.8. REMOVAL OF INVISIBLE LINES AND SURFACES | 52 |
| 3.8.1. <i>Algorithm of removal of invisible sides</i> | 53 |
| 3.8.2. <i>Algorithm of removal of invisible lines</i> | 56 |
| 3.8.3. <i>A method of the z-buffer</i> | 58 |
| 3.9. THE GRAPHIC INTERFACE OF THE USER | 60 |
| 3.10. SYSTEM X WINDOW | 61 |
| CHAPTER 4. SYSTEMS OF THE AUTOMATED WORKING OUT OF DRAWINGS | 67 |
| 4.1. ADJUSTMENT OF PARAMETRES OF THE DRAWING | 67 |
| 4.1.1. <i>Units of measure</i> | 68 |
| 4.1.2. <i>The sizes of the drawing</i> | 68 |
| 4.1.3. <i>A layer</i> | 70 |
| 4.1.4. <i>A grid and a binding</i> | 71 |
| 4.2. BASE FUNCTIONS OF PLOTTING | 73 |
| 4.2.1. <i>A straight line</i> | 73 |
| 4.2.2. <i>A circle and an arch of a circle</i> | 73 |
| 4.2.3. <i>A spline</i> | 74 |
| 4.2.4. <i>Removal</i> | 74 |
| 4.2.5. <i>A rounding off and removal of facets</i> | 75 |
| 4.2.6. <i>Shading</i> | 76 |
| 4.3. AUXILIARY FUNCTIONS | 77 |
| 4.3.1. <i>Copying</i> | 77 |
| 4.3.2. <i>A window</i> | 78 |
| 4.3.3. <i>Symbols</i> | 78 |
| 4.3.4. <i>Macroprogramming</i> | 79 |
| 4.3.5. <i>Measurements</i> | 80 |
| 4.3.6. <i>Additional functions</i> | 81 |
| 4.4. COMPATIBILITY OF FILES OF DRAWINGS | 81 |
| GLOSSARY | 86 |

THE LITERATURE88